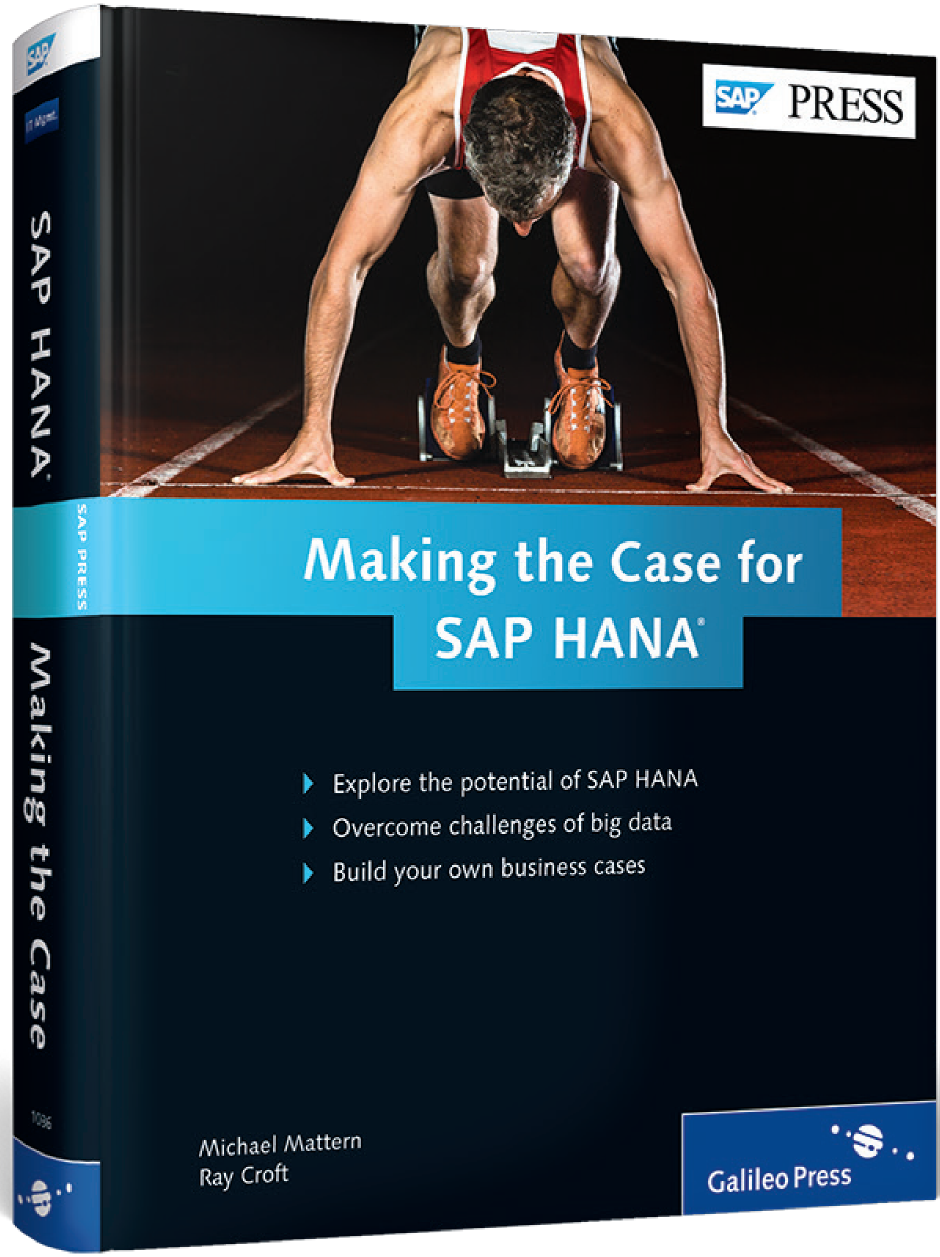
SAP PRESS

First-hand knowledge.



SAP PRESS

Making the Case for SAP HANA®

- Explore the potential of SAP HANA
- Overcome challenges of big data
- Build your own business cases

Michael Mattern
Ray Croft

Galileo Press

Contents

Introduction

Big data—collecting, processing, and analyzing giant amounts of data—has become a buzzword. Hardly a day goes by when the topic does not find its way into newspapers, blogs, or magazines.

Big data: heaven or hell?

Some authors believe big data will take us right into the hell of surveillance, a totalitarian *precog state* (named after the precogs—mutants able to divine the future—in "The Minority Report," a short story by Philip K. Dick) in which we will be punished for crimes that we haven't (yet) committed and might never commit. Others are hailing the brave new world of big data as the key to unlock an earthly paradise that will no longer suffer from traffic jams or incurable diseases.

There could be some truth in both scenarios. But no matter which of them you tend to believe in, the level of attention big data is getting seems to indicate that we are talking about an important new technology—something that will affect us (or might even affect us already) more than we realize. This alone would be reason enough to have a closer look at big data.

Big data and SAP HANA

For SAP customers, there are a few more motivations to think about big data:

- In 2010, SAP introduced their own big data product, *SAP HANA*—at that time, a product still separate from other SAP solutions.
- Just one year later, SAP's former co-CEO Jim Hagemann Snabe reported that SAP HANA had already become the fastest-growing innovation focus from SAP.
- Since April 2012, SAP has offered its data warehousing solution, SAP Business Warehouse (BW), on an SAP HANA platform. The solution is called *SAP BW powered by SAP HANA*.
- In early 2013, SAP announced that the first core applications of SAP Business Suite could now run on SAP HANA.

If you are following the news on SAP's corporate website (*www.sap.com/news-reader/index.epx*), the keynotes of its top managers, or reports within the respective forums of SAP, you have seen that there seems to be no way past SAP HANA. Apart from cloud or mobile computing, the IT industry (and the SAP community) are clearly considering big data (and with it SAP HANA) *the* silver bullet of the millennium's second decade—important enough at least to make major consulting firms gird their loins. In October 2013, Accenture—one of the major global players in IT consulting—announced that they would soon be training 1,000 consultants on SAP HANA in German-speaking countries.

Purpose of this Book

Technical aspects: flood of information

Information about big data or SAP HANA isn't scarce at all; the Internet is teeming with it. SAP has its own website on the topic (*www.saphana.com/welcome*). Likewise, both SAP's help pages (*help.sap.com/in_memory*) and the SAP Community Network (SCN; *http://scn.sap.com/community/hana-in-memory*) have areas exclusively dedicated to SAP HANA. In addition, there is a cornucopia of information on SAP Service Marketplace. A search for "SAP HANA" on Google comes back with some 10 million hits, and Amazon is offering somewhere in the order of 9,000 books in some way or another related to the term *big data*.

Business benefits: information deficit

Then, why another book about SAP HANA? After all, SAP PRESS already has titles such as *SAP HANA: An Introduction* (Berg and Silvia, 2014), *ABAP Development for SAP HANA* (Schneider, Westenberger, and Gahm, 2014), *Implementing SAP HANA* (Haun, Hickman, Loden, and Wells, 2015), and books on related topics, including *Predictive Analysis with SAP: The Comprehensive Guide* (MacGregor, 2014) and *OData and SAP NetWeaver Gateway* (Bönnen, Drees, Fischer, Heinz, Strothmann, 2014) in their portfolio. Despite this, there are three reasons that we believe that there are still information deficits when it comes to identifying potential benefits of SAP HANA:

- Most online sources of information are dealing with the *how*—such as "How do I migrate a classic SAP BW system to an SAP HANA database?"—but not with the *which*, *where*, or *what*, for example: "Which business processes in your industry can be made more efficient in what way by implementing SAP Business Suite powered by SAP HANA?"

- The majority of SAP HANA–related success stories on the Internet (see for example *www.saphana.com/community/learn/customer-stories*) are only casually considering a solution's quantifiable benefits. If *return on investment* (ROI) is discussed at all, you will usually only find one big impressive number that seems to come out of nowhere without sufficient information to back it up. Most boards will probably not be willing to provide program managers with budgets comprising millions of dollars for hardware, software, and services on the basis of lump-sum business cases.
- There are modules available on the Internet (on the SAP Community Network, there is an area reserved for those: *http://scn.sap.com/community/hana-in-memory/use-cases*) that are built around very particular and usually playful requirements, such as predicting your next follower on Twitter using *SAP Predictive Analysis* (search "Predicting My Next Twitter Follower with SAP HANA PAL" on SCN, or refer to the link in the additional resources we recommend in the book's online appendix). Although this might be entertaining and might still deliver valuable insight into the options and the handling of SAP HANA, such examples are of limited use when it comes to identifying chances within your specific environment—unless the number of followers on Twitter happens to be a key performance indicator (KPI) for your business.

Structure of this Book

How to use this book

The book you are holding in your hands is intended to help you fill these gaps. In a short preview, we are going to explain how you may use it to this end. Then, to make sure that you will get the maximum return on your investment (in terms of the price you paid for the book and the reading time you'll need), we will also provide you with an overview of the book's contents. Geared up in this way for your trip into the world of big data, you may then decide whether you would like to attack the subject in a systematic, step-by-step approach or perhaps rather nibble a bit in places that you find particularly relevant or interesting.

What is so new about big data?

In **Chapter 1**, we look at some fundamental concepts and identify the key innovative idea behind big data. In this context, we are going to discover that although big data is a bit more than a fresh take on an old idea, neither SAP nor its competitors have reinvented the wheel. Some of the concepts behind SAP HANA—such as in-memory-technology—have been around for a while and have already been used in older products (such as SAP BW Accelerator [BWA]). One of the main innovations is the fact that preexisting but previously independent solutions—such as enterprise resource planning (ERP), data warehousing, very fast databases, and inferential statistics—are now being integrated and merged. In this respect, SAP HANA goes an extra mile compared to other big data solutions (thanks to the bandwidth of SAP's product portfolio). We are going to show you that bundling such tools is creating new perspectives—perspectives that go far beyond generating cost center reports within seconds rather than hours.

SAP HANA = big data?

Building on these essential discoveries, **Chapter 2** will provide you with a closer look at some of the technical resources sailing under the SAP HANA flag. In addition, we are going to shed some light on the question of whether SAP HANA and big data are two different words for the same thing or whether one of them is a subset of the other. In the course of this analysis, we are going to identify the essence of SAP HANA (from a conceptual point of view, not from a technical one) and what SAP HANA-based solutions can and can't be used for. Finally, we'll round off your knowledge by pointing out particular challenges associated with implementing and using SAP HANA by providing you with information about the implementation scenarios available today and by sharing with you some of our (subjective) views regarding trends and future development perspectives. Once you have a good understanding of SAP HANA's key functions, you'll also have the basis for understanding the potential benefits discussed in the following chapters.

Cross-industry and industry-specific benefits

SAP HANA delivers potential benefits that are either cross-industry/cross-process or industry/process specific. In **Chapter 3**, we first consider potential benefits that are not tied to particular industries or business processes. Once we get to industry- and process-specific considerations, we are going to get our bearings from the so-called value maps in SAP Solution Explorer (*https://rapid.sap.com/se*). Through value maps, we will have a closer look at a couple of business processes and related value drivers.

How to create shareholder value

Looking at business processes and value drivers simultaneously will help you understand how and why SAP HANA can be used to generate shareholder value. Toward the end of Chapter 3, we are going to explain which criteria were applied to select the case studies used in **Chapter 4** through **Chapter 11**. On the basis of these eight (fictitious but not unrealistic) case studies portraying different business processes in a number of industries, we attempt to make the benefit types or categories identified in Chapter 1 (Figure 1.2) a bit more tangible. Unlike most of the SAP HANA success stories on the Internet, we are not going to head for a detailed analysis of individual cases. Instead, we will be extracting general principles from those individual scenarios.

Our case studies all follow the same pattern:

Status quo

- **As-is situation**
  We start with a nonjudgmental description of a case—that is, the background against which the example is unfolding.

The problem/ opportunity

- **Problems, costs, risks, and chances**
  Based upon this analysis, we are going to define the specific problem or the specific chance that is hidden in the scenario. In doing this, we are seeking to see problems and chances through business eyes—that is, turning our attention to facts that are commercially significant rather than exciting from a technical perspective.

What to do, how to act

- **Solution**
  Having explored problems and chances, we turn to possible solutions. At this stage, we don't look at how to implement things; instead, we focus on describing a possible approach and the potential benefits resulting from it. As most problems and chances are not one-dimensional, we usually review multiple benefit categories and their related value drivers. In this way, we also want to stimulate your own thought processes and make your creative juices flow.

How can SAP HANA help?

- **Implementation scenario and data architecture**
  Each chapter ends with a closer examination of how, why, and to what extent SAP HANA might be a suitable tool to implement the solution. We take a look at (case-specific) technical peculiarities (for example, in terms of the data architecture) and offer advice for implementation projects. Our hints regarding the data architecture are always built upon an architectural framework for SAP HANA–related

domains and layers presented in Section 4.5.2 (Figure 4.17) and Section 6.4 (Figure 6.2). With each case study, we then focus on deviations or distinctive features.

Needless to say, we are not able to come up with concrete data points for the case studies. If numbers are mentioned, we have made them up over a glass of Australian Pinot Noir. Their only purpose is to make abstract concepts more tangible and to prepare you for similar analyses in your domestic battlegrounds.

Eight case studies and summarized findings

The case studies in Chapter 4 through Chapter 11 are used to illustrate the generic perceptions presented in Chapter 3 and to derive new insights from more specific scenarios. Finally, **Chapter 12** summarizes some of these findings, using them to derive a decision matrix and some general guiding principles for the data architecture of SAP HANA-based solutions:

- **Decision matrix: implementation scenarios**
  The decision matrix will help you find the right implementation scenario for your own business cases (based upon the implementation scenarios defined in Chapter 2).
- **Architectural principles: general recommendations**
  The chapter sums up the lessons to be learned from each case study's paragraph on data architecture and uses them to derive ten general rules of thumb that you can use when designing your own data architectures.

Throughout the book, you'll find icons to assist you in your reading:

[+] This icon highlights further information about the topic at hand.

[»] This icon highlights terminology and definitions.

[Ex] This icon highlights case studies or examples.

[👁] This icon highlights a summary of an important theme.

Getting the message across

We all live in the age of *infotainment*. The term infotainment is derived from the words "information" and "entertainment" and describes the presentation of information in an entertaining way. Not just in advertising but also in politics and for internal presentations, the packaging of a message is sometimes even more important than its contents. Apple, for example, invests almost as much effort in designing nice boxes for their products as in creating the gadgets contained in them.

Online content for this book

Apart from the physical book you hold in your hands, there is also some unique online content available. At *www.sap-press.com/3647*, you will find a value driver database, an example for a tagged video (we'll cover tagged videos in Chapter 8), and quite a few hyperlinks directing you toward online sources of information. Some of them you may already know. There is, however, an enormous amount of material about big data and SAP HANA on the Internet; we hope you may still find the odd tasty morsel that you have not yet discovered.

To ensure that this book is also fun to read and entertaining (infotainment, remember), we have garnished our reflections with a fictional background story. Derek—a senior solution consultant with a Belgium-based IT consulting firm—travels the world brooding over project-related woes (some of which might sound familiar to you). Mulling over his customers, he sometimes comes to almost philosophical conclusions about SAP HANA, big data, and life in general.

Acknowledgments

Success has many fathers. Failure is an orphan.

Following this aphorism sometimes attributed to the English economist Richard Cobden (1804–1865), we would like to thank some of this book's fathers (be they male or female) and—in addition—adopt responsibility for all errors and omissions.

Major parts of this book have been conceived in the course of retreats at picturesque monasteries; special thanks go to the Cistercians of Southern Star Abbey at Takapau (Hawke's Bay, New Zealand; *www.kopuamonastery.org.nz*), the Brothers of the Capuchin monastery at Rapperswil (St. Gallen, Switzerland; *www.klosterrapperswil.ch/*), Brother Remigi Odermatt, the monks of the Prieuré de Ganagobie (Ganagobie, Département Alpes-de-Haute-Provence, France; *www.ndganagobie.com*), the Missionary Benedictines of the St. Otilien archabbey (Bavaria, Germany; *https://www.erzabtei.de*)—particularly Father Augustinus Pham—and the Benedictines of Buckfast Abbey (Buckfast, Devon, United Kingdom;

www.buckfast.org.uk). We thank all of them for their hospitality and for a couple of inspirations in terms of mindfulness.

The audacity to bear the challenge of writing a book at all is owed to a couple of chats with Sonja Schwarzl of SAP Germany and Thomas Schmischke of Return on Concept (*www.roconcept.de*). Strength to overcome occasional writer's block has been delivered by Beatrice Sigrist (*www.sigristcoaching.ch/*). Advice regarding health management and related topics has been provided by Christina Gramsma-Zimmermann. We received valuable contributions in terms of ideas, concepts, and structures from Michael's friend and mentor Hans Klott, our contributor Marcia E. Walker (who also wrote parts of Chapter 10), and from Wayne Pohe of Xelocity New Zealand (*www.xelocity.com*).

Last but not least, special acknowledgements go to Michael's wife Silke Mattern (both for providing the photographs at the beginning of each chapter and for forbearance and catering during intensive writing phases at home), to Ray's partner Bronwen for her hospitality during Michael's stays in Australia, and to Michael's and Silke's Birman cat Gina (see beginning of Chapter 11) for helping him to relax after writing. We also owe a lot to our editors Janina Schweitzer (with SAP PRESS Germany) and Kelly Grace Weaver (with SAP PRESS USA) for their tolerance in terms of the many amendments and quite a few chapters arriving later than expected (we know we really gave you a hard time...).

Michael Mattern and **Ray Croft**

Reality is merely an illusion, albeit a very persistent one.

Attributed to Albert Einstein

1 Big Data: More than Just Performance

The cold mistral had set in again. Derek pulled up his knees and moved over into the sun. Since the beginning of the week, the icy wind had been sweeping from the Scottish Highlands along the Rhone valley down into Provence. Thankfully, down here he was well sheltered. Protected by the valley's steep slopes and the cloister's walls, it was hard to believe that it was just the beginning of March. In Brussels, rush hour traffic would now be throwing watery mud on pedestrians hastening away from their offices to the trams through drizzling rain. Down here, the evening sun pleasantly stroked his face.

He had never been in the cloisters at this time of the day before; maybe that was why he never noticed the strange projection the setting sun threw on the sandstone. Derek stood up and walked over to the wall. Was this really a procession of monks who had thrown over their hoods, or was he just imagining it? It was certainly possible that crafty stonemasons had found a way to work round the Cistercian's strict building code. Bernard of Clairvaux, the father of the order, had banned distracting images; monasteries following his rule refrained from carved ornaments and from color. Or did he see figures in hoods where there was nothing but amorphous shades only because his brain found the idea amusing?

The investment planning meeting two weeks ago leapt into Derek's mind. The CFO of a dairy group had found out that the variance of two samples could be compared using Excel's FTEST function and had decided to let a comparison formula loose on all datasets he could find in the data warehouse. Subsequently, he came to the conclusion that the plant in his home town, *Charleroi (where he would soon run for a seat on the council), showed—in his words—a significantly lower variation in product quality than all other production sites. Hence, it was essential that this plant should be expanded as soon as possible and should also produce the latest product innovation: a yogurt that contained sherbet.*

Figure 1.1 Optical Delusion at the Cloisters of the Abbey of Sénanque, Département Vaucluse, France

Stephane van Leeuwen, the production manager responsible for a factory in the Flemish part of the country and significantly involved with developing Tickle-Gurt (the new yogurt's future trade name), had responded that Charleroi's stability was indeed impressive; the factory was consistently delivering lousy quality, and even this rather low quality level could only be maintained because only their simplest product—UHT milk—was being produced there.

Instead, he suggested, one should think about shutting down the plant or at least deleting Charleroi as a place of production from product packaging due to the negative effect on the company's image. After all, everybody in Europe knew that Charleroi had just been awarded the honorary title of being the world's ugliest town in a representative opinion poll.

Following that interjection, the CFO had mumbled something about Lenin and forged statistics, had left the meeting with a glowing red face, and had gone hotfoot to the CEO to discuss Monsieur van Leeuwen's future career.

Derek had been left behind stunned and trying not to imagine how many more fantastical ideas the CFO might be able to spawn following the upcoming *implementation of SAP Predictive Analysis on an in-memory database (SAP HANA). Imagining the first couple of meetings after going live gave him the creeps.*

Quite often, we hastily jump to conclusions in good faith only because these conclusions fit so well into our cognitive patterns. In the case of the dairy company, big data will probably not lead to improvements in that respect. On the contrary, big data is going to create the opportunity to produce more erroneous insights quicker using 80 CPU cores and to convert these insights into incorrect decisions faster than ever before!

Within this chapter, we first want to define exactly what big data means from our point of view. We want to emphasize that big data is a lot more than umpteen cores and fast databases and also a lot more than just raw processing power. Subsequently, we are going to provide you with a couple of ideas about how (and under which conditions) big data solutions may create potential benefits, where (that is, in which business processes) such potentials could occur, and how they can be realized in terms of shareholder value. In this context, we are also going to explain the term *shareholder value* and discuss via which factors (or *value drivers*) benefit can be converted into shareholder value. Based upon three dimensions (benefit, business process, and shareholder value), we are going to come up with a method that you could use when evaluating existing project proposals as well as searching for new ideas.

This approach—introduced in the last section of this chapter—will also become the framework for our case studies.

1.1 What Does Big Data Mean?

Big data: definition

A 2012 study of the German industry association BITKOM defines big data as follows:

> *Big data supports the commercially reasonable extraction and utilization of decision-relevant insights from qualitatively multifaceted and structurally diverse information which is subject to fast change and which is available in quantities yet unknown. Big data reflects the technical progress*

within the last couple of years and comprises strategic approaches developed for it as well as technologies, IT architectures, methods, and algorithms employed with it.

Technologies, architectures, methods, algorithms

Hence, big data does not refer to a particular technical solution but instead serves as an umbrella term that embraces technologies, architectures, methods, and algorithms, all of which are all aligned to achieve one objective: to gain and to utilize decision-relevant insights.

But if this is the essence of big data, then what is really new about it? Were R/3, R/2, their grandfather R1, or—even earlier (from 1973 onwards)—SAP's very first product, RF, not also meant to deliver and utilize decision-relevant insights, even in real time, as the "R" in their product names indicates?

Performance is just a means to an end

Five additional properties contained in the preceding BITKOM definition provide a clue that big data is not about gradual performance improvements (however impressive such improvements might be) but about a revolution in the way data can be used: *qualitatively multifaceted*, *structurally diverse*, *fast change*, *quantities yet unknown*, and *technical progress*. The first four properties are related to new challenges created by input data, and the last one refers to responses to these challenges.

Considered together, however, new technologies, architectures, methods, and algorithms can do more than just enable you to respond to new challenges. They also create new opportunities that could not be exploited prior to the arrival of big data.

More data than ever

Before we turn to these new opportunities, let us just briefly discuss the kind of changes we are dealing with in terms of input data and processing technologies:

- **Qualitatively multifaceted and structurally diverse data**
  Nowadays, organizations have to analyze growing amounts of very diverse, often unstructured data (text, natural/spoken language, audio files, images, videos, etc.). Since the early days of IT, humans—being the more flexible party—have always adapted to the limitations of computers.

  Our input interfaces were limited to screen templates; when filling in machine-readable forms, we properly placed every single letter in its own separate field. But a travel agent who only looks at data people deliberately enter into structured feedback forms might not even notice that 80 of 100 customers let him have it straight from the shoulder, calling his 5-star-plus flagship hotel Mosquito Beach a "cockroach-infested flophouse" on an online review site. If this should happen on a regular basis, the business' days—or at least the days of its marketing director—might be numbered.

- **Fast change**
  The business environment within which we have to make data-based decisions no longer changes every year, month, or day but rather every minute or even every second. An idea that might have been brilliant an hour ago can be old hat just moments later. If you have ever sold a used book on Amazon, you may be aware that the system advises you of the current best price for that specific book. If that current best price is $5, you—being a clever salesman—are going to post your offer at $4.99. But just after you press the `Enter` button, a message goes out to one of the other traders (triggered, for example, by Google Alert—or by a bot monitoring quotations on Amazon), making him reduce his price to $4.98.

- **Quantities yet unknown**
  Nowadays, data are available in quantities we did not even dare dream about in the good old days of R1. In the early days of SAP, mainframes had about one to four megabytes of main memory, which is about one thousandth of what a vanilla iPhone 6 can come up with nowadays. If you want to load a video recorded with that iPhone into main memory (for further analysis), four megabytes would (depending on *codec* and compression rate) only hold some 0.1 to 0.4 seconds of footage. Even worse, according to an article in the German *Welt* newspaper from July 16, 2013, the volume of all data worldwide is currently doubling every other year. Major organizations no longer talk about gigabytes or terabytes but petabytes when sizing data warehouses.

- **Technical progress**
  Around the middle of the last decade, a new type of database came into existence: one that is able to execute reports and analyses on very large amounts of data faster than ever before. We are talking about so-called in-memory databases (IMDBs, also called main memory database systems [MMDBs], memory-resident databases, or real-time databases [RTDBs]), which no longer read data from hard disks or

other persistent media but straight from a server's main memory. In the course of this book, we are going to show that this new technology can do a lot more than just make existing IT solutions faster.

Development of in-memory databases

In the beginning, in-memory technologies were only used to mirror data, which resided on hard disks in main memory (caching). For example, SAP BW Accelerator (BWA)—available since 2007—was based upon this principle. Since then, however, main memory has evolved into a primary long-term-storage medium. Better quality and lower prices for memory chips, the overcoming of the working storage's capacity limits by *distributed computing systems* (an approach via which calculation-intensive jobs are no longer processed by one very powerful machine but are instead split by software and then dealt with by a more or less loosely coupled network of computers—a virtual super computer also called a *cluster* or *grid*), the option to replicate data within multiple redundant, highly available environments, and new technologies for delta backups and logging have provided the necessary prerequisites. Today, memory-resident databases fulfill the so-called *ACID* (*atomicity*, *consistency*, *isolation*, and *durability*) requirements that are indispensable in a commercial setting.

Moore's law

In-memory databases also obey Moore's law, which states that the performance of new computer chips doubles about every 20 months. SAP HANA appliances, for example, can use the latest multicore processors with 64-bit architectures and suitably fast memory chips.

Once working storage becomes the database, not only retrieval but also write performance can be improved. Furthermore, modified data might not have to be stored persistently and then replicated once more into main memory. Instead, newly generated data are immediately available for further processing. In terms of data retrieval within SAP Business Suite, SAP is talking about data access being more than 3,600 times faster than before; when it comes to writing (as, for example, when activating data within DataStore Objects [DSOs]), SAP is still claiming a hundredfold improvement.

1.1.1 In-Memory Databases as a Key Technology

It follows that in-memory databases are one of big data's key technologies, which is why we will explain at this stage what exactly is meant by an *in-memory database*.

In-memory databases

An in-memory database is a database system that (at least in the first instance) does not deposit its data on conventional hard disks or *solid-state drives* (a persistent storage device that is used like a classic hard disk but does not contain any moving parts and is therefore a lot faster) but instead uses *random-access memory* (the term *random-access memory* [RAM] is—more or less—synonymous with main memory). Therefore, the basic theoretical principle behind in-memory databases is to only write into or read from main memory.

Supplementary persistent data storage

However, to fulfill the previously mentioned ACID requirements for database systems (and to save those responsible for IT from sleepless nights), this basic theoretical principle is broken in a couple of ways. When using main memory as the primary long-term storage location, it's chiefly the "D" (for "durability") of ACID that causes concern. Valuable business data must under all circumstances and at all times be conserved in case of software errors, power outages, or natural disasters, which is why even in-memory databases usually work with supplementary persistent data storage.

Safety nets for IMDBs

Such databases usually have the following safety nets at their disposal:

- **Redundancy**
  Holding data redundantly on a variety of levels allows high availability of all systems used. Nowadays, *high availability* means an availability that is distinctly higher than 99.99 percent; within a year, 99.99 percent still represents almost 53 minutes downtime!
- **Persistency/backups**
  Changes to data are not only recorded in main memory but also written to (delta) logs simultaneously. Furthermore, persistent and consistent images of the complete database are backed up periodically (at so-called *savepoints*). Furthermore, conventional backup tools are used to create backups of all relevant data (such as the logs). Frequently, solid-state drives (SSDs) are the medium of choice to store logs and backups.
- **Appliances**
  In-memory databases are frequently shipped in the form of appliances—that is, harmonized combinations of hardware and software. One-stop shopping for all components should (theoretically) reduce

the risk of issues arising from incompatibilities or unforeseen interactions between hardware and software and thus reduce the probability of failure.

- **Other measures**

  Irrespective of the above, with in-memory databases most organizations take the same kind of measures with respect to data safety and security as with all other IT systems (spatial redundancy, de-meshing, no single points of failure, etc.). *Spatial redundancy* refers to the fact that data and their backups should be held in different places, *de-meshing* describes the concept that individual components should not belong to more than one network, and *single points of failure (SPOFs)* are parts of a system that—if they fail—bring the whole system down.

[»]

Appliance: hardware and software

Appliance

An appliance (such as SAP HANA) is an integrated product that consists of hardware and software and is developed to execute one or more specific functions.

It is not common practice for IT or users to exchange individual components or modify source code (as it often is when using traditional hardware solutions or software packages). An appliance in IT has got a lot in common with a household appliance (such as an oven), which is typically sealed and not meant to be modified, reprogrammed, or maintained by its owner—but can be programmed or operated by its users to perform different tasks at different times for different people. Typical advantages of appliances are user-friendliness, reliability, and high performance; the main disadvantage is the customer's dependency on one supplier.

In some ways, the idea of an appliance takes us back to the early days of the IT industry. In the world of mainframes, the hardware, operating systems, and peripherals were all made by the same vendor. Apple, for example, even sticks to this philosophy today in terms of their operating systems, which may be one of the reasons why Apple's products are said to be highly dependable but also relatively expensive.

Other examples of appliances are IBM Netezza, Cisco UCS, Oracle Exadata, and Fluke Networks Visual TruView. IBM Netezza and Oracle Exadata serve purposes similar to that of SAP HANA; Cisco UCS and Fluke Networks Visual TruView deliver more specialized functionalities related to operating IT infrastructures.

Virtual memory

For safety reasons and also because the price of one gigabyte of main memory is still about 100 times higher than the cost of the same amount of (hard) disk capacity, even in-memory databases sometimes put data on persistent storage media (used as *virtual memory*).

Data that are only rarely accessed (so-called *cold data*) are stored on hard disks (usually not conventional disks but solid-state disks); *hot data*, which need shorter access times, stay in main memory. Sometimes, data swapped out into virtual memory are even further split into cold and *warm data*; cold data end up on conventional hard disks and warm data on solid-state disks (the cost per gigabyte is some 10 times higher than with conventional hard disks, but disk access times are about 1/100 of those of conventional hard disks).

Replication into main memory is not new

The idea of speeding up data retrieval and data analysis by keeping large amounts of data in main memory didn't suddenly and unexpectedly pop up in the middle of the last decade. In the 1990s, replacing hard disks by main memory was being tested and evaluated. Even SAP started to develop a number of in-memory solutions as early as 1999; many of these (SAP liveCache, Text Retrieval and Information Extraction [TREX], Business Intelligence Accelerator [BIA]/Business Warehouse Accelerator [BWA]), however, contained only very limited functionality compared to SAP HANA.

Conceptual features of IMDBs

The extremely high performance of in-memory databases is not only a result of sheer (hardware-driven) computing power. Most in-memory databases share the same conceptual features:

- In-memory databases usually store their data in a column-oriented and compressed format. Both steps are taken to save (relatively expensive) working storage and to speed up access.
- When writing data, these first reside in a separate storage area optimized for write access. This so-called *delta storage* is then assimilated into the general (columnar and compressed) database, either periodically or upon request. This practice enables fast write access when inserting new records without having to reorganize the complete columnar database with every transaction that modifies existing data.
- The organization of the database is guided by the principles of temporary and spatial locality. In the age of globalization, the concept of spatial locality is not only related to the main memory's address space but also simply to the question of where data are stored geographically.

- Packaging respective solutions as appliances is not only a measure that helps reduce the probability of failure. Optimally matching the components to each other and—in the long run—fine-tuning them based upon the vendor's experience with many implementations can also help improve performance.

Very much like the idea of reading data from main memory (rather than retrieving data from disks), the conceptual approaches listed previously are not entirely new; some of them (such as columnar storage) have been used before with classic, disk-based relational database–management systems.

1.1.2 What Else Do You Need For Big Data?

Big data needs more than just a lot of memory

The simultaneous appearance of new challenges on the one hand and new technical solutions on the other has certainly raised the profile of big data over the last couple of years. Nevertheless, super-fast read or write access alone is not enough to get a grip on huge volumes of data. As suggested by the BITKOM definition at the start of Section 1.1, you also need algorithms, methods, and architectures.

Algorithms for big data

In terms of algorithms, four areas of expertise are of special importance for big data:

- **Inferential statistics (for example, time-series analysis)**
  Descriptive statistics tries to describe data (such as net income and state of residence of all US employees throughout the last 50 years) using aggregated key figures (average income of employees per state and year).

  In contrast, *inferential statistics* goes one step further, trying to identify dependencies (for example, between income and state of residence), verify suspected interdependencies, or deliver forecasts (average incomes of employees in Texas next year). When time is an important factor as well (because trends and seasonal patterns are overlapping with such interdependencies), experts also resort to the field of time-series analysis (such as spectral or wavelet analysis).

  Dependencies and forecasts are not ends in themselves but rather serve businesses and other organizations as a basis for decisions that need to make sense in future—but yet unknown—environments (for example, which states are we going to focus on when it comes to opening new organic supermarkets?).

Going one step beyond descriptive and inferential statistics, *exploratory data analysis* (EDA) has gained considerable significance—not least because of big data. Rather than testing or confirming a preexisting hypothesis, EDA focuses on identifying correlations and on generating new hypotheses.

In Chapter 4 and Chapter 7, we work with case studies that use classic inferential statistics, whereas the ones in all other chapters—at least to a certain extent—also use methods/algorithms from exploratory data analysis.

- **Computational linguistics (for example, speech recognition and text mining)**

  Computational linguistics is an interdisciplinary field of knowledge dealing with modeling natural language from a computational perspective.

  Computers you can talk to are no longer science fiction. Solutions that can process simple free text or even spoken commands (individual words from a relatively small vocabulary) have been around for a couple of decades.

  You may have enjoyed meeting one of them when you last called your airline or your telephone provider. Even dictation systems that can convert spoken language from a far bigger lexicon into written text are not brand new; one of the first (TANGORA 4) was presented by IBM at the German CeBIT exhibition in 1991. If you are prepared to invest a bit of time in training, current solutions such as Dragon can deliver quite good results.

  Computational linguistics can be divided into two separate areas, the first of which delivers the raw material for the second:

  - *Speech recognition*

    Although trying to make sense of written free text is a challenge for machines, using spoken language to control systems or to produce documents is in a different league—even more so if we are talking about more comprehensive vocabularies and different speakers. In such cases, computers have to deal with a variety of pitches, moods, dialects, rates of speaking, vocabularies, and sentence patterns. There has been quite a bit of progress in these fields thanks to improved algorithms and increased computing power. Substantial progress has been made and is available to all of us when we use speech-recognition systems such as Google Now (on Android machines) or Siri (on Apple hardware).

- *Text mining*

  Text mining is the next logical step after speech recognition. The purpose of text mining is to enable machines to actually *understand*—within certain limits—the meaning of (written) texts. If the travel agent mentioned in Section 1.1 wants to analyze what customers are saying about Mosquito Beach on Facebook, Twitter, blogs, or travel portals, he could employ a team to work three eight-hour shifts seven days a week, 24 hours a day to check the Internet for this information. This would, however, be work intensive and quite expensive, and inevitably the result would be delivered too late. Even an enormous number of workers would not be able to analyze what is happening on Twitter alone (a couple of million tweets per day within the so-called *public streams*—the portion of short messages on Twitter that is available publicly and accessible via standardized interfaces) let alone on the Internet as a whole, no matter how much time they had to do it.

  Computers are a lot better at systematically combing through gargantuan amounts of data than are humans. The key challenge for machines, though, is assigning human expressions of opinion to categories such as "positive" or "negative." A lot has been achieved in this area as well, mainly due to enhanced statistical algorithms (often summarized under the term *data mining*, of which text mining is a subset) and the technical progress that provides the performance required to use them. Nevertheless, text mining still has its limitations; once irony or cynicism enters into the picture, computers struggle. How do you translate the concept of sarcasm into source code? A computer might take the statement "Considering your age, you still look very attractive" as a compliment; most people we know wouldn't share that view.

- **Geospatial data (for example, fleet management)**
  If somebody asked us to come up with a list of innovations created during the last 20 or 30 years (sorted by the effect these innovations had on our everyday lives), we would probably start with mobile communication and satellite navigation delivering geospatial data (as provided by the US military's NAVSTAR Global Positioning System [NAVSTAR GPS]).

  The idea of capturing and using positioning data is a few thousand years old; indeed, for the navy, GPS serves the same purpose as the astrolabe, the sextant, or radio navigation. There are, however, a couple of fundamental differences that are not owed to any individual new technology but to an intelligent combination of a number of inventions:

  - Today, positioning no longer takes a matter of hours but—due to faster central processing units (CPUs)—only seconds.
  - Instead of navigation based solely upon the stars above, we now have points of reference based on stationary orbit satellites permanently broadcasting certain data using electromagnetic waves. We are able to receive these broadcasts and to determine our position everywhere (apart from a few exceptions, such as in tunnels or basements or in urban canyons) and—unlike when using the stars—in all kinds of weather.
  - Thanks to very exact atomic clocks, GPS allows us to determine the position of ships, aircrafts, vehicles, or other objects accurately to about 10 meters horizontally and 35 meters vertically. The new European Galileo system is going to further improve horizontal accuracy to less than a meter from (hopefully) 2020 onwards.
  - Mobile communication enables us to continuously transfer positioning data to analytics solutions in real time and to respond to these data by providing feedback, recommendations, or instructions. Such feedback might be based upon the data received but might also use other information that the object (for example, the vehicle or the aircraft) does not have at its command (such as data about traffic jams or thunderstorms along the intended route).

- **Computer vision (for example, object recognition)**
  Nowadays, we are able to search and categorize still and moving images by their content, and as humans seem to be exhibitionist by nature the millions of pictures and videos posted on the Internet can deliver a wide variety of useful information. *Object recognition* (one discipline within computer vision) tries to extract useful information from such data.

  Using face recognition on holiday images (often unintentionally enriched by attached geospatial data; did you know that your smartphones, tablets, and modern cameras add data to pictures that includes your location? Send the picture to someone and you are also telling them where you were when you took the picture) can tell us who usually travels where and when. Although customers still protest against mobile phone providers selling their movement profiles to interested parties, video systems analyzing our travel paths within a shop are nothing new. Robot mannequins that can identify our genders and ages have been around for quite some time, and they will probably soon become even smarter and be able to read our moods and our willingness to buy. (Want to know more? Visit the link for "Why Do Mannequins That Spy on Us Creep Us Out?" in the additional resources we recommend in the book's online appendix.)

  You can test for yourself how good even freely available object recognition systems have become. Just go to *https://www.google.com/imghp* and drag a picture of (say) a famous building from your desktop into the search field.

Other relevant areas of expertise

Apart from these more general realms, other more specific disciplines, some but not all of which are subdisciplines of the ones just described, play a major role for big data. Examples of such subdisciplines include *fraud detection* (detecting fraud with credit card transactions by ferreting out certain patterns), *sentiment detection* (automatically analyzing texts to categorize these into positive and negative utterances), and *medical diagnosis* (which sometimes uses expert systems or statistical algorithms to infer diagnoses from symptoms or findings). In this context, inferential and explorative statistics are not only fields in their own right but also form the foundation for most application areas mentioned here (which is why we will not get around dipping our toes into a bit of mathematics and statistics in this book).

Methods

Big data not only needs computing power and sophisticated analysis tools often based upon mathematical or statistical algorithms. It is also based upon (relatively) new methodological concepts, including new approaches in terms of software engineering. Compiling and discussing a complete list of all of these methods would go well beyond the scope of this book; however, one such—more or less representative—example of these methods is *agile business intelligence* (agile BI). The term agile BI is based upon *agile software development*, a group of methods, about ten years old, trying to come up with new paradigms for software development. Instead of working in clearly delimited phases, agile software development relied on a less structured, iterative process. Agile software development came into being because development teams wanted to design software whose functionalities better matched user's requirements and could be adapted to changing environments faster than before.

Agile BI transfers the same principle to business intelligence. Business experts no longer have to deliver detailed reporting requirements before IT even starts implementing anything. Instead, both sides gradually determine final analytical requirements. The ultimate consequence of agile BI is a scenario in which the IT department no longer deploys solutions (this job is taken over by the users) but instead provides an implementation framework (such as a consistent database and clearly defined metadata).

Agility and speed go hand in hand

The idea of agile BI is linked to big data because agile BI only makes sense under certain circumstances:

- If it takes minutes or even hours to execute a report, iteratively developing requirements becomes a very cumbersome and time-consuming exercise. Agile BI needs an environment in which reports can be executed and evaluated without substantial wait times.
- If you need to build complex models or process chains just to test the simplest statistical algorithms (such as in SAP BW's Integrated Planning [IP] solution or in SAP BW's Analysis Process Designer [APD]), then agile BI remains nothing but an illusion. The integration of the statistical language R into SAP HANA or supplementary solutions such as SAP Predictive Analysis (PAL) opens new doors; on the other hand, such tools also make great demands on the user's mathematical and statistical knowledge.

- The user needs instruments that allow her/him to easily create or modify reports. Traditional SAP tools such as BEx Query Designer showed substantial shortfalls in that respect, but the latest releases within the SAP BusinessObjects BI portfolio are moving in the right direction.

Big data architectures

Finally, big data also needs new approaches in terms of data architectures for three reasons:

- The data architectures of quite a few existing business intelligence implementations relentlessly pursue the optimization of reporting performance—for example, by aggregating data through a series of layers between data acquisition and reporting. Big data solutions can often do without such preaggregations and therefore manage with less (persistent) layers.
- Certain object types (for example, *OLAP cubes*, a multidimensional array that stores data in a specific way that optimizes query performance) were invented because classic databases could not deliver the performance that was needed for flexible queries on aggregated data. Using faster in-memory databases eliminates (in most if not all cases) that restriction and therefore also the need for such objects and the data flows supplying data to them.
- New approaches in terms of the distribution of tasks between central IT and users lead to new priorities when defining the structure of systems. Fewer layers and fewer persistent objects can lead to leaner data architectures; they do, however, also raise the question of how the correctness and consistency of reports can still be guaranteed in the light of ever more virtualization, flexibility, and decentralization. This also leads to additional questions for data architects.

SAP has responded to these architectural challenges in a number of ways. The classic layered scalable architecture (LSA) model for SAP BW, for example, has been replaced by the so-called extended layered scalable architecture (LSA++), and new tools for enterprise information management (EIM) were introduced. (We talk more about LSA++ in Chapter 4, Section 4.5.2.)

Analysis: only the first step

So far, we have clarified the term big data a bit, but all our considerations up to this point have been dealing with the job of analyzing, interpreting, or exploiting some kind of data. Our example about selling a book on Amazon has already shown that algorithms, methods, and architectures for fast analyses are only half the battle. What kind of benefit do you get if your computer or your mobile phone knows you have been undercut on Amazon or overbid on eBay in the middle of the night? The process would only be complete if your mobile phone could wake you up at 2 a.m. and ask you to adapt the prices. Or—better for your beauty sleep—if a smartphone could do that for you automatically.

In Section 1.2, we are going to explain that *acting* faster is one of the key factors in terms of really bringing in the benefits of big data solutions. But due to this insight and for all following considerations, we already need to define big data in a slightly different way.

[«]

Big data: extended definition

Big Data—Extended Definition

Slightly different from and supplementary to BITKOM's definition, we will summarize our understanding of big data: Big data is an umbrella term for all technologies (for example, in-memory databases or columnar instead of row-oriented data storage), architectural approaches (for example, LSA++), methods (for example, agile BI), and algorithms (for example, text-mining algorithms) that can be used to analyze and to process very large amounts of diverse and/or unstructured data. The objectives of such analyses are gaining new insights, making better decisions based upon these insights, and implementing decisions more quickly or more wisely.

1.1.3 Is it All Just About Performance?

The glossy brochures of big data suppliers are bristling with impressive performance figures. They mention access times, goodput, storage capacity, and ever more processing cores—a kind of discussion that reminds us of acceleration, engine torque, cylinder capacity, and horsepower.

[Ex]

Performance might be useless

Sales Report

Willie Countem, a controller working for a medium-sized manufacturer of toilet seats, needs to produce a monthly report on product volumes sold to wholesalers, craftsman's establishments, and DIY stores. The report is usually generated on the fifth working day of the following month. After checking the numbers, Mr. Countem prints the report and puts it into an envelope that will be sent to the CFO's secretary via internal mail. In about the middle of every month, the company's directors meet with managers that are responsible for production and sales; in the course of this meeting, the information provided by Mr. Countem will be reviewed and used to decide on appropriate actions.

Mr. Countem joined the company as an apprentice; in the course of 45 years in his job, he has seen a lot of technological change. In the 1960s, sales figures were collected from salesmen via phone, aggregated with a mechanical adding machine, and added to printed forms by hand. Over time, the adding machine was replaced by a desktop calculator, the desktop calculator by a piece of homemade software running on a personal computer, and finally an ERP (enterprise resource planning) solution for medium-sized enterprises was implemented. As the report was reasonably complex, it always took a couple of minutes before it popped up on the screen, enough time for Mr. Countem to get hold of a fennel tea for his stressed stomach.

Recently, however, Mr. Countem's work rhythm has gone out of kilter. In March last year—after a laborious day at the *CeBIT* trade fair in Germany—the company's head of IT ended up at the booth of a data warehousing supplier. Maybe it was the two glasses of red wine, but he decided there and then to make a substantial investment in a data warehouse with more than 20 terabytes of main memory and 160 CPU cores, using the latest in-memory technology. The following implementation project went through quickly and without any resistance; toward the end of the same year the new data warehouse went live.

The result: it now only takes milliseconds to bring up the sales report, the former head of IT has got a lot of time to go fishing, and Mr. Countem's digestion is—due to a lack of time to brew himself some tea—completely derailed.

This short and fictitious scenario might sound a bit cheeky, but it still illustrates one thing: the fact that big data solutions can make reporting a thousand times faster does not generate shareholder value (we are going to return to the concept of shareholder value in Section 1.4), and even less so if these super-fast reports are generated a month after the relevant events and are taken note of—at the earliest—another two weeks later.

Speed is necessary but not sufficient

Admittedly, executing very complex queries in SAP Business Suite and even in SAP BW (without BWA or SAP HANA) can be painstakingly slow. In some 20 years of consulting, we have encountered quite a few reports that were on screen not within minutes but rather hours after pressing the Enter key. With a big group of companies, the previous day's sales figures were available just before closing time the day after (due to long load processes and unacceptable report runtimes). However, despite all criticism regarding the performance of SAP Business Suite, this is still an exception rather than the rule, and with many business processes it is absolutely irrelevant whether a report that is supposed to be executed overnight in batch now only needs a fifth of a second instead of five minutes.

Of course, as Mr. Countem now has to spend a substantial part of his working day in the bathroom anyway, the sales report could be delivered to top management on mobile devices via SAP BusinessObjects Dashboards (instead of printing and forwarding it via internal mail). Top management could check the latest numbers by just gently touching the screens of their tablets, and Mr. Countem could join the former head of IT at the banks of Trout River.

Up-to-date information delivered faster and smarter

Now, instead of executing the report once a month, it could be updated daily, and selected results could be forwarded to managers in sales and production as tweets or via *live tickers* (those tickers—sometimes also called *slides* or *crawlers*—that normally reside in the lower third of your TV screen, presenting headlines, stock quotes, and the like). But what exactly are people supposed to do with these real-time data, and how is the increase in reporting speed and frequency going to create shareholder value?

1.2 How Do Benefits from Big Data Come About?

Is big data not really the disruptive innovation of the decade or the century, the game changer for all industries and businesses, the perfect storm that is going to blow everybody who does not care about it out of their home markets like withered leaves?

Exploiting hidden data treasures

Well, let us have a look at two masters who have been performing with virtuosity on the keyboard of big data for years.

[Ex]

Masters of big data: Amazon

Amazon

During the last two decades, the *e-tailer* (electronic retailer) Amazon (founded in 1994) has ripped apart bookselling. In the USA, it has dramatically changed the world of publishing, and we can safely assume that the dominance of the retail giant will extend not only to the whole world but also to other electronic media and in general to many more retail goods and the complete value-creation chain.

When it comes to big data, Amazon is one of the early adopters (and—by the way—also one of the pioneers in distributed and cloud computing, two key approaches related to big data). We won't go into great detail on how exactly Amazon analyzes and exploits huge amounts of data and what these are already used for or can be used for in the future (apart from personalized book recommendations). However, one could, for instance, imagine that the kind of books you buy says a lot about the risk of selling you a health insurance policy; hence Amazon might be better than the current top dogs when it comes to calculating insurance premiums.

Further information about such topics can be found on the Internet (just search online for "amazon" and "big data," and skip the first entries in the list of results, which are, significantly, books offered by Amazon). Amazon's expertise in the realm of big data is one reason that SAP has decided to make SAP HANA and other solutions available via Amazon's cloud (*http://aws.amazon.com/sap/*).

[Ex]

Masters of big data: Google

Google

Google (which came into existence in 1998) is another Internet giant. In the press, there are frequent reports suggesting that Google systematically scans our emails and data and that this data juggernaut sells its insights (or maybe even forwards them to the NSA free of charge). But, the probability that you use Google when searching the Internet is still around 68 percent if you are based in the United States, and above 90 percent if you live in Germany (regardless of the fact that Germans are usually far more concerned about their privacy than Americans). Be honest: do you still remember the names of once leading and highly rated search engines like Excite, Lycos, or AltaVista? Or are you one of the few remaining bastions of (relative) freedom and privacy still depending on AOL's search function (1.3 percent market share in the United States) or even more trustworthy but also exotic ones like ixquick?

Details of Google's indexing, ranking, and search algorithms are as secret as the Coca-Cola recipe. According to Google, however, more than 200 factors contribute to a page's ranking. Although pages might be evaluated in batch (which could still cause problems in terms of the up-to-datedness of search results), search requests that take into account the user's search history or location or that are supposed to perform some kind of text mining using current or previous search terms have to be dealt with in real time.

If it is true that Google has supposedly indexed a few trillion URLs, then it becomes absurd to assume that such volumes could be managed by utilizing traditional relational databases and without any specially developed algorithms, methods, or big data architectures. Instead, Google has become a trailblazer for big data-related approaches.

The MapReduce programming model developed by Google has become one of the standard methods in big data environments. By the way: some proof of how big Google's databases and processing capacities have become can be found in the fact that the company is slowly moving toward the North Pole to cool its giant computing centers with icy sea water.

[«]

MapReduce Programming Model

MapReduce is a programming model developed and patented by Google; it is used to process huge amounts of data in many parallel processes on computer clusters. Many common programming languages, such as Erlang, Java, and Python, provide MapReduce-based algorithms.

When it comes to designing distributed systems, MapReduce has become one of the key tools; it is also part of Apache Hadoop (a framework written in Java used to store very large amounts of data).

Global dominance or stomach pain?

So, why are some companies in a position to grab global dominance in their respective markets, but big data means nothing but stomach pain for Mr. Countem? Having worked in information technology and business intelligence for quite a few years, we have found at least three ways that data (big or small) can deliver potential benefits. In this section, we are going to introduce you to each one of them. Our list might not be complete, and you may well be able to improve it further based upon your own requirements; nevertheless, it will still help you generate ideas.

1.2.1 Gaining New Insights, Making Better Decisions

Knowing more about your customers

If big data can provide you with insights you were not aware of before, then these insights can help you make better decisions. Even if you guessed some of these new insights before, big data can still help you verify your assumptions or monitor their validity over time.

[👁]

Better Decisions by New Insights

Big data can help you make better decisions by generating new insights from existing data and by continuously checking the explicit or implicit assumptions on which your decisions are based. Unfortunately, such benefits do not occur inevitably or automatically. In the course of this book, we also mention quite a few pitfalls you could encounter.

[Ex]

Cross-Selling More without Foregoing Margins

You have noticed that customers buying pork cutlets (priced at one dollar per piece, of which 10 percent is your margin) also reach for homemade grill sauces placed in the vicinity (priced at five dollars per bottle, of which 30 percent is your margin) with a probability of 65 percent.

If you now—as a special offer—reduce the price of cutlets by 25 percent and thereby increase sales volume by 30 percent, your margin on 100 cutlets sold is no longer $10 but $7.50. On the other hand, you are now selling 130 instead of 100 cutlets, which gives you an additional margin of 30 * $0.075 = $2.25, reducing your loss in absolute figures to $0.25. Even better, you are also pushing 84.5 instead of 65 bottles of grill sauce into the market, thus generating an additional margin of 19.5 * $5 * 30 percent = $29.25, which more than absorbs your loss on the cutlets. Even better, if you knew more about your customers demand curve (see Figure 7.2), you could probably slightly increase the price of grill sauce without a negative effect on sales volume there. Which might not be nice to your customers, but hey, do you know anybody who got loaded by being nice instead of clever?

If your shop managers are smart, they might have discovered the link between selling cutlets and grill sauce and might have drawn proper conclusions from that interdependence long ago and entirely without big data. But what if the correlation is no longer valid because the discounter next door has seen through your success and has thrown a wrench in the works by offering super-cheap grill sauces whenever you lower the price of cutlets?

If you only become aware of that when you check your month-end sales reports, you will already have lost quite a bit of money by selling low-priced meat to customers buying their sauces next door. Big data can help you continuously monitor that interdependence and can inform you once it is no longer in line with empirical data.

1.2.2 Using Sophisticated Tools Properly

Sophisticated tools: chances and risks

In Section 1.1.2, we emphasized that success with big data does not exclusively depend on storage capacity and processing power. Performance is a means to an end, not the end itself. Big data also includes a number of pretty sophisticated methods and algorithms that can deliver exceptional results but also mislead you into decline (think of Tickle-Gurt).

Typically, one or more of the groups of algorithms mentioned in Section 1.1.2 are applied, so if you are confronted with tasks that might profit from statistical forecasting tools or that might have to process written or spoken language, big data might help you exploit new opportunities.

[👁]

Big Data Uses Sophisticated Tools

Big data opens the door to algorithms that might have been around before but could not be used extensively due to a lack of computing power. Some algorithms (for example, in text mining) might have been too demanding, both for batch and for real-time use, whereas others might already be common in your organization but restricted to batch processes that sometimes run for days on dedicated servers.

[Ex]

Customer Segmentation

One example for an SAP HANA solution using an established algorithm in a different way is the SAP HANA Customer Segmentation rapid-deployment solution. Identifying customer segments and assigning customers to them has been possible in SAP Customer Relationship Management (CRM) for quite some time. Clustering algorithms used for that purpose were not new when we went to university—that is, a couple of decades ago.

Due to the high performance of SAP HANA, it is, however, now possible to segment your customer base a lot faster and even a couple of times a day. This means that you now have the chance to detect and to react to sudden, unexpected, short-term changes in customer behavior, changes that you would not even have been aware of before.

Risks for inexperienced users

The fact that big data often leads you into territories inhabited by new, hardly known but still very powerful, statistical creatures also harbors certain dangers. The cover of the SAP PRESS book *SAP HANA: An Introduction* (Berg and Silvia, 2014) shows a racing car. This might not only be a hint in terms of the speed of the appliance but also in terms of the demands on the driver. Even a novice driver will get ahead faster in a Formula-1 missile than by using a Chevrolet Spark, but although his average speed might go up, 750 horsepower under the hood might also dramatically reduce his life expectancy.

When it comes to big data, there are two substantial risks lurking for inexperienced users:

- **Statistics is all but trivial**
  In common textbooks about inferential statistics, you will find a lot of examples of how mathematically correct statistical results can lead to wrong conclusions. A key concept in this context are the so-called

spurious relationships. A statistical dependency between two variables A and B does not indicate a cause–effect relationship nor does it mean that A and B are related at all. Instead, A and B might both be related to a common (but yet undiscovered) variable C that creates the illusion of a dependency between A and B. Furthermore, a statistical dependency between A and B *could mean* A causes B; it could, however, also indicate that B causes A.

For all practical purposes, it is safe to assume that most of your competitors will step into one of these traps. Hence if you do not fall for spurious relationships, then you are already ahead of the rest.

- **Easy handling plus speed—masquerading as harmless**
  Quite a few big data–related software products (including SAP HANA) are sold on the promise of making even the most complicated statistical algorithms accessible to everybody at the push of a button. SAP claims that forecasting—thanks to drag and drop—has become a piece of cake with SAP Predictive Analysis powered by SAP HANA (*www.saphana.com/community/learn/solutions/predictive-analysis*). Well, maybe, and one could probably train a chimpanzee to use drag and drop functionalities. But when it comes to predicting our risk of getting cancer, we probably prefer being treated by a medical doctor who could—at least, theoretically—do all the required calculations using nothing but a pencil and a piece of paper. He wouldn't, of course, but it's good to know that he understands what's going on inside the machine. In the same way, every pilot flying a fully automated Airbus A380 has been trained on a rattly Piper Tomahawk (or the like), learning to find his destination and to land the plane without the help of satellite navigation, flight-management systems, or autopilots. Before trusting a flight captain or a first officer with the life of almost 1,000 passengers, an airline wants to be sure this person can still handle the plane if some or all of these systems fail.

[👁]

Big Data Can Create Major Damage

Using big data (successfully) assumes that you (or your company) are proficient in dealing with the required technologies, architectures, approaches, methods, and algorithms. If that is not the case, the resulting damage could become far bigger than your big data.

A new job profile: data scientist

Pavlo Baron's book *Big Data für IT-Entscheider* (*Big Data for IT Decision Makers*; Baron, Hanser Fachbuchverlag, 2013) provides you with a good overview of the kind of expertise (both functional and technical) you need to tame big data. The rise of big data has already led to the appearance of brand-new job profiles and training programs; one of these new jobs is that of a *data scientist*. Data scientists are interdisciplinary experts with a sound knowledge base in terms of system development and mathematics/statistics (mainly related to those areas listed in Section 1.1.2). Appropriate training programs are already in place at universities all over the world, ranging from the Winterthur-based Zürcher Hochschule für Angewandte Wissenschaften (Zurich School for Applied Sciences) in Switzerland to the Auckland-based University of Auckland in New Zealand (on the other side of the planet).

1.2.3 Realize, Decide, and—Above All—Act!

Knowing and willing are not enough

Some 200 years ago, German poet Johann Wolfang von Goethe documented a crucial discovery in respect to business intelligence:

> *Knowing is not enough; we must apply. Willing is not enough; we must do.*
>
> *(Wilhelm Meister's Journeyman Years, 1821/1829)*

[◉]

You Need to Act!

Big data can be useful if faster insights lead to faster decisions and faster decisions to faster actions. Neither insights nor decisions but only actions (by employees, suppliers, or customers) can create shareholder value.

Triggering buying decisions

Prosperous e-tailers make their living by detecting patterns in customers' buying behavior and by generating recommendations or advertising messages on that basis:

- Apple's iTunes store comes up with suggestions for music you might be interested in under a section titled "Listeners Also Bought."
- On Amazon, there are a number of recommendation categories, not only for books but across all their product categories ("Recommended for You," "Customers Who Bought This Item Also Bought," or "What

Other Items Do Customers Buy After Viewing This Item?"). Right after buying the latest tuning kit for your Golf GTI, you may also be offered the up-to-date schedule of fines.

Vendor and customer need to act

If Apple or Amazon knew what their customers wanted but did not use that knowledge to actually make tempting recommendations, that knowledge would be worthless; if customers never responded to such recommendations, Apple and Amazon would make no money with them. Do you see the point? It all depends on actions (Apple and Amazon need to act by making recommendations, and customers need to act by pressing the Buy buttons).

Acting now, not four weeks later

As well as coming up with proposals in the course of the purchasing process, Amazon could also send them to its customers by mail three weeks later. The buying incentive, however, is likely to be a lot higher if recommendations appear right after picking an article and even before completing the transaction and checking out; this is also the case because the customer is in the middle of the purchasing process, and not checking his email while on the run. Quite often, even seconds count. Just as the lady of the house has taken note of a recommendation and decided to buy another book, her baby might start to cry in the next room. Mom might still exercise the patience for *one-click buying* (a process patented by Amazon) but might get nervous if she had to go through 10 more data-entry and confirmation steps.

1.3 Where Are Benefits from Big Data to Be Found?

Processes that can benefit from big data

As well as *how* big data can generate shareholder value, you may also want to know *where* (that is, in which industries or business processes) you might find potential benefits. In Chapter 3 and the ensuing case studies, we are going to address this question in detail.

Nevertheless, we would still like to go ahead and at least give you a taste of the areas you may want to look at when it comes to digging for (big) data gold. We are going to do this for big data in general and independent of SAP HANA; these initial thoughts are like an *amuse-bouche*; they are meant to whet your appetite and pique your curiosity while preparing your (intellectual) taste buds for the more advanced considerations in Chapter 3 through Chapter 11.

1.3.1 Real Time versus Batch

Big data might also make sense in batch

In the big data space, there is a lot of talk about real time or near-real time. But the benefits big data can deliver are not limited to processes in which insights are immediately followed by decisions and actions. A good example for a batch environment is Google's indexing and ranking process. Although most users are expecting an immediate response on their search requests, they often don't care if websites are captured and the respective page rankings are updated overnight (apart from, for example, news websites). By properly separating the respective processes, Google ensure that news and tweets can be found immediately, whereas it might still take a while until their search engine becomes aware of rank-changing modifications on ordinary websites.

Page ranking in batch

So, if getting the results is not mega-urgent, why consider big data-like performance in batch processes at all? Well, did you ever try to read five trillion URLs from a persistent, relational database, then visit all these web addresses, process some 200 ranking-relevant parameters per URL, and write the ranking data back into your database? There are also batch applications that cannot be built without using methods (such as the concurrency of computations facilitated by Google's MapReduce programming model) and algorithms custom designed for dealing with giant amounts of data.

Machine learning in batch

Other examples can be found in training models that can learn from data. If Apple would like to improve Siri's speech-recognition performance based upon known mistakes Siri has made, such training activities can quite happily take place in batch, and there is no problem if they take hours, days, or weeks.

Nevertheless, quite a bit of computing power is required for this (consider the number of iPhone/iPad users and the amount of feedback they provide). Without the right kind of technology, methods (in the case of speech recognition, splitting spoken text into so-called N-grams), and algorithms, we would talk about runtimes of years, not days. The application would become useless.

[◉]

Do You Need Big Data?

So to recap: The three key questions which determine whether or not big data technologies, methods, architectures, or algorithms should be taken into account to support an existing or new business process are as follows:

- Would deciding and acting faster create tangible benefits?
- Do you need complex, performance-consuming algorithms for that?
- Can you provide the required performance (in terms of storage capacities, access times, goodput, FLOPS [floating-point operations per second], and so on) by classic means (for example, persistent, disk-based databases), or might some special tools from big data's toolbox come in handy?

Other than that, it doesn't really matter whether you use such algorithms in dialog or batch processes.

1.3.2 Improving Existing Business Processes

When considering application areas for big data, most businesses, quite understandably, think of existing business processes first. Big data can help optimize these in two ways: you may be able to make the process more efficient, or you may find ways to reduce response times within the process.

Making Existing Business Processes More Efficient

Learn more about your processes

In many cases, even just having new insights into existing processes can net you cash. There is a neat example from the US parcel delivery service UPS. Some time ago, UPS realized that making left turns in the United States (or in any country in which you drive on the right) is costly in terms of time and fuel. The same applies to right turns in countries in which you drive on the left. The driver will have to wait for oncoming traffic, and if the vehicle does not feature a start–stop system the engine keeps running during that time. After a couple of adaptions to navigation and route-planning algorithms, UPS trucks have systematically avoided routes with lots of left turns since about 2004. According to the company's own reports, these simple measures have saved the business about 10.5 million gallons of fuel and 186 years of waiting time at crossroads. Although UPS gained these insights quite a while before big data became a buzzword, the business case predominantly deals with optimizing routes for vehicles—a typical application area for big data today.

Thanks to dramatic advancements in locating people and objects from knowing more about their situations due to a variety of sensors and from transferring geospatial or sensor data by mobile means of communications, we now have a lot more status-related data at hand—much more than UPS could have dreamt of 10 years ago. It seems obvious that we can use all these data to reach new insights that were inaccessible in 2004. Back then we would have had neither the data nor the technologies to analyze them, regardless of whether we would have thought about batch or real-time analysis.

Some things are only possible in real time

Using status data from positioning or sensor systems, one could, for example, think of applications for optimization that could only work in real time. Consider the fact that UPS not only delivers parcels but also picks up parcels from its customers. At every given point in time, one could reevaluate the decision about which driver should deliver/pick-up which parcel, taking into account the position of all vehicles, the current traffic situation on their potential routes, the packages they already have on board, the amount of fuel remaining in each vehicle's tank, each vehicle's fuel efficiency on routes with a certain gradient, the latest list of pick-up requests, and so on; in theory this list is endless. The optimum answer to that question will probably change every ten minutes. Some trucks that were happily moving around before might now be stuck in a traffic jam. New orders have arrived, new parcels are waiting at the delivery centers, and one driver might have been involved in an accident.

Clearly, there are huge potential benefits here that a company could tap into. Although it might not make much sense to change a driver's routing every minute, sending out jobs on a one-by-one basis while always considering the latest state of play still needs a level of flexibility that batch optimization cannot deliver. Furthermore, when optimizing routes, UPS could even take into account *predicted,* instead of only actual, pick-up requests, using even more data (weekday, month, weather, and so on) and dynamically react to events that have not occurred yet. With classic, relatively static IT environments, such ideas would drive most IT managers to despair.

[◉]

Sometimes, processes do not have to change

Processes Stay the Same

When making existing business processes more efficient, the process as such (in UPS' case, picking up, shipping, and delivering parcels) stays very much the same. No matter whether or not a parcel-delivery service uses sophisticated tools, somebody or something (either a human or a system) still has to decide which driver is going to do what next.

Big data algorithms just help improve these decisions, either by combing through huge amounts of data after the fact and helping to learn more from past experience or by optimizing decisions in (near) real time.

Responding Faster with Existing Business Processes

Extreme example: high-frequency trading

In some cases, gaining new insights through batch analysis or improving the quality of decisions by using more data in better ways to support them is not enough. You may have already heard of *high-frequency trading* (HFT)—sometimes also called *algorithmic trading* or *algo trading*—in financial markets. Computer algorithms trade securities and carry out transactions in tiny fractions of a second. About half the volume traded on the electronic trading system Xetra, used by more than 14 stock exchanges around the world, is algo trading. With algorithmic trading, requirements in terms of processing speed are so high that extreme computing power alone is not sufficient; even the distance to the marketplace (or its servers) and the type of data transfer (for example, tethered or microwave link) can become decisive for the outcome of trading battles.

Financial markets impact most organizations

Data transfer times have become so important that billions are invested in intercontinental beeline cables or new directional radio links just to gain a couple of milliseconds or microseconds. Developers working in this segment would sneer at the performance data of solutions like SAP HANA. But even if your business model is not based upon speculating with shares, bonds, or derivatives, nowadays hardly any company working internationally can unplug itself from financial markets. Furthermore, the Euro crisis has shown that even states and governments seem to be unable to seal themselves off against adverse effects from financial markets. Exchange rates, commodity prices, and short-term interest rates can (among other factors also due to HFT) change dramatically in the blink of an eye.

In September 2011, the Swiss National Bank's decision to enforce a minimum exchange rate of 1.20 Swiss Francs (CHF) per Euro (EUR) with all necessary means made the EUR:CHF exchange rate jump instantaneously by more than 8 percent. Imagine what this could mean for a German-based company with an *operating margin* (margin divided by sales) of just 5 percent mainly selling into Switzerland. True, even SAP PAL is not going to help you foresee such political decisions (the philosopher and mathematician Nassim Nicholas Taleeb calls them *black swans*; we will get back to this term later), but the right kind of big data solution might still help you spot them sooner, assess their impact on your business and check alternative strategies in seconds, and initiate defensive measures faster than your competition (see also Chapter 4).

Reducing processing times

Let us take a closer look at response times. Response times consist of two components: *processing times* and *latency* or *waiting times* (except in the case of algo traders, we can ignore a third component, *data-transfer times*, for most practical purposes).

Simply speaking, processing time is the time during which the user is waiting for a system (for example, the time between pressing the [Enter] key and the appearance of a report on the screen); latency is the time during which a system is waiting for the user or—more generally—for an external event to occur. As shown in our exchange rate example, shortening processing times by just a couple of seconds, or even fractions of a second, can mean real money. And when it comes to user-friendliness, shorter processing times also lead to higher user acceptance.

Reducing latency

Latency is often a result of extreme processing times. If executing a complex analysis is taking hours, most users will not patiently wait for the result while fiercely staring at their screens. Instead, they will either leave the analysis to a batch job or return to the respective window every few hours, checking whether or not anything has happened. In the meantime, they will probably attend to other jobs. In both cases, they will most likely miss the exact moment at which the result finally comes up and hence not initiate decisions or actions straight away. If the result comes up after the end of their shift or if a batch process was used, then sometimes days pass before anybody attends to it.

If there is latency, then respective business processes are often designed to work around that restriction. The process is then no longer based on sensible design decisions but degenerates to kludge. As big data can help reduce response times (a) by reducing processing times and (b) by automating decisions and actions so that the user's attention will no longer be the bottleneck, latency can often be virtually eliminated.

[◉]

Reacting Faster to Unforeseen Events

(Near) real-time solutions with big data can help you spot risks in your business processes earlier, help you make better informed and well-founded decisions in the face of unexpected events/developments, and take counteractive measures faster or utilize chances more quickly.

1.3.3 Implementing New Business Processes

Sometimes, developing brand-new business processes or business models that would be unimaginable without big data is more exciting (and potentially more rewarding) than improving historically grown ones. In this context, we are mostly talking about real-time or near-real-time solutions that create new customer benefits or possibly even new markets.

Serving Customers Individually

Mass customization

More efficient production processes and new manufacturing technologies—such as so-called 3-D printers, devices that create components layer by layer from CAD data and thus print three-dimensional objects—make it possible to serve customer requirements more individually than ever before. Small and medium-sized enterprises have been the first to notice the trends toward mass customization, which is why companies like My Muesli and My Swiss Chocolate now allow you to buy your personalized muesli or your custom-made chocolate online.

More options = more planning effort

This gain brings pain; the more configurable variants of a product there are and the shorter their production cycles become, the tougher production and requirements planning are going to be. It's no coincidence that the first time SAP productively made available large data volumes in main memory (SAP liveCache technology) was with SAP Advanced Planning and Optimization (APO); even with long production cycles and highly standardized products, optimizing capacity planning is a very demanding task. The number of potential products and scenarios and sourcing options increases exponentially with the number of configurable characteristics. If customers expect their product to be available within next to no time, then we are dealing with a scenario that is as demanding and as challenging as route planning for picking up parcels.

Custom-designed products are nothing new. In ancient Rome, those who could afford it had their own personal togas made to measure; today, well-heeled gentlemen go to London's Saville Row to buy shirts, suits, or shoes that fit them to a T, or to Hamburg-based Lufthansa Technik to have their brand-new Airbuses turned into flying palaces that suit their personal tastes (or lack thereof). Even less loaded customers have now (within certain limits) become used to individually configuring their cars or computers online. In a world in which everything is becoming more similar and exchangeable, there seems to be a growing desire to stand out from the crowd; the fox brush on the car's antenna, the gray-haired fashion designer's or investment banker's ponytail, size-does-matter yachts, or stretched earlobes are all means to the same end.

Thanks to big data and real-time optimization, suppliers can now satisfy individual customer demands more cost efficiently. As the Internet lets end customers access global supply chains directly, tailor-made shirts from Hong Kong become affordable for everybody from online companies like Shirts My Way.

Revolutionizing user interfaces

Even if we are not focusing on new products or new customer groups, big data can radically change customer experiences. The boom in self-service processes at railways and airlines (travel information, ticket sales, check-in, etc.) has reduced the costs to many service providers, but such cost reductions come at the price of externalizing effort and frustration—that is, transferring work to their customers. Some of these customers are now starting to realize that they have become unpaid employees of these firms. You have to deal with badly programmed user interfaces on ticket machines, struggle through confusing websites when buying a railway ticket, and spend hours on hold while calling a service center instead of just queuing up at a counter.

Imagine you didn't have to do any of this: What if you could just ask your smartphone to take care of all that for you? Would you be prepared to pay your mobile phone provider a slightly higher fee if they were to offer such a service?

[◉]

Better Products and Services with Big Data

As our examples from mass customization and improved human–machine interaction show, big data can help you offer new products, tap into new customer segments with existing products and services, or serve existing customers in cost-efficient yet more user-friendly ways.

Record, Understand, Forecast, and Manage Customer Behavior

More data than ever before

Classic business solutions such as SAP ERP collect a lot of data about your customer's behavior. Who has ordered what and when? How often does a customer pay on time? Are there any patterns in customer complaints? Reporting such data in real time has been possible for quite some time (give or take minor delays owed to the *dispatcher*—the link between users and the system assigning dialog requests to work processes—or the queuing of update tasks); in the end, the "R" in SAP "R/3" always stood for "real time."

During the last couple of years, however, the bandwidth of data collected has continuously increased. Customer relationship management (covered by SAP CRM in the SAP world) not only deals with sales or delivery processes but also with related upstream activities (like marketing campaigns), collecting a lot more data than SAP's classic Sales and Distribution (SD) in SAP ERP. New communication channels have let the volume of these theoretically available data skyrocket (among these channels are the Internet as such, analysis tools such as Google Analytics, location data collected by smartphones, data from RFID-transponders or sensors, tracking and tracing data showing customer's travel paths in shops, and a lot more).

SAP ERP solutions (or extensions for them) may be able to collect, aggregate, and report such data, but in most cases they are not able to analyze them properly. When it comes to not only recording customer behavior in the broader sense but also to deconstructing and understanding it, other products (in many cases data warehouses and bespoke analysis software) are needed. Data from an ERP system are then periodically transferred to such data warehouses in batch and—at best—evaluated there with hindsight. What this means is that you may know how customers behaved in the past (for example, during the last four weeks or twelve months) but not how they are behaving now or are going to behave in the future.

Forecasts in classic data warehouses

Many of these data warehouses also come with simple built-in forecasting functionalities (with SAP, for example, SAP BW-IP features a planning function called forecasting). But apart from the fact that such functions only offer very basic statistical models (the planning function forecasting identifies trends using linear regression), they also usually only make sense under the premise that your customer's behavior can be extrapolated into the future.

This assumption will, however, only hold if you are dealing with boring buyers and also with dopey competitors who are not trying to pull the rug out from under your feet via spontaneous sales campaigns (as shown in the grill sauce example). In most cases, classic forecasting faces two challenges:

- If you want to generate competitive advantage, you will need forecasting models and insights that go well beyond what an average business student, having completed a course in basic statistics, can pull together using nothing but a pocket calculator.
- You have to brace yourself for the risk that models that worked perfectly for years can become obsolete overnight and that the well-established behaviors your customers are exhibiting can change in a flash.

The proof is in the pudding

With big data, you can go one step further with analyzing customer behavior. As you are able to interpret your customer's reactions in real time, you can figure out how they respond to which measures. If you were running an airline, you could offer fees that are 10 percent higher than usual to one percent of your passengers booking flights on your website. Because you can immediately see how that impacts demand, you don't have to estimate the price elasticity of demand or your customer's price sensitivity. Gone are the days of weird economic theories or highly paid external marketing consultants; instead, you can just determine what you need to know experimentally and incorporate the results in your pricing strategies straight away.

[◉]

New Ways to Interact with Your Customers

With big data, you can radically change the way you interact with your customers. Instead of offering something to them, then waiting for their reaction, and later trying to make some sense of it, you can now enter into a real-time dialogue with customers or other external parties (clients, employees, suppliers, etc.). In this way, you will be able to guide their behavior in the direction you desire; you will be able to steer them where you want them to go based upon predicted rather than historical patterns. This will help you get the desired results more quickly and hit more nails right on the head rather than relying on luck or trial and error.

1.4 How Benefits Turn Into Shareholder Value

Benefits are not enough

You now know what big data means and how/where big data solutions can help you create benefits. You also know a couple of key factors that influence whether these benefits will materialize or not. You may, however, rightly ask yourself whether mentioning such benefits would suffice when it comes to winning support from senior managers. Based upon our experience, in most organizations you would have to deliver something more concrete to get major implementation projects approved. Ultimately, you are trying to motivate others to put their careers on the line for your ideas.

Evaluating benefits

In Section 1.5, we will take a closer look at how business cases should be evaluated in a professional environment. One of the key building blocks in this context is the business case's value proposition; that value proposition is usually based upon the idea of shareholder value, which is why we will first have a closer look at this term and at how you can create shareholder value with big data. You may know the keyword *shareholder value* from the news and background stories cursing corporate raiders. Leaving out politics and ideologies, the idea of shareholder value is built upon a fascinating construct of ideas.

Target audience: technical experts

If you're reading this book as an employee of your company's finance department or with a business background, concepts like shareholder value or value drivers will sound familiar to you, in which case you might just want to skip this section. Our thoughts regarding shareholder value are mainly addressed to readers who don't have a commercial or economic background—for example, IT experts, natural scientists, or engineers—but even if you know what value drivers are and how they are linked to shareholder value you might want to refresh your knowledge. Maybe you'll find one or two new ideas in our discussions.

Have you ever considered why financial markets often euphorically celebrate measures that cause long-term damage to a company? One reason for that (which we're going to have a closer look at in the following section) is that shareholder value is not based on the longer-term financial benefits of what a company does or refrains from doing; the only thing that counts is the shareholders' current perception.

1.4.1 The Concept of Shareholder Value

Focusing on shareholders

In classic economic theory, a company's sole reason for existence is to make money for its shareholders. This money usually finds its way into the shareholders' pockets in two forms: as higher share prices or as dividends. Hence, a project should *only* be undertaken if it would lead to higher share prices or higher dividends (now or in the future)—that is, if it generates shareholder value.

Likewise, *every* project that does create shareholder value should be undertaken; financial and other means aren't limiting factors within this theoretical approach. Simply speaking, the company could always borrow money for projects that make financial sense and use these funds to acquire whatever resources it may need to execute that project.

[«]

Shareholder Value

Shareholder value for a public company is exactly what it sounds like: the shareholders' valuation of their shares, that is, the total market value of the company's share capital. For privately owned companies, a very similar concept can be applied; the only difference is that there is no market on which shares are evaluated. Instead, shareholder value represents the total wealth that accrues/will accrue in terms of returns, dividends, or other capital gains.

Shareholders' perceptions and expectations

In the case of both public and private companies, the perception and the expectations of investors play a major role. Nobody knows how much the shares of a public company are going to be worth tomorrow nor what kind of dividends it is going to pay in the future. Investors are—

like most of us—not omniscient. They make their decisions based upon subjective perceptions, their resulting expectations, and their personal preferences. With privately owned companies, the fact that there is no transparent market makes things harder. Individual investors don't have a published market value that they can use as a yardstick for their personal assessments.

Key performance indicators

For government bodies or nonprofit organizations, shareholder value can be replaced by their political or social value to the community that they serve or their achievement against targets set for them. The only modification required is exchanging the term *value drivers* (see Section 1.4.2) for *key performance indicators* (*KPIs*).

Such KPIs are often not related to financial targets. Therefore, judging whether an investment of *x* is justified to accomplish an improvement of *y* with a KPI is usually more difficult than in a business environment. Deciding, for example, whether or not reducing the number of school dropouts or teenage moms by 10 percent (the improvement *y*) would be worth 10 million dollars (investment *x*) is a purely political matter.

Because the tools to be applied are nevertheless similar, whenever we talk about shareholder value we also want you to mentally include the term *political/social value*. Each time we talk about value drivers, also think of KPIs in a noncommercial environment or even stakeholder value (if applicable and measurable in your specific context). When we say *business/company*, this should also embrace government bodies and nonprofit organizations.

Shareholder value is a double-edged sword

Some of you might argue that shareholder value is much too narrow a concept for measuring human happiness (very much like gross domestic product [GDP] for the economy as a whole). Instead, not only government bodies but also companies operating on a commercial basis should take into account the well-being of all stakeholders and not just focus purely on their shareholders. There are also claims that the short-sightedness of shareholder value has been responsible for excessive risk taking and thus many of the recent economic crises. Indeed, saving money on sewage processing and only doing the legally required minimum to protect the environment (or working on having maximum permissible values increased via lobbying) might lead to higher dividends while at the same time depriving the local population of its main source of drinking water.

Shareholder value is still very common

Nonetheless, shareholder value still persists as the main criterion for evaluating projects or investment options within a commercial environment. Under many legislations, management is even legally obliged to focus on the interests of the shareholders, though not necessarily on shareholder value only. A Frankfurt-based CEO who decides to spend a few hundred million to cater to the interests of local farmers and fishermen in Nigeria might soon find that he is treading a very fine line separating responsible behavior from a five-year prison sentence under Section 266 ("Breach of Trust") of the German criminal code. For the purposes of this book, we will therefore assume that everything that creates shareholder value is good and everything that reduces it is bad. The ethical discussion about the concept of shareholder value (as well as the one regarding the moral aspects of big data) may take place where it belongs: in the wider public and in political forums.

1.4.2 Value Drivers

What creates shareholder value?

But, how on earth are you going to know (in advance!) whether a certain activity will make a company's share price or dividend payments explode or go down south, now or in the future? And how do you know what the share price or dividends would be without the project?

Well, the truth is that you can't know any of these things. However, what you can still do despite this lack of knowledge is look for factors or parameters (so-called value drivers) that may be easier to predict or to influence than shareholder value and for whose impact on shareholder value you have some kind of statistical evidence (for example, correlations).

[«]

Correlation

A *correlation* is a link between two measures. A correlation between characteristics or parameters indicates that these depend on each other in some shape or form. It is irrelevant what the reason for this interdependence might be or whether or not we know anything about it. It doesn't matter, for example, whether there is a cause-effect relationship between the respective measures or whether they depend on a third measure to keep them aligned.

Old proverbs are good examples of correlations with an unclear mechanism of action. Century-old, sometimes date-related climate rules are surprisingly accurate when it comes to forecasting mid-term trends; nevertheless, the cause–effect relationship between the population's observations and subsequent weather events often remains a mystery. Only in a few cases (for example, with the weather: "red sky at night, shepherd's delight") do scientists understand the basis of the correlation.

Correlation ≠ cause-effect relationship

No matter how high the correlation between a value driver and shareholder value might be, it does not, by the way, indicate any kind of cause–effect relationship between the two of them.

[Ex]

Beware of spurious relationships!

An Unusual Value Driver

There could, for example, be a strong correlation between your CEO's body height and the share price of your company. If so, you wouldn't have to worry about *why* that might be the case. You would just keep putting former players from the LA Lakers onto your board, focusing on size and not worrying about useless decorum such as academic titles or job experience.

What you would need to be really sure of, however, is that there *is* such a kind of relationship, even if it is just founded on unconscious behavioral patterns of your shareholders. Keep in mind that there are statistical mirages out there! And remember that your shareholders might like tall CEOs but at the same time don't want inexperienced board members (in which case you would have to deal with two separate value drivers that might work in opposite directions).

Although shareholder value represents your objective, value drivers are operational parameters that—like a compass—help you move in the right direction (that is, toward your objective). This view is a bit more generic than the classic idea of shareholder value; it therefore applies equally well to government organizations in which both value drivers and objectives could be internal parameters that are not related to financial figures.

Value drivers interpreted broad-mindedly: example

A simple example might help explain the universality of this concept. You might have a personal objective, such as reducing your blood pressure. Most of us cannot manage our blood pressure by sheer will power. Hence, reducing it can only be done indirectly by reducing your salt intake, taking a brisk walk every morning, remembering to take your beta-blockers, even sprinkling two tablespoons of flaxseed on your yogurt in the morning (highly recommended if you like that sort of thing). If reducing your blood pressure would result in increasing your shareholder value (with yourself being the only shareholder), then you may work with the following value drivers:

- The amount of (visible and hidden) salt you have each day
- The number of training units you do per day
- Whether or not you take your daily dose of medicine
- The amount of flaxseed you have on your yogurt in the morning

Three additional remarks:

Value drivers can come along in different forms

- You may have noticed that we used different types of value drivers in this example. With the training units, we used something that can be measured, and we are assuming that more is better than less. With the medicine, we are instead introducing a binary (or *dichotomous*) value driver that could either be "yes" or "no" but not "sometimes," as it would probably not be a good idea to change your dose on a daily basis without consulting your doctor. These different types of value drivers (or so-called *levels of measurement*) are explained in a bit more detail in Section 1.4.3.
- As with the body height of your CEO, the cause–effect relationship between the value driver and shareholder value does not have to be evident. Your blood pressure pills could be placebos, in which case it would be very difficult to find or understand the underlying cause–effect relationship. Nevertheless, taking them or not could still be a valid value driver as long as there is a proven (high) correlation between taking them and your blood pressure.
- Some value drivers are tangible (the amount of salt), some are intangible (your training units), but all are observable (though not necessarily measurable), either objectively or subjectively.

Value drivers versus cost drivers

Other business concepts (such as *activity-based costing* [ABC]) use parameters that have similar names (such as cost drivers), which—at first glance—seem to be related to value drivers. It is, therefore, important to recognize the difference between cost drivers and (cost-related) value drivers.

Tightly managing costs does not always lead to an increase in shareholder value; value drivers instead (assuming that you are working with the right ones and haven't forgotten any) have a direct effect on shareholder value. We once worked for a manufacturer of IT hardware; the company was in decline and became so focused on cost drivers that the staff employed in the accounts department to manage cost eventually outnumbered the direct and indirect sales staff. The company folded.

Therefore, it is value drivers rather than revenues or costs that we are going to primarily consider in more depth when evaluating the business cases presented in this book. Although some of these value drivers might be linked to costs and hence to cost drivers, the concept of shareholder value and value drivers tends to be more holistic than focusing on costs only; high costs without any corresponding value received could often reduce shareholder value.

Value drivers are often industry/ company specific

Clearly, such value drivers are very specific to each organization; you will have to find out which ones are key in your environment. We will try our best to provide a number of generic and business-case-specific examples to give you an idea of what kind of prey you might have to chase. We also provide you with a sample value driver database as an add-on to this book (see *www.sap-press.com/3647*). You may use this database to spark your own creativity or modify it to meet your specific requirements. If you tend to suffer from high blood pressure during SAP go-live phases, you may even use it for your own personal purposes.

Positive and negative effects of IT projects

As you can imagine, many projects will not just have a purely positive impact on shareholder value. Most business ventures come at a price. A project may reduce your liquid assets, deteriorate your debt to equity ratio or your reputation, increase the number of employees leaving the firm, or so on. In the end, management will have to assess whether the positive or negative impacts will prevail and what the net effect on shareholder value might be. Such an analysis might involve simple statistical tools (for example, multiple linear regression) or highly complicated models and simulations (possibly using SAP Business Planning and Consolidation [BPC]). Sometimes it might not even be possible to convert numbers related to value drivers into amounts of money; in this case, management would have to weigh positive and negative effects without being able to calculate a net effect on shareholder value.

1.4.3 How to Identify Value Drivers

Nine characteristics of value drivers

Although many value drivers are very specific to each and every organization, there are still some clues we can provide you with to help you recognize a value driver if you run into one. In principle, a great variety of factors might have an impact on shareholder value. There are, however, a number of characteristics that might help you separate the wheat from the chaff. As mentioned before, the same thoughts are, in general, equally valid for private or nonprofit organizations. These characteristics include the following:

- **Continuously observable for an extended period**
  You should only consider parameters as value drivers that can be observed or measured more or less continuously and over a sufficiently long period of time. If a factor cannot be monitored, it would be impossible for you to know whether it changed when there was a shift in shareholder value.

- **Not necessarily on ratio scale**
  Value drivers don't necessarily have to be measures on a ratio scale. Even ordinal data can be handled, and there are statistical tools as well for very simple dichotomous (0/1 or yes/no) value drivers (such as the point-biserial correlation coefficient).

  In terms of dichotomous value drivers, just remember the preceding example regarding blood pressure and the question of whether you took your pills or not.

- **Accessible internally and externally**
  Usually, value drivers are parameters that can be tracked by people within and outside of the organization. Unless a huge number of shareholders are also company insiders, it would be hard to imagine that something the market cannot even perceive would still change market participants' behavior regarding a company's shares. This doesn't necessarily mean that players on the market always have to be consciously aware of what's driving their behavior. (Remember the remark about placebos in Section 1.4.2?)

  An exception to the rule that value drivers should be visible to outsiders are parameters that might not be observable to shareholders but that still have an impact on other factors that are observable. Shareholders might know nothing about the room temperature in an organization's office building; nevertheless, air conditioning could still have an effect on employee turnover, which in turn would then change shareholder value.

The following item explains why—in this specific example—room temperature might still be a better value driver than the number of people leaving.

- **Significant for the shareholders**
  Value drivers don't *really* have to have a positive or negative effect on a company's value. It is good enough if they are *perceived* as such (which is grist for the mills of those claiming that shareholder value is short-sighted and superficial).
- **Manageable**
  When looking for value drivers, you should look for factors that you can control and manage. An organization can do very little to directly modify employee turnover (very much like the fact that you can't directly change your blood pressure). It may, however, be able to work on things that have an impact on employee satisfaction (room temperature, remuneration schemes, flextime rules, or the menu choices in the cafeteria).

  This is why it would make more sense to evaluate projects based upon their contribution to these aspects rather than trying to figure out whether or how a certain project could reduce the number of employees giving notice.
- **Stable over time**
  Value drivers and their impact on shareholder value should be stable over time. It doesn't help you to know that positive reports on a specific blog have had a positive effect on the company's shareholder value if that blog will be offline next year. If the choice of menus in your cafeteria is subject to weekly changes, you cannot judge whether this will have an impact on the mid-term happiness of your employees nor on the percentage of staff resigning per month.

  You need clear and sufficient evidence to convict suspected value drivers. Because you need to verify whether value driver candidates really have an impact on shareholder value (using statistical tests), you want a sufficient amount of historical data (that is, a sample that is large enough). During the period covered by your sample, neither any relevant frame conditions nor the behavior of your value driver should change (which is sometimes hard to tell and even harder to ensure). From a mathematical/statistical perspective, this property is called *stationarity*. Sometimes, stationarity can be accomplished by transforming data first (for example, by removing seasonality).

- **High explanatory power**
  Your collection of value drivers should have reasonable *explanatory power* throughout the period you are considering, meaning that value driver changes substantially contribute to understanding/predicting changes in shareholder value (which in no way implies there is some kind of cause–effect relationship!). Instead, we are talking about the level of dependency between a value driver and shareholder value.

  Menus and room temperature might have an impact on employee turnover (and thus on shareholder value), but their combined effect might be negligible if the amount of money your main competitor is spending on headhunters luring people out of your organization has been quadrupled recently; in that case, you may have to consider recruiting expenses as a value driver instead, as their explanatory power might be higher than the combined explanatory power of menus and room temperature.

- **Statistical relationship fully understood**
  Don't get tunnel vision because you are trying to detect cause–effect relationships. More importantly, you need to be convinced of, or understand, the following:

  - Your shareholder value is statistically dependent on your value driver. *Statistically dependent* means that changes with the value driver go hand in hand with changes of shareholder value but does not mean there is any kind of cause–effect relationship.

  - The dependency needs to be strong—that is, have a higher explanatory power than competing or supplementary value drivers. Do changes with the value driver explain almost 100 percent of all changes in shareholder value, or are they just one small factor among many others?

  - How does shareholder value change if the value driver goes up or down? Does shareholder value move in the same direction? Does it react linearly or exponentially?

- **Beware of spurious relationships, correlations, or regressions**
  On the one hand, don't restrict your search for value drivers to easily understandable cause–effect relationships. On the other hand, don't let statistical effects, such as spurious relationships, spurious correlations, or spurious regressions, lead you astray. Spurious relationships might seduce you into believing that two parameters are linked, whereas in reality they might be indirectly connected by a third parameter (as mentioned in Section 1.2.2).

  A classic example in statistics is the high but misleading correlation between the number of storks nesting in certain regions and the number of babies delivered (by women, that is) in these areas. There are even websites listing examples of such phenomena! A politician trying to encourage families to have more children by reintroducing storks to the center of Berlin will probably fail (not just because in a megacity many of these storks might end up as roadkill). Instead, making it easier for young families to live in rural areas rather than in big cities (by providing appropriate jobs) might have the desired effect. Why? Well, one underlying factor for both figures in the preceding example is the rurality of the region (others are the time of the year and the weather conditions nine months before the survey).

  More examples of spurious correlations can be found at *www.tylervigen.com.* Perhaps one of them explains why my grandmother told me that bad dreams were caused by eating cheese before I went to bed!

Your own specific value driver repository

Organizations make substantial efforts to learn more about their specific value drivers. Major consulting firms (such as Bain & Company, Horváth & Partners, McKinsey & Company, etc.) and even SAP itself (via its global Business Transformation Services [BTS] teams) are offering cross-company, industry-specific expertise in that area.

When trying to identify your relevant value drivers yourself, you could proceed based upon the following task list:

- **Use existing value driver repositories**
  Before starting to invest substantial amounts of time trying to figure out which value drivers are relevant in the context of your project, first check for some kind of value driver repository within your organization. Not only will using value drivers that have already been

identified save you time, but also using parameters that have already been reviewed and approved by management will probably make it easier to find a sponsor for your project and to overcome internal resistance.

- **Consider the benefits you are after**
  If no value drivers have been identified yet, or if those that are around are not affected by the SAP HANA solution you are planning to implement, think about the benefits of your project. What are you planning to deliver and via which potential value drivers is your project supposed to have a positive impact on shareholder value? What are the negative effects of your plan (every project costs something; you will at the very least have to account for the time you are spending thinking about your business case), and via which potential value drivers are these negative effects going to reduce shareholder value?
- **Check your value drivers**
  Check your value driver candidates against the nine characteristics at the beginning of this section and confirm that they really are value drivers. If you go through all the steps on the list, then you will not only have the results of the respective statistical tests measuring, for example, their explanatory power; as you go along, you will also produce a proper mathematical model of the relationships between your value drivers and the shareholder value of your company.

Checking value drivers is not trivial

One more caveat: statistical testing of value drivers is not trivial. You will need substantial knowledge of descriptive and inferential statistics and of the pitfalls that may await you when applying the related tools; in terms of the effort required you could easily end up with a project in its own right. In many cases, companies may want to buy professional services from companies who specialize in this area. On top of that, you may also want to discuss the topic with us at our Facebook page (*https://www.facebook.com/saphana.makingthecase*).

The good news is that if you already have access to an SAP HANA box, you can use its performance plus the analytical capabilities of SAP PAL or the statistical programming language R for your research.

An existing SAP HANA implementation might also come in handy when you are trying to come up with value drivers for other big data business cases. Furthermore, the system architecture needed here very much resembles the proposals we are making in some of the case studies. In the end, it doesn't matter whether you are trying to detect dependencies among—for example—certain key figures and value drivers (as will be seen in Chapter 6) or between value drivers and shareholder value.

Generic value drivers

Apart from the value drivers that we are going to discuss in the course of our business cases, we have promised (toward the end of Section 1.4.2) to mention a few generic value drivers that play a role in many big data projects. These generic value drivers are discussed in the following three subsections.

Reducing Expenditures, Increasing Income

Two factors that play a major role in evaluating each and every IT project are *expenditures* and *income*. On the one hand, IT projects often immediately lead to expenditures (and sometimes even to income); on the other hand, the purpose of such projects usually lies in longer-term reductions to expenditures or increases of income.

[»]

Expenditures and Income

At this point, we are taking the liberty of ignoring some subtleties of financial accounting, treating expenditure/costs/expenses/outgoing payments and income/revenues/earnings/incoming payments as synonyms. For the purpose of evaluating a business case, it does not really matter whether the project might, for example, result in higher operating or financial income. The only thing we are interested in is shareholder value.

Depending on whether we are talking about profit-oriented or nonprofit organizations, expenditures and income might be more or less important.

Expenditures and income are not value drivers themselves

Also, expenditures and income are not value drivers themselves but rather categories of value drivers. Value drivers need to be more specific. Some examples for expenditure- or income-related value drivers include the following:

- Costs for TV advertising at certain times on a certain station
- Expenses for sponsoring a sports team for a season
- Development costs for product design for a new product

- The margins of individual product groups of products
- The revenues of all your branches within a certain town or country
- The price customers are willing to pay for one of your products

Reducing Uncertainty, Speeding Up Cash Inflow, Slowing Down Cash Outflow

Just like my grandmother, yours might have taught you that a bird in the hand is worth two in the bush. Being born in 1900 and having gone through two world wars plus the severe economic downturns before and after them, she probably meant that one should not be greedy but rather should be humble minded and content with what one has. But apart from environmental conditioning (leading to decisions based on experience rather than on reasoning or deduction), there are also good logical reasons why many companies and individuals would share her view. How many birds in the hand one would trade for those in the bush varies and reflects a person's or an organization's individual attitude toward risk. Nevertheless, the basic statement probably applies to all of us (with the exception of gamblers).

Risk and time preference with individuals

So, why would a bird in the hand be worth more than ones in the bush?

- The birds in the bush might fly away as soon as you let the one in your hand go, leaving you without lunch for today or even for the whole week. Usually, we prefer secure options over risky ones—in this example illustrated by the fact that we would usually let two uncertain birds go for one we can be sure of.
- Furthermore, the bird in your hand could be turned into a meal without any major delay, whereas it might still take you an hour or two just to catch the two in the bush. Most of us tend to prefer the immediate satisfaction of a need to the promised satisfaction of a deferred one (the decline of the home-cooked meal and surge of ready-to-serve ones is a striking proof).

An organization's preferences

In a business environment, these insights can be phrased as follows:

- In general, people and organizations prefer less risky options to more risky ones.

- An individual's or an organization's attitude toward risk (also called *risk aversion* or *risk love*, respectively) can be measured by the number of birds in the bush it would trade for one bird in its hand. By the way, once you really get into it, measuring attitudes toward risk with people and organizations becomes tricky and inconsistent.
- Furthermore, people and organizations also tend to prefer one dollar today to one dollar tomorrow. One way to measure an individual's or organization's *time preference*—that is, the preference for getting money sooner rather than later—would be to look at the interest it would want for a safe investment in a zero-inflation environment (a "safe investment" because we would want to eliminate the effect risk would have on expected rates of return and "zero-inflation" to eliminate the compensation one would want to balance the loss of purchasing power over time).

Risk, time preference, and shareholder value

Therefore, if shareholders behave like the rest of us, they should be willing to pay more for shares of companies that reduce their risk exposure, speed up cash inflow, and delay cash outflow, thus increasing the shareholder value of these companies. The link between risk/inflow and shareholder value is as follows:

- **Less risk = higher shareholder value**
  Reducing the probability of adverse events or increasing the probability of beneficial ones creates shareholder value. Investors looking for investment opportunities would give up more birds in their hands (i.e., cash) per bird in the bush (i.e., shares) if you made it more difficult for the birds in the bush to escape. The result: increasing share prices and happy shareholders. (Unless, as noted before, potential buyers are addicted to gambling and therefore have an unnatural appetite for risk just for risk's sake.)
- **Faster inflow = higher shareholder value**
  Bringing cash inflows closer to "now" or moving cash outflows further into the future creates shareholder value. Investors looking for investment opportunities would also give up more birds in their hands (i.e., cash) per bird in the bush (i.e., shares) if you developed a new bird-catching and processing technique that would reduce the time required to convert the ones in the bush into lunch by 50 percent. The result would once again be rising share prices and satisfied

shareholders (again: unless potential buyers have a masochistic desire to stay hungry longer than necessary).

Big data impacts risk and speed

We are emphasizing these facts because reducing uncertainty or speeding up revenue generation are typical potential benefits of big data solutions like SAP HANA. If only my grandmother had lived long enough to see that her simple worldly wisdom is becoming a key driver behind big data projects.

The Shareholders' Perceptions, Expectations, and Preferences

Investors are not omniscient

Although Goldman Sachs' Lloyd Blankfein once claimed that bankers "do God's work," even he is probably not all-knowing. When defining shareholder value, we have already suggested that investors have to be omniscient to know the *true* value of a company's shares. In the end, to properly evaluate a business, one would need all relevant data and facts now and in the future, and "relevant" would also include all future political decisions, market data, and the secret thoughts of each individual investor (such as his present and future preferences).

[Ex]

Facts Relevant for Valuation

Almost everything can have an effect on shareholder value: the political environment, love affairs, and superstition, to name just a few:

- If a government decides to raise higher taxes on dividends and at the same time to no longer tax capital gains, companies who pay no dividends at all but whose share prices are expected to double over the next couple of years become more attractive.
- If a major shareholder falls newly in love and is therefore planning to get divorced from his wife, he may suddenly develop a strange appetite for securities whose value could decline in the short run but rise in the long term (keywords in this context are alimony and compensation).
- German airline Lufthansa's planes—out of respect for frightened passengers—do not accommodate seat rows 13 and 17 (17 is considered an unlucky number in Brazil and Italy). If an airline is able to improve their passenger load factors this way (and if resulting gains are higher than extra costs in system development), not just superstitious investors might reach for their shares.

In all three cases, there is going to be an effect on supply and demand with the company's shares and therefore an impact on shareholder value. This also means that shareholders don't need to be omniscient for the concept of shareholder value to work.

Subjectivity counts

In the examples discussed in the above box, shareholder value is based upon nothing but perceptions, expectations, and preferences. It does not matter one iota whether there is a link between such perceptions, expectations, or preferences and reality, which is why the question of whether the planes of an airline come with or without row 13 could be a valid (dichotomous) value driver.

Other examples for less exotic but still nonmonetary value drivers include the following:

- Durability of your products
- Level of customer satisfaction
- Attractiveness of your company as a potential employer

Such examples also demonstrate that not only shareholders influence shareholder value; the views of your customers, suppliers, and employees could also affect the valuation of your company. The key difference is that other stakeholders' views are not directly linked to share supply and demand but affect share prices via their effects on shareholders' perceptions, expectations, and preferences.

In the same way, high customer satisfaction levels only make a difference if your shareholders are aware of them and are concerned about customer happiness. If not, you may well be better off moving production to the cheapest possible location and not caring about product quality and customer feedback. This would, however, only work as long as customer satisfaction does not affect other value drivers that shareholders *do* care about—for example, rebuy rates.

1.5 Evaluating Business Cases

Sixteen ways to generate benefit via big data

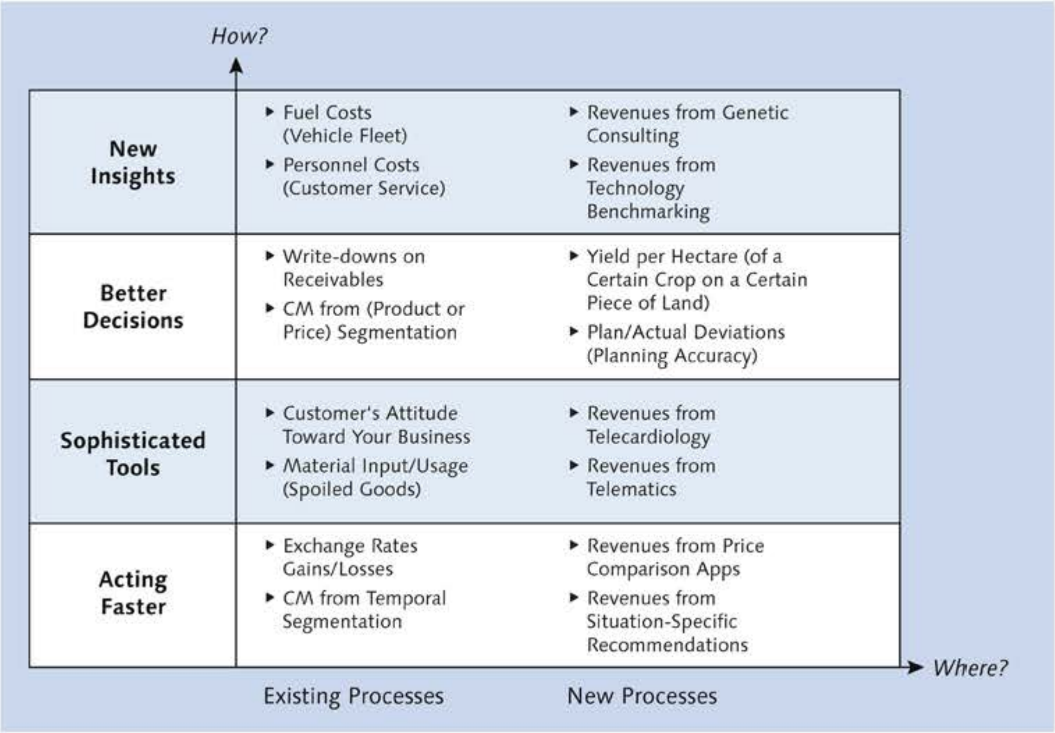
The matrix shown in Figure 1.2 summarizes our insights from Section 1.2 and Section 1.3 and lists a few more sample value drivers (CM stands for *contribution margin*).

The matrix is to be read as follows:

- In Section 1.2, we explained that big data can create benefits by helping you to gain new insights, make better decisions, use sophisticated tools (that needed too much computing power before), and act faster than before. These four different ways to exploit big data are shown on Figure 1.2's vertical axis (*How?*).

- In Section 1.3, we mentioned that you can either modify *existing* business processes or create entirely *new* ones (new business processes or even new business models) that did not exist before because they could not be supported without big data. These two characteristic values sit on Figure 1.2's horizontal axis (*Where?*).

This gives eight different combinations of how and where—for example, New Insights with Existing Processes or Sophisticated Tools with New Processes. Each cell of the matrix in Figure 1.2 stands for one of these eight combinations. For each of these eight cells, you can probably think of a wide array of potential business or use cases, each one of which would (hopefully) have a positive effect on shareholder value via different value drivers.



	Existing Processes	New Processes
New Insights	▸ Fuel Costs (Vehicle Fleet) ▸ Personnel Costs (Customer Service)	▸ Revenues from Genetic Consulting ▸ Revenues from Technology Benchmarking
Better Decisions	▸ Write-downs on Receivables ▸ CM from (Product or Price) Segmentation	▸ Yield per Hectare (of a Certain Crop on a Certain Piece of Land) ▸ Plan/Actual Deviations (Planning Accuracy)
Sophisticated Tools	▸ Customer's Attitude Toward Your Business ▸ Material Input/Usage (Spoiled Goods)	▸ Revenues from Telecardiology ▸ Revenues from Telematics
Acting Faster	▸ Exchange Rates Gains/Losses ▸ CM from Temporal Segmentation	▸ Revenues from Price Comparison Apps ▸ Revenues from Situation-Specific Recommendations

Figure 1.2 Benefit–Value Driver Matrix

In Figure 1.2, we have put two sample value drivers into each cell (for example WRITE-DOWNS ON RECEIVABLES or REVENUES FROM TELEMATICS). Every single one of these 16 value drivers represents one scenario of how to use big data applications. In conjunction with the case studies in Chapter 4 through Chapter 11, they are meant to serve as an inspiration for developing your own ideas.

Using the matrix

You can use the matrix in Figure 1.2 in two different ways:

- **Brainstorming**
  Use the matrix as a starting point for a morphological analysis. You could, for example, further amend or break down the two sample dimensions (how and where) that we came up with.

  Put business or use cases into the matrix's cells; if your dimensions are practical/observable and your business/use cases are sufficiently specific, then finding related value drivers will be a breeze.

- **Evaluating business cases**
  For a specific business case within your environment, just locate its position within the matrix determining how and where you are expecting big data to make a difference. The matrix will then help you come up with appropriate value drivers for that business case. For each of our case studies, we will decide on a couple of value drivers and put them into the matrix shown in Figure 1.2.

[»]

Even oddball solutions are worth exploring

Morphological Analysis

Morphological analysis (sometimes also called *Zwicky box*) is a creativity method invented by the Swiss astrophysicist Fritz Zwicky (1898–1974). Morphological analysis helps you solve complex problems by exploring all possible solutions to them (regardless of whether these solutions seem to make sense or not). Just proceed as follows:

1. If, for example, you are trying to develop new business cases for big data, start with collecting dimensions/characteristics/attributes of such business cases. We have used two dimensions (how and where), but you can certainly think of many more (for example: "involved partner roles," "affected products/services," "affected functional areas [production, administration, sales, etc.]," and so on).

   Try to make sure that your dimensions are independent of each other (which is probably not always the case for our benefit–value driver matrix in Figure 1.2);

if possible, they should also be practical and implementable, meaning you should have an idea of what they refer to in terms of your environment and processes. In idea-generation workshops with customers, we are usually able to come up with 10 to 20 nonoverlapping dimensions without major effort.

2. Next, list all characteristic values you can think of for each of the dimensions/characteristics. For the involved partner roles dimension, this could, for example, mean customer, supplier, employee, or bank.
3. Then select one characteristic value for each dimension (that is, select a cell within your multidimensional matrix) and let the resulting combination (for example, "acting faster in existing business processes that are related to suppliers") sink in for a while; try to think of specific processes or business cases that could be described by that combination. Quite often, there is more than one resulting scenario for each set of characteristic values.

Morphological analysis is especially useful for brainstorming in meetings and with teams.

At *www.mindtools.com/pages/article/newCT_03.htm* you will find an example of how morphological analysis can generate new ideas that nobody had thought of so far. As you are putting down all possible values for all characteristics, morphological analysis also takes into account combinations you might never have thought of in the first instance.

Sample value drivers

As mentioned previously, we have put 16 sample value drivers into Figure 1.2. Next, we would like to explain the scenarios we thought of and the respective generic value drivers (as defined in Section 1.4.3).

Fuel Costs (Vehicle Fleet)

Driving behavior and driving environment

Imagine a parcel delivery company (like UPS) that not only collects very detailed positioning and route-tracking data but also—from a variety of sensors built into their vehicles—data regarding driving behavior and related factors (acceleration/deceleration, road conditions, fuel consumption, traffic situation, weather, and so on). These data are analyzed, searching for ways to minimize the vehicle fleet's fuel consumption. Total fuel costs go down and shareholder value increases (generic value driver: reducing expenses).

Personnel Costs (Customer Service)

Data collected on (customer) premises

A company repairing household appliances collects data about the jobs performed by their service technicians. This not only includes geospatial data but also data about customers visited, service orders, devices worked on, causes of problems, spare parts stock in their vehicles, revisits, and so on. Such data can be used to find ways to handle the same number of jobs with fewer technicians. This would reduce personnel costs, which in return would have a positive effect on shareholder value (generic value driver: reducing expenses).

Write-Downs on Receivables

Real-time credit-rating data

If more (relevant) data about a customer's current and predicted future creditworthiness are available in real time, then sales agents can make better decisions about payment terms (like down payments) in the course of the buying process. This can help reduce losses from bad debt, which again increases shareholder value (generic value drivers: reducing expenses, reducing uncertainty, and speeding up cash inflow).

Contribution Margin from (Product or Price) Segmentation

Businesses used to divide customers or markets into segments to sell segment-specific products at segment-specific prices. We are going to talk about this in more detail in Chapter 7 (generic value driver: increasing income).

Customer's Attitudes toward Your Business

Text mining

Text mining and sentiment detection enable you to find out what customers think about certain products, or your company as a whole, plus how these views are changed by certain types of information (for example, reports about working conditions with your suppliers in Bangladesh). The positive or negative views of existing and potential customers will have an effect on shareholder value (generic value drivers: perceptions, expectations, and preferences not only with shareholders but first of all with customers and then indirectly with shareholders).

Material Input/Usage (Spoiled Goods)

Monitoring geospatial and sensor data

With food processing and agricultural and chemical products, transport and storage conditions and product lifespan are of the utmost importance. Kept in a warehouse for an extended period of time and at low humidity, raw tobacco will lose volume and weight—which equals money from a cigarette manufacturer's perspective. If a variety of geospatial and sensor data (temperature, humidity, vibration, and so on) can be continuously monitored and analyzed, then the risk that input materials or semifinished or finished goods could suffer from damage can be reduced substantially. This again reduces material usage and costs, which will have a positive effect on shareholder value (generic value driver: reducing expenditures).

Exchange Rate Gains/Losses

Chapter 4 will tell you more about how monitoring exchange rates in real-time can help you generate additional financial income or reduce related losses (generic value drivers: reducing expenses and reducing uncertainty).

Contribution Margin from Temporal Segmentation

If you are not only able to segment customers or markets faster but also over time (meaning that customers at 1 p.m. are segmented and therefore served differently from those at 2 p.m.), then you can generate additional contribution margin. Chapter 7 will tell you more about this idea (generic value driver: increasing income).

Revenues from Genetic Consulting

Calculation-intensive genetic analysis

Genetic analyses and diagnoses need a lot of computing power. Cost-efficient (and cloud-based) big data solutions widen the group of those (clinics, doctors, and so on) who are able to offer related consulting services and thereby generate respective revenue at lower costs than ever before. If they are organized as profit-oriented entities, then this should have an impact on their valuation (generic value drivers: reducing expenditures and increasing income).

Revenues from Technology Benchmarking

Anonymized benchmarking data

For many business processes, there are expert consulting firms offering benchmarking services. They do this by collecting anonymized data from a number of companies and comparing them in reports that often take months to complete.

With big data, as an example, manufacturers of printing machines get the option to collect and analyze data regarding the performance of their products in real time. Furthermore, they are—again in real time—also able to share these data with their customers, who can use them to improve their own manufacturing processes.

For the maker of such machines, this creates an opportunity to generate additional revenue; they can either sell such services at a premium or use them to make their own products more attractive than those of their competitors. The technical ability alone to offer these kinds of services should be expected to have a positive impact on shareholder value (generic value drivers: increasing income, perceptions, expectations, and preferences).

Yield per Hectare (of a Certain Crop on a Certain Piece of Land)

When size does matter

An average Swiss farm used to manage some 4.7 hectares in 1905 and cultivates almost 20 hectares today, which is next to nothing compared to New Zealand's biggest farm (Molesworth Station, over 180,000 hectares) or Australia's Anna Creek Station (some 2.5 million hectares). Farms have not only grown in terms of size. Hand in hand with farms getting bigger, they also have embraced industrial production methods, figuring out how to best use fertilizers, herbicides, or pesticides and using geospatial data for their deployment to generate maximum yields.

Big data can make a major difference in farming. Yield is driven by a number of factors (rainfall, humidity, temperature, exposure to solar radiation, appearance/numbers of pests, properties of soil and crop, and so on), and all these factors interact in ways that are not yet fully understood. The more data you have, the better you can fine-tune your activities.

By applying concepts similar to those presented in Chapter 5 or Chapter 11, farming corporations are able to analyze such interactions and fine-tune their activities (ploughing, sowing, irrigation, deploying agrochemicals, and so on) to dynamically adapt to changing environments. Remember that climate is anything but stable these days.

Going one step further, they could also (within certain limits) adapt their output to market demand (for example, by timing the insemination of cattle), or maximize the price they can get for their products (generic value drivers: reducing expenditures, increasing income, and reducing uncertainty).

Plan/Actual Deviations (Planning Accuracy)

For many businesses, exact planning data are key. As we are going to show in Chapter 4's case study, forecasts and plans are the basis for a variety of management decisions (generic value drivers: reducing expenditures, increasing income, and reducing uncertainty).

Revenues from Telecardiology

Telecardiology is the transferring and monitoring of certain parameters relevant to cardiac function (ECG, weight, pulse, blood pressure, etc.) via mobile communications. Big data enables the monitoring of such data for a huge number of patients simultaneously and also provides the statistical algorithms that are needed to identify suspect patterns.

Instead of confronting medical doctors with a flood of information, appropriate systems are able to preprocess them, triggering alerts and setting rescue services in motion automatically. One can also think about extending the range of services to other disease patterns or even general fitness. This creates a whole new spectrum of business opportunities for health providers. You will read more about this in Chapter 9 (generic value driver: increasing income).

Revenues from Telematics

Route planning based upon future traffic jams

Service providers such as Navteq Traffic or TomTom collect data by tracking mobile phones or via sensors on motorway bridges. Such data are used to capture the current traffic situation and sold to private and professional customers who use them for dynamic route planning. Nowadays, each and every smartphone is able to collect and transfer positioning data, which has led to enormous volumes of geospatial data.

At the same time, better and better algorithms can use such data for predicting traffic jams earlier and more precisely than ever before, which means that even congestion that has not yet occurred can be taken into account when determining the best possible route between two points. Existing services, such as TomTom Traffic, are nothing but precursors that vaguely suggest a whole new range of offerings to revolutionize the way we plan routes on the ground or in the air. Such offerings are going to create new sources of income for service providers and increase the shareholder value of those that grab the chance (generic value driver: increasing income).

Revenues from Price Comparison Apps

Try here, buy there

Since introducing their new price comparison app, for many traditional main street retailers American e-tail giant Amazon has turned into the incarnation of evil. This is understandable. Teenagers strolling through shopping malls or main streets on a Saturday afternoon just take pictures of barcodes of shoes or gadgets they like. Within an instant, Amazon's app tells them where they could get the jeans they are just trying on or the cool new headset cheaper online; buying elsewhere is just a one-click operation. Shops are turning into some kind of free display window for e-tailers while resulting revenues end up in the e-tailers' pockets. All the shops are left with is the costs for rent, personnel, and stock keeping.

From Amazon's perspective, the app enabling real-time price comparison is creating revenues that would not have come in at all or at least not that quickly. Although quite a few retail sellers are desperately trying to defend themselves (even using illegal jamming transmitters), such an app (which can only work using very fast, distributed in-memory databases) is creating substantial shareholder value for Amazon's investors (generic value drivers: increasing income and speeding up cash inflow).

Revenues from Situation-Specific Recommendations

"It's not the customer's job to know what they want."

(Steve Jobs)

Because smartphones know where they are and have high-speed Internet access at their disposal, the number of apps that process location- or movement-specific data has grown rapidly, and every new kind of data that a smartphone can process (temperature, step frequency, skin resistance, etc.) multiplies the number of potential applications.

Although the number of apps seems to be limitless, many of them share the same purpose. They are about recognizing a need the moment it comes into being and offering a means of satisfying it, regardless of whether this entails showing alternative sources of supply (like with Amazon's app) or projecting new pieces of furniture into your own flat (as with a new app from Ikea).

Presenting an offer promptly after a need has arisen makes it more likely that this need will lead to revenues that would otherwise not have occurred at all or, at best, later. Such revenues are the source of additional shareholder value (generic value drivers: increasing income and speeding up cash inflow).

Steps to Evaluate Business Cases

Using the matrix for your own business cases

When using the matrix in Figure 1.2 to evaluate your own business cases, you should proceed as follows:

1. Start with checking the value drivers in this book or the ones supplied in the value driver database at *www.sap-press.com/3647*. Do any of them fit your business case? Are they specific enough?
2. Collect your own supplementary value drivers. Use the tips in Section 1.4.3 to find them.
3. As well as benefit-related value drivers that increase shareholder value, also think of those that might reduce it. (With most IT projects, you will have to invest in hardware or software or spend money on implementation work or training.)

4. Try to quantify the impact of your business case on the respective value drivers and to determine what that means for your shareholder value. This is admittedly the most difficult step in evaluating business cases. On the other hand, you are facing this problem not only with big data or IT projects but also with each and every management decision—which is why your company should have tools at hand to deal with this challenge.
5. Keep timing in mind when looking at the impact on value drivers and shareholder value. By when do you expect positive and negative effects? You will need at least a rudimentary project plan for this.
6. Tune up your business plan. You need a presentation that is not too technical and will catch the attention of the non-IT crowd. Instead of talking about petabytes and FLOPS, you need to catch the attention of commercially minded executives.
7. Do not give yourself the willies. You do not have to know all the consequences of what you are planning to do here and now. It is much more important that you have thought about not only the technical but also the financial implications of your business/use case and that you are able to prove that you have thought through the implications of what you are suggesting.

Tips for handling business cases

Last but not least, a bit of advice regarding handling business cases in general:

- Business cases do not only consist of financial considerations. You need to tell a compelling story, come up with ideas for implementation, have a project plan ready, and be able to answer technical questions. In this book, however, we focus on the evaluation (that is, the financial or shareholder value-related side of business cases).
- Think about how your business case might align with strategic initiatives in your company. Most major organizations execute one or more *programs* (portfolios of projects) that were started on the initiative of top management and that are meant to serve the organization's long-term development. If you are able to align your project with one of these programs, then your chances of finding a sponsor increases by orders of magnitude.

- Very much like when designing a business warehouse, big data–related data models should be approached by what SAP once called the *chess analogy*: think strategically, act tactically. Do not try to solve all your company's problems with the very first big data project. If your project needs a lot of resources and does not deliver benefits for ages, then the risk of management losing their patience is pretty high, making you vulnerable for the attacks of opponents who have been fighting the changes your project brings from the very beginning.

  Instead, it's better to start with a small project (which might be designed as a building block of a greater whole but nevertheless can be justified on its own merits) that generates noticeable benefits as quickly as possible. Once you have demonstrated what big data/SAP HANA is able to deliver, management will be poised for assigning a higher priority to this subject.

- As well as this book, there are a lot of other potential sources of inspiration. At *www.saphana.com/community/learn/customer-stories*, you will find quite a few examples of what other customers do with SAP HANA, and quite a few consultants (including SAP itself) provide you with business case repositories. Such lists of examples might be helpful; nevertheless, you should keep two limitations in the back of your mind:

  - If you get your bearings from what other companies within your industry do with big data, this will—by definition—not lead to competitive advantage. Imitating somebody else's strategies can—at best—help you not fall behind and at least have a small chance of survival with cutthroat competition. This is the weak point of all best practice approaches. Best practice will never turn you into a leader.

  - There is no reason that you should not use such sources to get your creative juices flowing, but if you are after an unassailable lead there is no way around running your own workshops to generate ideas that are custom designed for your business and your environment. SAP calls such sessions *value discovery workshops*.

[◉]

You Do Not Have to Be Clairvoyant

The future may not be clear ...

Evaluating business cases from a commercial perspective is not about delivering a perfect forecast regarding the effects of the project you are proposing. Nobody can see into the future, and neither you nor your management really knows what kind of environment your business will have to function within later today, tomorrow, or in a year.

... but you can plan to be ready for it

Assessing business cases is not about projecting their resulting shareholder value on the spot. If you were able to predict shareholder value, you would not dwell in your cubicle but would instead live on a yacht lying at anchor at the Marquesas Islands or treat yourself to a sundowner in Hana (a village on the island of Hawaii).

Instead, assessing business cases is meant to show the following:

- You have thought about everything that might affect your scenario.
- You have thoroughly explored the sunny as well as the shady sides of your project.
- You have an idea of the factors that make or break your business case from a financial perspective.
- You have translated your technical inspiration into the language of decision makers (i.e., shareholder value).
- You have reviewed weaknesses of your reasoning and are prepared for questions and critical remarks.
- You have a set of parameters at your fingertips that you can monitor throughout your project and that can be used to measure success and initiate corrective action (i.e., value drivers).

The Babel fish . . . is small, yellow, and leech-like . . . The practical upshot of all this is that if you stick a Babel fish in your ear you can instantly understand anything said to you in any form of language.

Douglas Adams, The Hitchhiker's Guide to the Galaxy

2 SAP HANA: Capabilities and Limitations

It was cold. Derek pulled his scarf tighter around his neck. Fine, blown sand fell from his hair, trickling down his neck into the collar of his shirt. They had already been out here in the Sossusvlei—a solitude as dry as a bone in the world's most ancient desert—for about two hours. They had spent the night in the national park and hit the road at four in the morning to beat the tourist hordes.

Derek looked to his right. Soon, the horizon would take on color, letting the ferriferous sand glow for a short but perfect moment. Dune 45's eastern slope would then turn blazing red, with its western side remaining pitch-black, both only separated by a razor-sharp, curved crestline. Some kind of natural sentinel—at least on days when no marine layers drift in from the Skeleton Coast.

Light and dark, east and west: the glorious play of colors was photogenic but could also serve as a compass. It was thought that humans were roaming this area as long as 150,000 years ago. They probably didn't have GPS devices in their backpacks, but they still managed to find their way in this endless monotony, maybe by using the patterns of light and shadow but perhaps also because they provided cardinal directions and elevations with labels, laying a mental grid over the landscape. Migrant birds were able to use the Earth's magnetic field for navigation, but humans needed fixed points, location names, and patterns to find their way.

Figure 2.1 Dune at Sunrise, Namib-Naukluft National Park, Namibia

Today, the whole planet is covered by a tight and clearly defined system of coordinates, every mountain has its name, and the sand hills by the side of the dirt track have been numbered by road kilometers. A bit further to the southwest, near the Deadvlei, a few bumps have even more poetic descriptions: Big Daddy and Big Mama are two of the biggest dunes on the planet. Might there also be a dune called Big Data? And if so, would SAP HANA be just another term for it, or would it refer to a subset, such as the eastern or western slope of a heap of sand?

In the middle of this shoreless sea of sand, Derek could not help but once again think of all these new products and terms from SAP that had—coming from forums and newsletters—percolated through the Internet onto his screen during the last couple of months. He badly craved some kind of classification system, some kind of orientation—or even more a Babel fish that would nestle in his inner ear and not only translate all these acronyms and flowery names but also structure them for him.

Coordinate system for big data

Whether you are planning to cross the Namib Desert or just renovate your data warehouse during the next couple of months, a coordinate system as a basis for navigation and planning will dramatically increase your chances of survival. This is one of the key reasons that humans have, since antiquity, assigned names to rivers, mountains, canyons, and constellations, using systems based upon longitude and latitude, and why more recently they have come to rely on procedure models, modeling languages, and framework architectures in IT.

Classifying SAP HANA

In the previous chapter, you had a first glance at the world of big data and pigeonholed everything that belongs to it into a virtual grid consisting of the four categories technology, algorithms, methods, and architectures. But because the title of this book isn't *Making the Case for Big Data* but *Making the Case for SAP HANA*, the next step for you will be localizing the components of the SAP HANA world within this grid while at the same time refining the coordinate system itself.

How are SAP HANA and big data related? Are both the same, and is SAP HANA just SAP's trademark for big data? Does SAP HANA provide you with more or less functionality than big data solutions from other software vendors?

Functional scope of SAP HANA

To begin with, we are going to put both worlds under the microscope, explaining which elements of big data are not components of SAP HANA and vice versa. Then, we are going to have a closer look at some highlights of SAP HANA (mainly its integration with other SAP and non-SAP solutions). This is because just listing individual functionalities won't be enough to fully understand the potential of SAP HANA. Instead, we are going to introduce a couple of *implementation scenarios* (an SAP term for what is called *system architectures* in The Open Group Architecture Framework [TOGAF]). When defining these implementation scenarios, we are going to focus on two questions: what additional benefits can be derived from synergies between the various components, and which scenario should be used when?

Finally, we are going to have a look at some trends that are already visible to the naked eye; the thoughts about trends that affect big data will lead us to the conclusion that ideas will be one of the key limiting factors in big data during the next couple of years.

Are you safe?

Why do you need to know and understand all this? Because your operation is probably under threat already from a competitor who has caught onto the competitive power of big data before you have. If you are not being targeted yet, you probably will be soon, possibly by a company not currently in your market but planning to be. So, read on—it will be worth the effort—or take a quick peek at Section 2.3.2 to understand why industries that seemed to be worlds apart a couple of years ago are now fighting for the same customers.

SAP HANA knowhow available free

In this context, we will provide you with a couple of indications of how we intend to familiarize you with the functionalities of SAP HANA. In the introduction to this book, we mentioned that a great deal of information about SAP HANA is freely available. In terms of SAP HANA, SAP is following a strategy pretty similar to that of Apple in terms of apps: technical information is made available generously and often free of charge, enabling developers big and small to come up with as many applications as possible for SAP HANA in the near future.

For beginners, a lot of this information can be difficult to understand. Which is why—in addition to information that is available on the Internet—quite a few good books about the technical details of SAP HANA have been published. Unlike this freely available information and those books, we are not focusing on how to handle, configure, or administer the solution (that is, on questions such as how to create analytic views; we'll explain analytic views in Chapter 4, Section 4.5.2). When discussing technical details, we will always look at them through the eyes of business experts and their requirements. By examining the worlds of big data and SAP HANA in this way, we are going to bridge the gap between potential benefits in your business environment and the functionalities of SAP HANA.

Focusing on software

As mentioned before, SAP and its partners are offering SAP HANA as an appliance. This means that customers are not totally free to select their hardware boxes and that the interaction between hardware and software is not transparent from the customer's perspective. Furthermore, the performance of hardware in terms of main memory and maximum number of CPU cores changes on an almost daily basis. A discussion of hardware goes beyond the scope of this book, but relevant information (for example, with regards to sizing and sizing tools) can, for example, be found in the SAP PRESS book *SAP HANA: An Introduction* (Berg and Silvia, 2014).

2.1 Big Data and SAP HANA

Technology, algorithms, methods, and architectures

In Chapter 1, Section 1.1, we explained that big data solutions as a whole consist of more than just an in-memory database running on a cluster of superfast servers with working storage and CPU cores galore. Big data consists of technology (which embraces hardware as well as languages and platforms), algorithms, methods, and architectures. To establish a border between big data and SAP HANA, we will first explain which technologies, which algorithms, which methods, and which architectures are used by big data solutions that get along without SAP HANA or have come into existence before SAP HANA. After that, we will take a closer look at SAP HANA.

2.1.1 Big Data without or before SAP HANA

Open-source solutions

Big data solutions outside of the system boundaries of SAP HANA often run on a foundation of very ordinary, more or less generic hardware. Based upon this foundation, a variety of open-source platforms and products are knocked together to form toolchains. *Open-source software* is software that is freely available within the public domain and that can be used, modified, and distributed without paying license fees. A *toolchain* is a combination of independent IT tools (platforms, databases, programming languages, and so on) that interlock—more or less seamlessly—like the links of a chain. Every tool uses the output of another tool as its own input, carries out a specific task within a process, and passes on its result to the next one.

To ensure you won't get lost with all the technical terms introduced in the following paragraphs, we are going to file them into our big data classification system (technology, algorithms, methods, and architectures) established in Chapter 1, Section 1.1.

Technology (Hardware)

Off-the-rack hardware

The hardware components of the most powerful big data solutions often don't come from just one supplier. Big data pioneers such as Amazon or Google don't use appliances or exotic supercomputers for their distributed systems but instead link up high-grade, though still standard, boxes from different manufacturers. Special software then coordinates the operation of these components.

This practice (that is, using a large number of standard hardware components to build high-capacity systems for distributed computing) is often called *commodity (cluster) computing*. Figure 2.2 shows a (not at all complete) list of hardware vendors whose products might be used as building blocks of big data solutions.

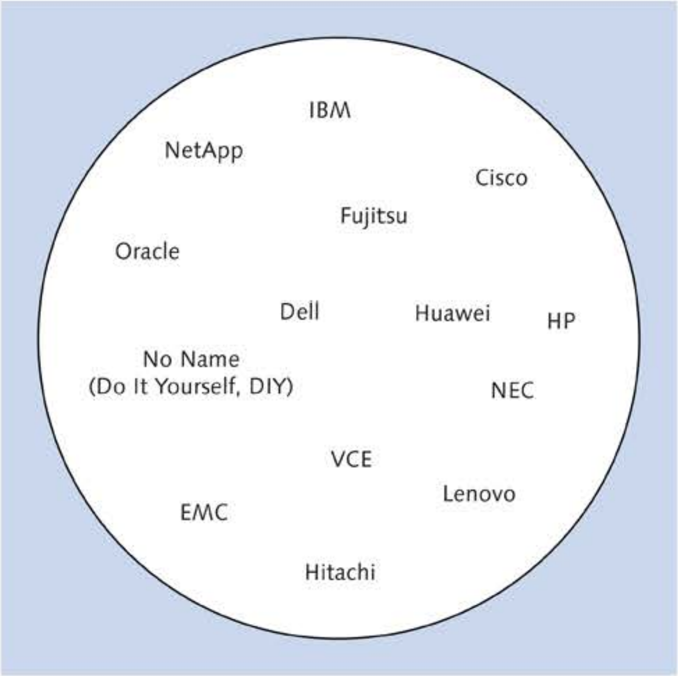


Figure 2.2 Big Data Hardware

Technology (Languages/Platforms)

Toolchains for big data

Big data solutions bring together a multitude of different products, not only in terms of database-management systems, languages, and platforms. Big data toolchains may contain, among others, the following chain links:

- **In-memory databases (such as SAP HANA)**
  Many big data applications use in-memory databases for data storage. Apart from SAP HANA (which is more than "just" a database), there are other, nonpersistent databases, which are either freely available or sold via traditional licensing models. Examples of other in-memory databases include Apache Derby, IBM Informix Accelerator, and IBM BLU.

- **NoSQL databases (such as Apache Cassandra, CouchDB, MongoDB, or OrientDB)**
  NoSQL stands for *not only SQL* and refers to nonrelational, schemaless databases that are able to handle huge numbers of write and read requests and extremely high data volumes. NoSQL databases are often deployed in distributed environments and for special purposes—for example, storing documents or graphs.

- **Database manipulation and query languages (such as SQL)**
  Today, SQL is the dominant database-manipulation and query language for relational databases. Database-manipulation languages are used to define structures within a database (such as tables) and to manipulate (write, update, and delete) data that are stored using these structures; query languages read data from databases. With classic as well as with in-memory databases (like SAP HANA), SQL is often the key link between application and database.

  Apart from special-purpose query languages, such as MDX with OLAP, other query languages are only of secondary importance.

- **Programming languages (such as Python, R, Java, or Erlang)**
  - Python is a programming language that can (for example) be used to tap into tweets. An example in which Python Script is used to provide Twitter data for further analysis via SAP PAL on SAP HANA can be found by searching "Predicting My Next Twitter Follower with SAP HANA PAL" (or referring to the link in the additional resources we recommend in the book's online appendix). Another reason that Python is a valuable resource in the realm of big data is the fact that there is a powerful Python-based suite of libraries and programs for symbolic and statistical natural language processing, the so-called *Natural Language Toolkit* (*NLTK*). Developers often make use of NLTK for text mining and sentiment detection.
  - R is a language that is primarily used to develop statistical applications. The language is available on a number of operating systems (UNIX, MacOS, and Windows) and can be downloaded free from servers all over the world (*http://cran.r-project.org/mirrors.html*). For SAP customers, R is of special interest, because SAP HANA supports calling R procedures from SQLScript procedures.
  - Java is probably the best known of all open-source programming languages. There are Java libraries serving almost each and every

  purpose one can imagine (including libraries related to MapReduce, Google's famous invention mentioned in Chapter 1, Section 1.2). Even SAP uses Java (on top of its own, well-known ABAP language) with the SAP NetWeaver Application Server (AS).

  - Erlang is a programming language developed by Ericsson and named after the Danish mathematician Agner Krarup Erlang. Originally devised for use in telecommunications (where it is still widely used), Erlang has become very popular with distributed environments that need to meet high expectations in terms of system availability (which applies to many big data applications).

- **Storage/processing frameworks (such as Apache Hadoop)**
  Apache Hadoop is a framework developed in Java that mainly consists of a file system used to store very large amounts of data (Hadoop Distributed File System [HDFS]) and configurable classes for Google's MapReduce programming model. In the context of SAP HANA, SAP considers Hadoop *the* solution for storing practically unlimited amounts of data.

- **ETL tools (such as Pentaho Data Integration, SQLServer Integration Services [SSIS], and SAP Data Services)**
  As we are going to show, certain persistent layers within data models for data warehouses are no longer needed with in-memory databases. Nevertheless, data processed by big data solutions still need to be acquired, cleansed, and enriched afterwards, which are jobs performed by ETL (extract, transform, and load) tools, such as SAP Data Services or the built-in ETL functionalities within SAP BW. Transforming data is still indispensable, regardless of whether such transformations take place between persistent data pools (such as data in classic relational databases), transient data stores (such as data in in-memory databases), or even between virtual objects (such as InfoSources in SAP BW).

- **Visualization tools (such as Tableau or products from SAP BusinessObjects BI)**
  The results of highly complex analyses and algorithms still need to be prepared in a form that is digestible for humans, for three reasons:

  - Big data applications are not always meant to decide or act autonomously. In many cases, their only purpose is to deliver the foundation for human decision making and the resultant human actions.

- Some dependencies remain hidden to even the most powerful algorithms, yet they might be obvious to every human looking at a graphical representation of the data.
- The graphical representation of an algorithm's input or output data often indicates whether false conclusions have been drawn or whether the algorithm itself, its designers, or its users have been falling for statistical mirages (see Chapter 1, Section 1.2.2).

In all three cases, one needs powerful presentation tools that can do a lot more than Microsoft Excel. The number of vendors in this area is legion. Almost every supplier of a data warehousing product has also come up with his own proprietary reporting solution. In parallel, there are numerous specialized suppliers whose products are either available free or for a cost. Some of these products are traditional pieces of software that need to be licensed and then installed locally; others can be used online as a service. In addition, programming languages such as R or special tools such as SAP PAL come with functions for graphically representing data.

The answer to the question of which tool might best suit your needs results from the desired display format; defining the display format is the job of data artists (we are going to introduce this new job profile soon, in the "Algorithms" section).

Figure 2.3 provides you with an overview of some of the key languages and platforms in the big data world, regardless to which of the two categories they belong. In other parts of this book, we will once again pick up some (but not all) of these terms.

Open source: advantages

In terms of hardware, many customers choose commodity computing to keep down costs. With languages and platforms, the decision for open source is often not (only) cost-driven; on the contrary, customers have realized that no single supplier offers the whole bandwidth of instruments that are needed for big data. Further, even if such a supplier existed, it would not be able to deliver the best possible solution in each individual category.

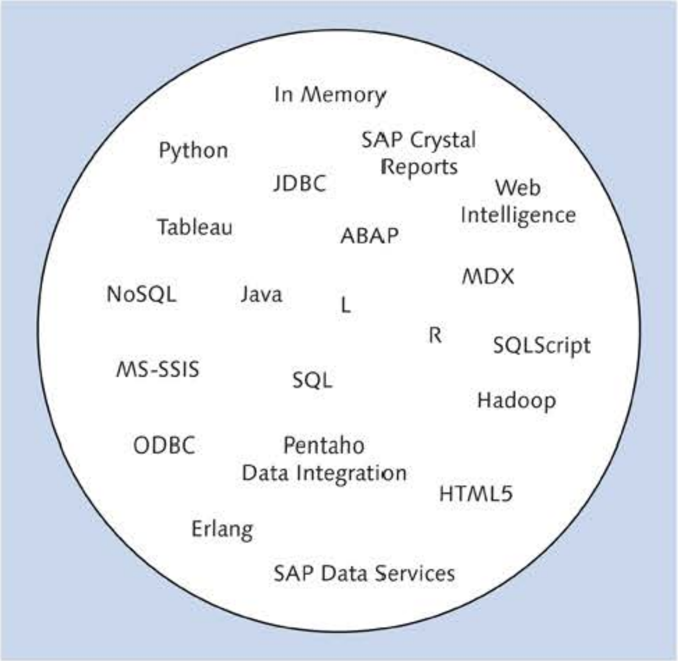


Figure 2.3 Big Data Languages/Platforms

Algorithms

In Chapter 1, Section 1.1.2, we listed some areas of expertise (such as speech recognition, text mining, and object recognition) that are especially important for big data solutions. In all these domains, there are a variety of mathematical, statistical, heuristical, and other algorithms. Most of them have emanated from academic research and are therefore in the public domain.

Standard algorithms are not innovative

But, as we have already mentioned, when talking about how to find inspirations for business cases (see Chapter 1, Section 1.5), marketing materials and published vendor showcases will not provide you with exclusive and revolutionary innovations. Ideas and algorithms that are already in the public domain will not help you set the world on fire.

PAL, for example, provides you with the *k-Means algorithm* for clustering. A number of case studies on the Internet demonstrate (for example, with SAP HANA Academy; see *www.saphana.com/community/hana-academy*) how this algorithm can be used for market segmentation. k-Means, however, is a technique that was developed almost 60 years ago and is being taught at every respectable university in basic classes on statistics. If you rely on such standard algorithms, you will only get a head start on very small or very backward competitors. Granted, the backlog of many companies in terms of data analysis is so big that even the smallest improvements may make a difference, but if you intend to turn business processes inside out, if you really want to use the potential of big data big time, and if you want to take leadership, then you will need more than that. There are two measures that can really help you gain ground. Get hold of well-trained data scientists and data artists, and use crowdsourcing.

Data scientists/ data artists

The emerging data scientist profession was mentioned in Chapter 1, Section 1.2.2. In contrast, a *data artist* does not involve himself in data architectures and data analysis but in presenting data in brain-friendly shapes. A data artist's job consists of developing display formats that go beyond simple line charts or bar charts. His objective is to come up with figurative representations of data that will make it easier to spot relationships, trends, or patterns. Unlike the data scientist, a data artist therefore also needs skills in areas such as communications design and cognitive psychology. Without such knowhow in your organization—from data scientists, data artists, and experienced external consultants—you will only get "me too" big data solutions going.

Crowdsourcing as a source for top-level expertise

When searching for basic algorithms, even your experts might want to rely on proven, public, and free sources. When, however, it comes to developing more sophisticated procedures, there is another very interesting alternative: *crowdsourcing*. Quite a few books have already been written about this topic; we will therefore only briefly discuss it here.

Even if you have top-notch data artists and data scientists on board and even if you work closely with leading universities, you may not—or at least not *exclusively*—want to rely on your staff and your partners when it comes to developing new algorithms that are meant to take you well ahead of your competition. Indeed, it may be a lot more efficient to use your own resources to define, manage, and evaluate crowdsourcing projects than to have them invent algorithms. Crowdsourcing provides a highly exciting means to tap into the brains of those who may not happen to be on your payroll but who are the best experts on the planet. We believe that successful organizations—especially when it comes to big data applications—will not be able to ignore crowdsourcing; in fact they will gain great benefit from embracing it.

[»]

Crowdsourcing

The term crowdsourcing is a neologism composed from the words "crowd" and "outsourcing." *Outsourcing* stands for transferring—in terms of time and scope—clearly defined internal tasks, processes, or projects to external suppliers. The difference between outsourcing and crowdsourcing lies in the fact that, in crowdsourcing, jobs are not transferred to one contractor but instead to a more or less anonymous group of people. Tasks are defined precisely—in terms of scope, duration, objectives, remuneration, and so on—and are then made available to a limited number of interested parties via crowdsourcing portals.

One of the biggest such portals is Amazon's Mechanical Turk (*https://www.mturk.com*), named after the mechanical chess player in Edgar Allan Poe's essay "Maelzel's Chess Player." Other examples include *http://crowdflower.com* and *www.ideaconnection.com*.

Crowdsourcing is buying brain not brawn

In crowdsourcing, a number of individuals or teams often work on the same task in parallel, competing against each other. There is usually a fairly loose relationship (often arranged by the portal) between client and contractor. The required services are usually not provided on site with the client but remotely, using modern information and communication technologies (mainly the Internet).

[Ex]

Crowdsourcing Portals

One example for a crowdsourcing portal that primarily deals with big data–related problems is *www.kaggle.com*. A quick glance at their website and its active projects will reveal that not only research institutions and universities but also very respectable business giants (such as General Electric [GE]) use their services. While this book was being written, GE—a very important manufacturer of jet engines—was searching for an algorithm to optimize flight routes in commercial aviation—an almost exemplary big data topic, considering the many ever-changing influencing factors (such as weather, traffic situation, flight control restrictions, etc.) and the fact that an optimization only makes sense if performed in real time.

By the way, SAP also has its own crowdsourcing portal, called *SAP Idea Incubator*; unlike Kaggle (which focuses mainly on big data), the SAP Idea Incubator is used to exchange ideas related to SAP HANA and SAP's solutions in general. For further information, refer to the additional resources we recommend in the book's online appendix.

Company-owned portal

To play in crowdsourcing, you don't have to use a portal's services. A separate area on your website that you advertise in appropriate social media and that you use to present your challenges and the promised rewards will be more than enough. One thing you cannot do without, however, is experts that are able to phrase problems in meaningful ways and that can manage resulting projects.

Crowdsourcing versus working customer

A key difference between crowdsourcing and the (not particularly nice but widely spread) idea of the *working customer* (the trend to transferring value-adding jobs to the customer, such as by self-service, tidying up at fast food places, or self check-in for air travel) lies in the fact that in crowdsourcing the customer–supplier relationship is clearly visible and the supplier gets paid for his work. Payment for services also distinguishes crowdsourcing from open source. Although developers of open-source products might still get paid voluntarily by end users, there is no clearly defined commercial customer–supplier relationship with open source.

Classic algorithms for big data

Although we emphasize that you probably will not get around (and should not even try to get around) to developing your own algorithms, we have compiled some classic ones that you may encounter in the context of big data in Figure 2.4.

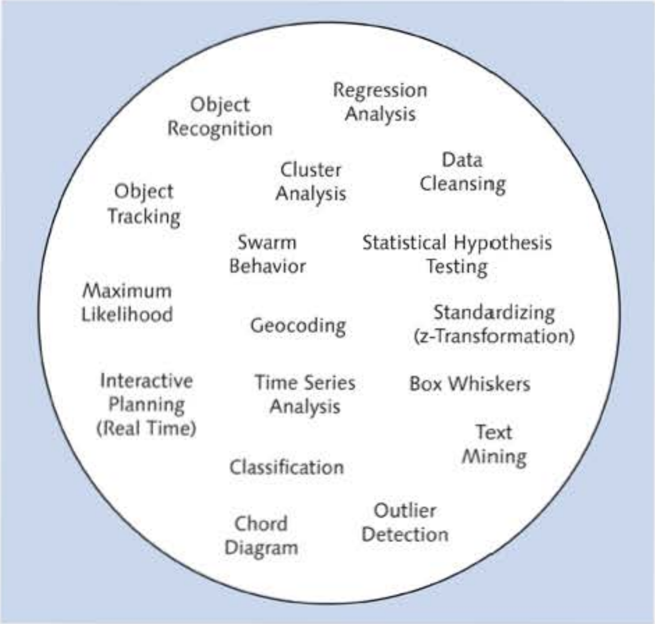


Figure 2.4 Big Data Algorithms

Please note that most of the terms in this figure do not refer to individual algorithms but rather to classes or groups of algorithms.

[Ex]

Statistical Hypothesis Testing

One class of algorithms shown in Figure 2.4 is statistical hypothesis testing. As of the time of writing, the programming language R provides you with six different statistical tests (in the language's standard scope or via downloadable packages) dealing with only one question: whether a certain parameter was normally distributed or not!

Some of the groups of algorithms shown in Figure 2.4 are relevant for our cases studies, and we will explain them in the relevant context. The purpose of Figure 2.4, however, is to give you an idea of the spectrum of areas your experts might have to deal with when building big data applications.

Methods

When discussing methods, we have already mentioned agile business intelligence (BI) or Google's MapReduce programming model. Other examples already touched on as well are schemaless databases (that is, the NoSQL approach) and all principles of operation that are employed with parallel processing and concurrency.

Methods are neither solution- nor product-specific

As with algorithms, reinventing the wheel in terms of methods doesn't make much sense either. Imagine building a house: your architecture and your building materials might be revolutionary, but the rules of structural analysis still remain the same regardless of your innovative power. In terms of methods, there is no reason that you should not rely on proven best practices. Unless you have developed a brand new approach allowing you to drastically shorten the process of designing, developing, or implementing new solutions, you could well go for the same methodological approaches your competitors are depending on. Figure 2.5 contains a couple of keywords related to methods that are common in big data environments.

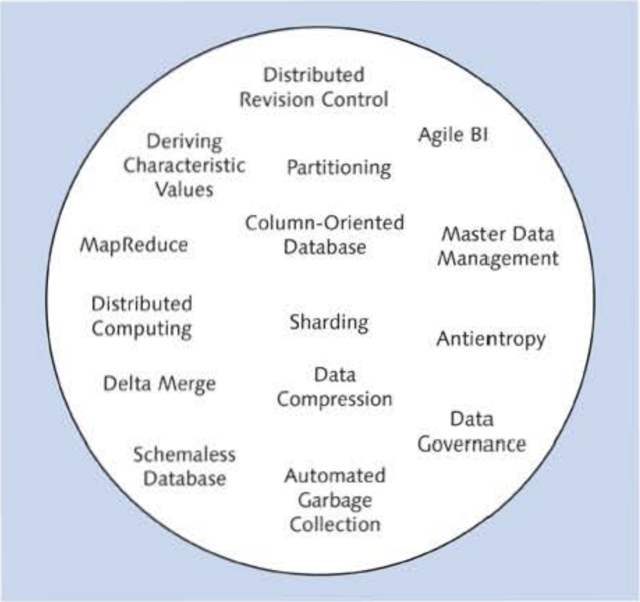


Figure 2.5 Big Data Methods

Architectures

Architectures are neither solution- nor product-specific

So far, there aren't many architectural approaches that are exclusively devised for big data applications; LSA++ from SAP may be one of the few exceptions. But as many big data solutions are implemented on clusters or grids, architects often revert to reference architectures for distributed systems (for example, event-driven architecture [EDA]). Figure 2.6 shows some architectures or components of architectures that play a major role in big data.

Big data recipes

Figure 2.2 through Figure 2.6 show just a small fraction of the ingredients you could use in your own big data solution. Based upon your specific business cases, your organization will use these plus other elements to engineer its own big data recipe. Think of your big data solution as a pizza: just as with a great Quattro Stagioni (a "four-seasons pizza"), pleasure does not originate from one single ingredient but from the interaction among all the toppings. Comprising hardware, languages/platforms, algorithms, methods, and architectures, your big data flatbread actually comes with five instead of only four seasons (or segments). To convince and convert your top management, all five toppings should come appealingly arranged, thoroughly baked, captivatingly fragrant, and laid out on a crunchy–crispy business case.

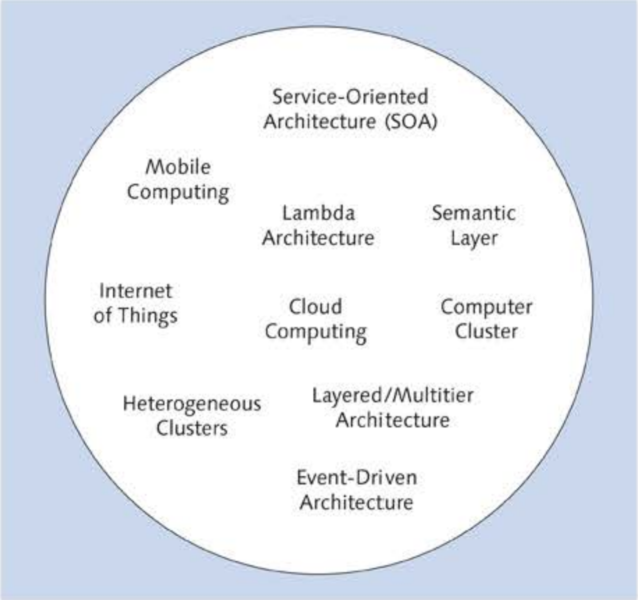


Figure 2.6 Big Data Architectures

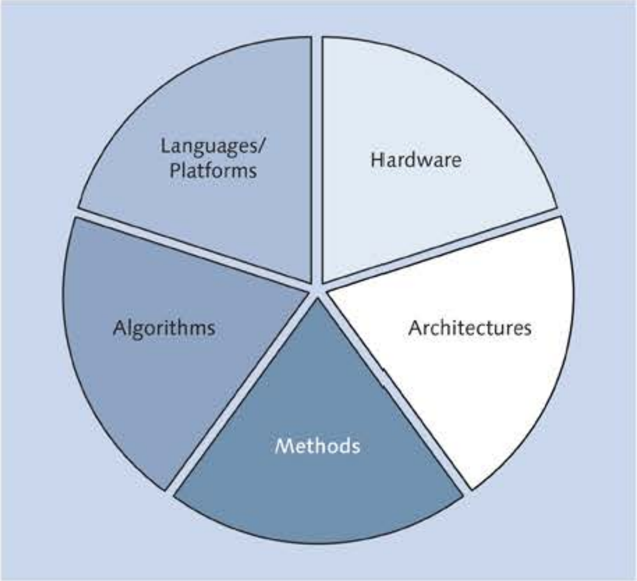


Figure 2.7 Big Data: Your Solution

2.1.2 What Does SAP HANA Consist of?

SAP HANA is not convenience food

Maybe breakfast is already a couple of hours behind you. You have checked your e-mail, finished the most important calls, and then taken some time to read this book during the rest of the morning. Slowly, your appetite for lunch is starting to stir. The readout on your bathroom scale this morning, however, has induced you to limit yourself to two apples, one orange, and a couple of nuts, which is good for us: a growling stomach might be useful in terms of arresting your attention a bit longer via the pizza metaphor. We will thus continue to use this image to define the scope of SAP HANA compared to the scope of big data. If you imagine a delicious big data solution as a Cinque Stagioni (a "five-seasons pizza"), SAP HANA is more than microwavable convenience food from a refrigerated display case, and it is certainly not just a collection of recipes.

First of all, SAP provides you with a portfolio of use and business cases in terms of big data. You may consider this portfolio—for example, the customer success stories at *www.saphana.com/community/learn/customer-stories*—a culinary source of inspiration. Customer success stories provide you with an idea of the spectrum of potential business cases and teach you that there are a lot of different ways to make pizza crust.

You will select your level of individualization

If you do not feel like baking at all, SAP HANA can also come up with a ready-mix packet of dough (SAP Rapid Deployment Solution [RDS], see *https://service.sap.com/rds* or */www.sap.com/solution/rapid-deployment.html*) or a frozen, ready-to-eat pizza that you just need to pop into the oven (to be found on the SAP marketplace at *http://marketplace.saphana.com/search/all*).

Furthermore, there is everything that you can think of between these two extremes. You may prepare the crust yourself (that is, develop your own business case using morphological analysis; see Chapter 1, Section 1.5) and used canned tomatoes (that is, the preimplemented algorithms in SAP PAL) for the topping.

Innovations do not come canned

But cross your heart and hope to die: on Facebook, you have found your partner for life and you finally managed to bridge the gap between cyberspace and the real world to arrange a date. Are you going to march the woman/man of your dreams to a Pizza Hut in a nearby shopping mall or—if you should dine at your place—peel the plastic wrapping from a frozen pizza or mess around with a can opener? Of course not! Instead, you would probably try to find a virtuoso who still kneads his pizza dough using a traditional recipe of his Neapolitan nonna (granny), having flown fresh Ramallet tomatoes from the Balearic Islands every day.

Similarly, when it comes to presenting a big data project to top management, you probably do not want to base your business case on an example your main competitor has been peddling at trade shows and conferences for years. Furthermore, it might not be recommended—as *the* highlight of your presentation—to suggest using a "brand-new" algorithm called k-Means for customer segmentation.

If we stick to our pizza analogy, SAP HANA is neither convenience food nor a cookbook providing nice photographs but still leaving you hungry. You might rather consider SAP HANA to be a product group within an Italian delicatessen store that specializes in everything you need to make pizza. With some ingredients (like hardware), the choice of ingredients is a bit restricted (maybe to keep prices high or maybe to guarantee a minimum quality); with others (such as algorithms), you may go for SAP's store brands (that is, preimplemented stuff, like in SAP PAL). At the same time (due to the openness of SAP's interfaces), you are also free to stock up elsewhere.

SAP HANA within SAP's product portfolio

To make a good pizza, you also need a rolling pin, a pizza wheel, and maybe even a wood-fired oven. Once the pizza is ready, you may also want to have a good glass of Montepulciano d'Abruzzo with it. Which is why the delicatessen store called "Alimentari SAP" (SAP Groceries) has taken up residence next to a couple of other specialized shops also owned by SAP. Most of them are easy to reach via passages; you don't even have to leave the building to buy kitchen accessories (SAP NetWeaver Application Server), electrical appliances (SAP Business Suite) or fine wines (SAP BusinessObjects BI).

SAP HANA and open source

Alimentari SAP's direct neighbors are, however, only offering SAP products. But as the whole ensemble of SAP's shops is located right in the middle of Little Italy (that is, as SAP HANA is a pretty open system in terms of data acquisition and data distribution), a few other dealers with similar assortments of goods are within walking distance.

If any of the following applies, you may want to consider buying some toppings elsewhere:

- You need toppings (solutions) that SAP is not offering at all (such as Hadoop or Python).
- You need a special variant of a topping (that is, you need to satisfy a business requirement such as showing relationships in a *chord diagram*, sometimes also called a *radial network diagram* or—after a software package—*Circos diagram*) that SAP does not have on its shelves. SAP might sell mozzarella cheese (SAP BusinessObjects BI solutions to create diagrams) but not buffalo mozzarella from Campania (chord diagrams).
- You simply do not want to buy certain products from SAP because it is your company's policy to always buy flour (that is, certain pieces of software) from other preferred suppliers.

But even if you do not get all your food from Alimentari SAP, integrating products from nearby shops into SAP HANA-based solutions (like calling R functions from procedures written in SQLScript) is often feasible with reasonable effort—in very much the same way that frozen spinach from Birds Eye could be combined with Giorgio sliced mushrooms on one and the same pizza.

As in Section 2.1.1, we are now going to look at each of the five segments of our pizza individually.

Technology (Hardware)

Hardware partners for SAP HANA

SAP HANA is sold as an appliance. Hence, some hard restrictions apply in terms of the hardware you can use. As of April 2014, you could choose from among the following 10 hardware suppliers (in the e-book, the hyperlinks will take you to the SAP HANA–related areas on the hardware manufacturers' websites): Cisco, Dell, Fujitsu, Hitachi, Huawei, HP, IBM, Lenovo, NEC, and VCE.

Based on Figure 2.2, your hardware options are those shown in Figure 2.8 (for the time being, you are limited to the suppliers within the circle when selecting your boxes).

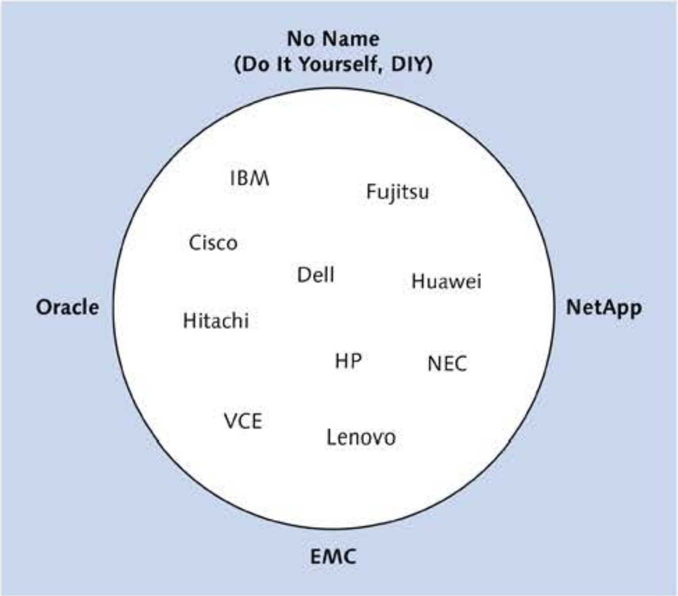


Figure 2.8 SAP HANA Hardware

Up-to-date information about hardware options

As SAP continues to form new alliances, the list of possible hardware options will be in a constant state of flux. You will always find your current choices on SAP's website (*www.sap.com/pc/tech/in-memory-computing-hana/partners.html*) or at SAP Service Marketplace (*https://service.sap.com/~sapidb/011000358700000701932011E*). The most reliable source of such information is SAP's Product Availability Matrix (PAM), which can be found at *http://service.sap.com/pam*.

Technology (Languages/Platforms)

Open and proprietary platforms

Defining the domain of SAP HANA in terms of languages and platforms is a bit more complicated than it is with hardware. Due to SAP HANA's openness, there are no hard borders. In terms of languages and platforms, one can distinguish among four categories:

- **SAP HANA languages/platforms**
  First of all, there are languages and platforms that SAP specifically developed for SAP HANA and which are an integral part of the solution (more or less forming its core). This includes SAP HANA's database-management system (with its column- and row-based in-mem-

ory technology) and the database-manipulation and query language SQLScript.

- **SAP languages/platforms**
  A bit further away from SAP HANA's core are other languages and platforms that are also part of SAP's product portfolio, using SAP HANA as a database or serving as a foundation for SAP HANA-based solutions. Members of this class are the business intelligence solutions within SAP BusinessObjects BI, the programming language ABAP, and SAP's application platform SAP NetWeaver Application Server.

  Step-by-step, SAP is also enriching many of these languages and platforms with supplementary functions and extensions supporting SAP HANA. Unlike SQLScript, however, all of these products have been around before and are independent of SAP HANA.

- **Supported languages/platforms**
  Furthermore, SAP HANA-based big data solutions can be built on the basis of languages or platforms that are either provided by other suppliers or freely available and whose teamwork with SAP HANA is, in all cases, enabled by SAP or by the supplier of the language, platform, or open-source community. Members of this group of languages and platforms include the following:

  - The programming languages L and R (which can be used in SAP HANA procedures)
  - The storage solution Apache Hadoop (SAP is a reseller for Intel's Hadoop)
  - The database-management, manipulation, and query language SQL (on which SAP HANA's SQLScript is based)

- **Other languages/platforms**
  Finally, there are some languages or platforms that can interact with SAP HANA but have to rely on generic standard interfaces for accessing data stored within SAP HANA. Two well-known representatives of this category are the programming language Python and Microsoft's SSIS:

  - Python can connect to SAP HANA databases via the Open Data Protocol (OData); OData is an HTTP-based access protocol developed by Microsoft and used to execute database operations.

- SSIS is an ETL solution developed by Microsoft; SSIS is able to get data from SAP HANA via Microsoft's .NET framework.

In Figure 2.9, the distance of the various languages and platforms from the center of the inner circle indicates their level of integration with SAP HANA or SAP solutions in general:

- The components within the inner circle of the diagram are an integral part of SAP HANA.
- Those in the first ring (counting from the center) are products that are made by SAP but are not part of SAP HANA.
- The next ring contains components that do not come from SAP but are supported by SAP HANA in standard.
- The last, outer ring lists languages and platforms that can be used in toolchains together with SAP HANA, but for which there are no special connections to SAP HANA (for example, in the form of custom-built interfaces).

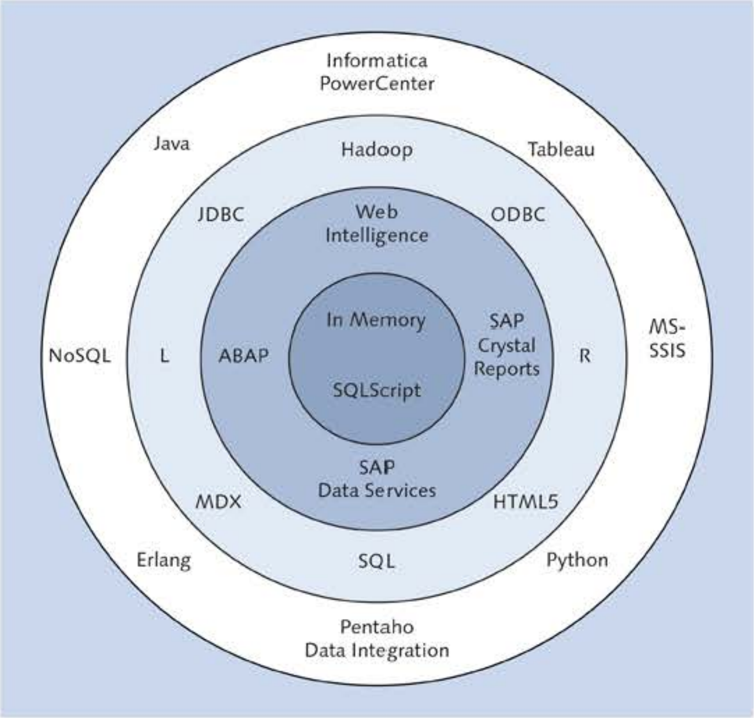


Figure 2.9 SAP HANA Languages/Platforms

[+]

Completeness of Components Mentioned in this Book

Please keep in mind that Figure 2.3 through Figure 2.6 do not show all possible building blocks for the big data world; in the same way, our lists regarding what SAP HANA consists of are far from being complete. We are not trying to exhaustively describe all components (you can always find this information online), but rather we are trying to classify some of the more important ones, helping you find your way through the abundance of ready-to-use algorithms.

Algorithms

Ready-to-use algorithms

We have already mentioned that most of the algorithms required for big data are either within the public domain or have to be invented for the solution you have in mind. As far as we know, there are no (earth-shaking) mathematical or statistical algorithms (such as those for classification) that SAP itself has invented or developed from scratch and which would thus only be available to SAP customers.

SAP HANA or products based upon SAP HANA (like SAP PAL) do, however, contain quite a few common preimplemented algorithms. In general, algorithms can be classified in much the same way we have classified languages and platforms—that is, via their distance from the core of SAP HANA (see Figure 2.10):

- **Algorithms within SAP HANA**
  Some algorithms are implemented within SAP HANA. One example is text analysis; text analysis became available as of SAP HANA SPS 6 (revision 60) and includes basic functions for sentiment detection (for example, via the `EXTRACTION_CORE_VOICEOFCUSTOMER` option).
- **Algorithms within solutions based upon SAP HANA**
  Additional algorithms are provided by solutions based upon SAP HANA (for example, by SAP PAL); two examples are outlier detection (function `ANOMALYDETECTION`) and cluster analysis (function `KMEANS`).
- **Algorithms within non-SAP solutions**
  The spectrum of accessible algorithms becomes even wider once we cross the limits of SAP's own product portfolio and enter the realm of languages and platforms supported by SAP. In R, there are, for example, a multitude of algorithms for statistical hypothesis testing (such as the tests for statistical normality mentioned in Section 2.1.1) or

graphical functions (such as the box-and-whisker plot often used to compare distributions). The list of functions provided outside of R's standard vocabulary via custom-designed packages is more or less unmanageable. An up-to-date overview of these packages (there are over 11,500 listed!) can be found at *http://cran.r-project.org/web/packages/available_packages_by_name.html*.

- **In-house development still indispensable**
  Nevertheless, there will still be numerous—more or less complex—cases in which you will have to build your own algorithms using SAP tools (such as ABAP) or non-SAP languages and perhaps business requirements for which you will have to adapt existing ones.

  This does not only affect very innovative or exotic requirements. Two examples for which algorithms exist but might have to be improved are as follows:

  - Recognizing and tracking objects/people (used to find out how, for example, customers roam your shops).
  - Optimization via *simulated annealing* or *stochastic tunneling*; simulated annealing is a heuristical algorithm that can be used for scheduling resources (field service technicians, planes, etc.); stochastic tunneling helps determine global (instead of local) optima and is often used for designing integrated circuits.

Functionality is release specific

Much like the choice of hardware suppliers you can select from, the scope of preimplemented algorithms changes with each and every new release and service pack. Figure 2.10 categorizes the groups of algorithms in Figure 2.4 much like Figure 2.9 classified languages and platforms.

[+]

Advantages and Disadvantages of Preimplemented Algorithms

Preimplemented algorithms are great because:

- You can use them without any further development effort.
- They are implemented properly; you do not have to worry about errors or testing.

On the other hand, you still won't get around the following activities:

- You must select the most appropriate algorithm for your purpose.
- You must make some algorithm-specific settings (that is, setting certain control parameters for that algorithm).
- You must feed data to the algorithm.

These three activities require far more mathematical and statistical knowledge than customizing an ERP solution. Even worse, the damage you could cause by making mistakes (and thus triggering incorrect human or automated decisions) goes far beyond posting material consumption to the wrong account.

An incorrect posting can easily be reversed; incorrect decisions are often only reversible with a lot of goodwill from friends at the stock exchange and could, for some companies, cost millions of dollars per minute.

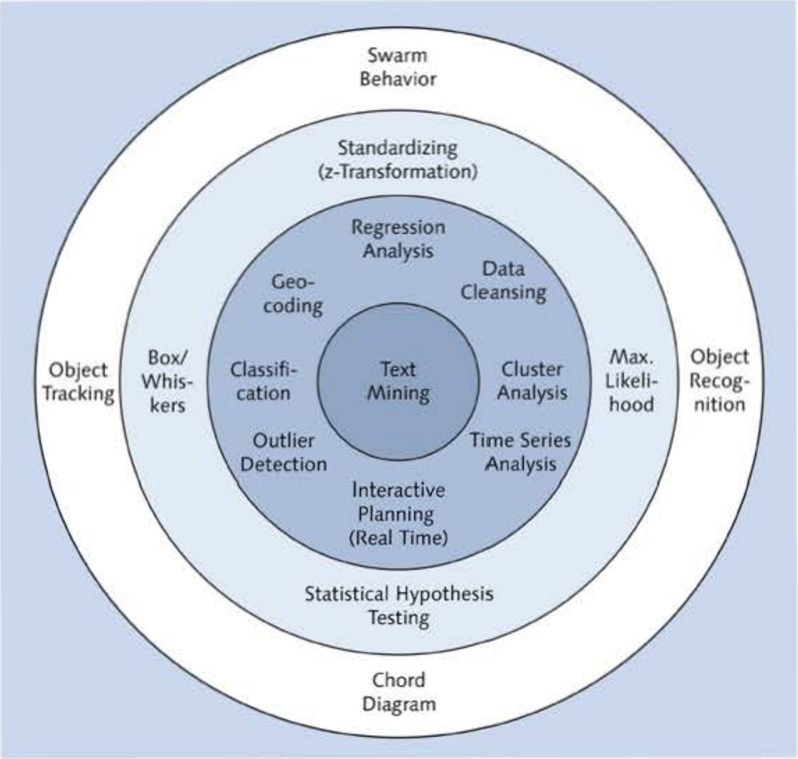


Figure 2.10 SAP HANA Algorithms

Methods and Architectures

As a new solution, SAP HANA contains quite a few innovative approaches and methods. The idea to execute write operations in a separate storage area and later (asynchronously) consolidate them with the rest of the database via a so-called *delta merge* is just one of them. But SAP has not reinvented the wheel; in terms of methods and architectures, SAP—very much like all other big data suppliers—is drawing from a giant, publicly available pool. Figure 2.11 provides you with an overview of SAP HANA–related methods.

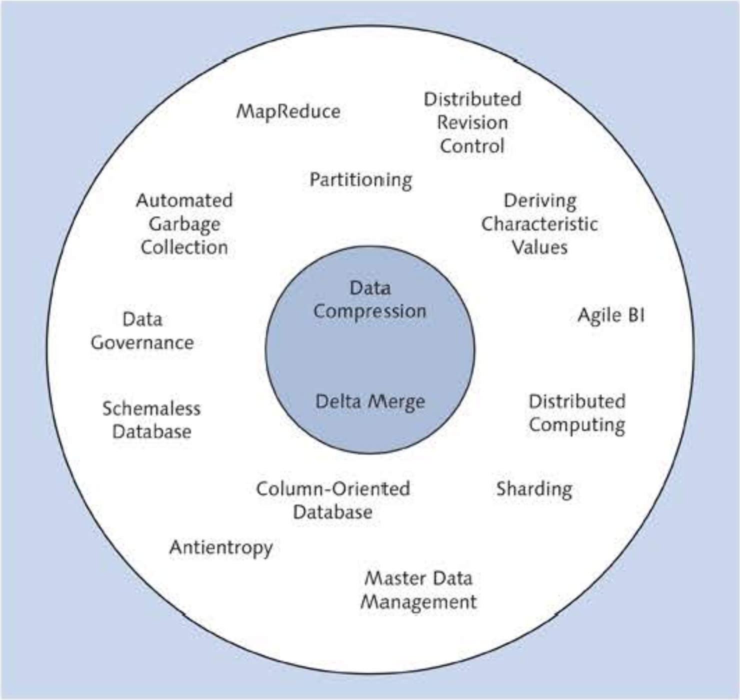


Figure 2.11 SAP HANA Methods

Methods from the public domain

With some of these generally available methodological approaches, SAP has done nothing but use them for its own purposes. Two examples of this are *hash* and *round robin*, two methods SAP (and others) use to partition large database tables. Hash and round robin refer to partitioning methods used to spread data evenly around a number of partitions. Other approaches (such as column-oriented data storage) have been extended and made more flexible by SAP; in SAP HANA, one can, for example, store the same data in columns as well as in rows.

SAP HANA patents

There are even a couple of SAP HANA–specific methods that are patent protected. This includes approaches for high-performance analysis of OLTP data (called ETL-Less Zero-Redundancy System and Method for Reporting OLTP Data, one of many things developed by Alexander Zeier, one of the masterminds in the world of SAP HANA). If you would like to know more about SAP's own innovations in the world of in-memory databases, just use Google's patent search function (*www.google.com/patents*), and enter "Hasso-Plattner-Institute Fur Softwaresystemtechnik Gmbh" as a search term.

SAP variants of reference architectures

The state of play in terms of architectures (see Figure 2.12) is much the same as with methods. There are some SAP-specific approaches (such as the LSA++ reference architecture for SAP BW); in the end, however, all SAP HANA–related framework architectures are based upon generally available ideas—sometimes extended or enhanced by SAP.

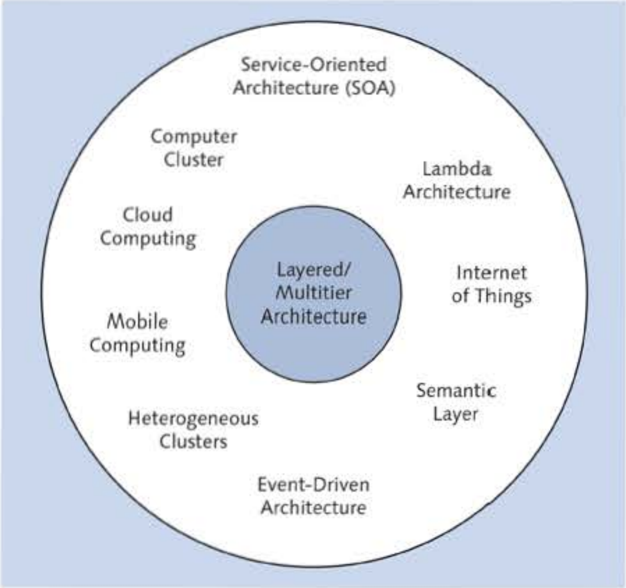


Figure 2.12 SAP HANA Architectures

SAP's toolbox for service-oriented architectures, the Enterprise Architecture Framework (EAF), is an extended, SAP-specific version of The Open

Group Architecture Framework (TOGAF). Alternatively, if you want to develop real-time solutions that can make decisions and act autonomously, you could use non-SAP approaches, such as the Lambda Architecture (LA; a two-layered model for real-time systems) to design SAP HANA-based solutions.

Or—to put it in layman's terms—you can use SAP architecture frameworks, or alternatively non-SAP architecture frameworks, to design SAP HANA-based solutions.

2.1.3 The Difference between SAP HANA and Big Data

SAP HANA as a subset of big data

So far, we have looked at SAP HANA as if it was just a subset of big data, which is not really the whole story. Now we are going to introduce some special features of SAP HANA. In this section, we are going to focus on properties, options, benefits, and advantages that go beyond what we have suggested so far.

Before doing that, let us summarize a couple of insights from Section 2.1.1 and Section 2.1.2:

- **SAP HANA is (in principle) open**
  When designing SAP HANA-based big data solutions, you are (in principle) free to use all technologies, algorithms, methods, and architectures that are at your disposal in the great wide big data world.
- **Limitations: hardware and database**
  Compared with a big data solution on an open-source architecture, there are two limitations:
  - *Selecting hardware*
    When selecting your hardware, you are not entirely free. However, this apparently restrictive limitation is really not as bad as it seems, for the following reasons:
    - The circle of hardware partners is in constant flux. Many important suppliers—apart from SAP's archfiend Oracle—are represented here.
    - With other proprietary offerings (such as Oracle Exadata), there are also (sometimes even more restrictive) constraints in terms of hardware.

  - Due to the close partnership between suppliers of hardware and software, you should expect fewer compatibility problems; in terms of performance, hardware and software can be attuned particularly well.
  - The history of personal computers (Apple versus Microsoft) demonstrates that even end users reward stability, comfort, and performance and are willing to dig deeper into their pockets if fewer problems are to be expected.
- *(Relational) database system*
  - SAP HANA mainly consists of a relational in-memory database (which is maintained and accessed via SAP's own variant of SQL) plus some admin tools, which means that using SAP HANA as a basis for core applications also leads to a specific, predetermined setting in terms of databases in your company in general.
  - From a maintenance and knowhow perspective, you will probably try to unify your platforms, also running SAP Business Suite or SAP BW on top of SAP HANA. Other databases would then only be deployed if you need certain (non-SAP) applications that cannot run on SAP HANA, for which the migration process is cumbersome, or if you need to satisfy certain requirements SAP HANA cannot cover (such as storing data in NoSQL databases).

- **Canned algorithms**
  SAP HANA and some solutions sold by SAP on top of it come with a number of well-established algorithms; this drastically reduces the effort needed to develop solutions that can—at least to a certain extent—benefit from such standard algorithms (like those for text mining or classification). This statement, however, only applies to algorithms and not any kind of preimplemented support for certain methods or architectures. Furthermore, the greater the competitive advantage you are expecting from a new solution the less likely it is that standard algorithms will suffice to build it.
- **Apps for special requirements**
  In terms of applications, SAP is following an approach similar to Apple's App Store, making it easy for developers to write applications for SAP HANA. SAP and its partners provide the development environments

and the information essential for app development either free or at relatively low cost, thereby enabling smaller companies and even individuals to write apps for SAP HANA. It could also be a smart move to advertise a project on a crowdsourcing portal, asking people to bid for the development of ready-to-use apps for SAP HANA (which is probably one reason why SAP has set up such a portal).

[Ex]

SAP HANA for Developers

The following three offerings are examples of how developers can access SAP HANA at a relatively low cost:

- **Amazon Web Services**
  At *http://aws.amazon.com/sap*, Amazon offers a variety of SAP solutions for use in cloud-based environments, including SAP HANA One (the pure database without ERP, data warehousing, or BI functionalities).
- **SAP HANA Developer Edition**
  Especially for developers, there is a licensing model at *http://scn.sap.com/docs/DOC-31722*. For nonproductive use, developers are able to access SAP HANA One in a number of partner environments.
- **SAP BW powered by SAP HANA: trial offer**
  For those who don't only want to see SAP HANA One but also want a data warehousing environment, there is another trial offer: a complete SAP BW powered by SAP HANA, including sample scenarios and an SAP BusinessObjects BI frontend for up to 30 days, to be found at *www.saphana.com/docs/DOC-3954*.

This palette of offerings is in a constant state of flux.

SAP's range of products for big data

Returning to our good old pizza analogy, examine the diagram shown in Figure 2.13. With hardware, you have to accept minor restrictions compared to open-source solutions, such as Apache Derby. With languages, platforms, and algorithms, you actually get a couple of advantages compared to generic big data solutions by way of preimplemented options. When it comes to methods and architectures, although SAP does not give you a lot, it also doesn't squeeze you into a limited set of SAP-certified approaches.

Understanding the added value of SAP HANA

Having looked at all the areas in which SAP HANA is *less* than generic open-source big data, we would now like to see where SAP HANA can also be *more* than that. Let us turn the tables and investigate whether SAP

HANA contains functions, components, or advantages that—in generic applications—are hard to get or make.

- **Specific (preimplemented) functions**
  Does SAP HANA's database have functionalities that other in-memory solutions struggle to provide?
- **Application logic in the database layer**
  Which additional opportunities arise from the fact that SAP HANA annihilates the strict separation between database and application layers?
- **(Meta) data integration with SAP's products**
  Does the fact that SAP HANA is one of many SAP products generate additional advantages (via synergies or via easy integration)?
- **(Meta) data integration with non-SAP products**
  How open is SAP HANA? Can toolchains that are based upon SAP HANA be built more easily than others whose foundations are corresponding open-source products?

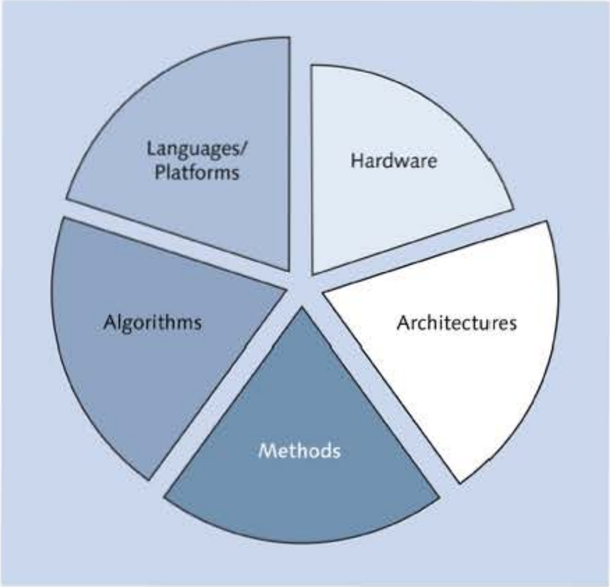


Figure 2.13 SAP HANA: Your Solution

Benchmarking is difficult

Within this book, we are not going to discuss any kind of performance measurement or benchmarking; such measurements only apply to very specific architectures, settings, and data volumes. If a certain supplier should be seeded in your environment, convincing your CIO by non-company-specific benchmarks that he has made a terrible mistake will be pretty difficult anyway; you would be steering too hard into the wind, which is exactly what we want to protect you from.

Specific (Preimplemented) Functions

Ease of use

Although SAP HANA and related products are delivering a couple of innovations, none of them stands head and shoulders above the rest (which are often available for free). We have, for example, not (yet) spotted anything in SAP PAL that is not available (or couldn't be built with relatively little effort) in R. Instead, one of the key advantages of products such as SAP PAL is their ease of use. Complicated algorithms can often be used without having in-depth statistical knowledge of all the parameters involved; furthermore, data can often be processed as they are without the need to transform them first via other functions.

However, remember that building an application that applies a couple of SAP PAL's functions to data generated by SAP Business Suite is not going to sweep your competitors off the market via its pure existence (in the end, you competitor could buy SAP PAL as well). With standard algorithms, you are just going to save some implementation effort.

Minimum requirements redefined

Nevertheless, the fact that sophisticated tools from inferential statistics are now oozing into the leading business software package (SAP) is going to redefine minimum requirements in IT, very much like all disruptive technological innovations do.

If you had to book a flight some 20 years ago, you visited or called your local travel agent; today, airlines or main street travel agencies without fancy online shops do not stand a chance. The idea of what is absolutely indispensable in a major organization's IT has changed a bit since Herman Hollerith started to use punch cards to process the results of the US population census in 1890, and common requirements will again change dramatically with big data. This change is not about processing speed but about the fact that basic statistical algorithms in data analysis—until recently the domain of very large organizations with huge server farms—are going to become as common as using an IT system to print pay stubs.

[+]

Faster, Cheaper, Less Risk

Quite a few well-established and proven standard algorithms are (so far) still not common in all industries. If you are the first to use them in your industry, size range, region, or market segment, then even standard algorithms can give you a head start, and cashing in on your business case with preimplemented standard algorithms comes fast, at limited cost, and with relatively little risk.

Application Logic in the Database Layer

Fuzzy border between application and database

One of SAP HANA's distinctive features is blurring the boundaries between application and database. Integrating processing logic into the database via SQLScript was the first step. Unlike classic SQL, SQLScript is not a declarative language; instead SQLScript (directly or indirectly via embedded procedures in other languages) also provides you with elements of imperative or functional programming languages.

[«]

Declarative versus Imperative Programming Languages

Declarative programming languages work by describing a problem instead of proceduralizing how to solve it. Thus, they focus on "what"; generating the "how"—that is, the control flow—is left to the language's implementation. Well-known declarative languages include Erlang, Prolog, and SQL.

They differ from *imperative programming languages,* which consist of instructions or statements telling the computer to execute certain tasks or changing the state of a program. To get to a desired result via an imperative language means coming up with a sequence of actions telling the machine *how* to do it. Some examples of languages that tend to be imperative rather than declarative include all *assembly languages*, *FORTRAN*, and *Pascal*—that is, older rather than newer languages.

Most new languages, such as C++ or Java, are hybrids containing declarative as well as imperative elements. And because, for example, SQL statements are often embedded in code written in other languages (for example, into Java via Java Database Connectivity [JDBC]—a Java database interface that connects to databases, sending SQL statements to and receiving responses from the databases), most code is a mixture of declarative and imperative elements.

In addition to these two, there are other types of programming languages/paradigms, such as functional programming (which is defined in Chapter 4).

New programming language: RDL

Further innovations in terms of blurring the limits between database and application logic and in terms of declarative programming are to be expected from SAP. One of the software giant's more recent brainchildren is the new *River Definition Language* (*RDL*; see also Section 2.3.1). RDL is a kind of metalanguage used to implement complete solutions within the database layer and facilitating such implementations by focusing on an even more abstract version of the "what" than most declarative languages.

Advantages of RDL

Compared to big data outside of the SAP world, the idea of moving logic to the database layer produces (at least) five clearly visible advantages:

- **Design match of application and database**

  Once application and database grow together and parts of the application's logic (as with SAP Business Suite) are moved to the database layer, buying both products from different software vendors might not make that much sense any more. If you are already an SAP customer, you may still have other (persistent or in-memory) databases running underneath your SAP ERP solution, but using SAP HANA instead would provide you with greater performance gains than the products of other suppliers or open-source options. The reason for that is pretty simple: a lot of the code in SAP Business Suite has its roots in the 1970s and was never meant to make full use of the benefits provided by in-memory databases and distributed systems. Hence, SAP's classic solutions are often anything but brilliant in terms of scalability.

  Moving logic into the database is SAP's way of doing away with this problem. But if you go for an in-memory database other than SAP HANA, the programs in your SAP ERP solutions will still remain unchanged and *database agnostic* (see Section 5.4.3); you will only get the benefits of reviewed SAP code if this code can, at the same time, access functions that have in the meantime been moved into SAP HANA for faster processing.

  There is another reason to go for the highest possible level of integration. Having all your business data and all your analytical capabilities in the same environment can save you a lot of worries in the areas of data logistics and interfaces. For companies using SAP Business Suite or SAP BW, it therefore becomes increasingly attractive to use SAP HANA instead of other in-memory databases. In the same way, SAP HANA customers might consider integration an incentive to also get their ERP solutions from SAP.

- **Facilitating custom development without limiting your options**
  These benefits arising from a higher level of integration and design matching between application and database mainly occur with standard business applications and relational databases but not in areas in which SAP does not provide (state-of-the-art) solutions (see Section 2.1.2). This restriction refers, for example, to non-ERP solutions or to company-specific big data applications and algorithms.

  In a certain way, however, you can have the omelet and still not break the eggs with SAP HANA. Although you can reap the rewards of a better match between application and database and also use preimplemented functions from SAP PAL, you can still use your database in SAP HANA with your own custom apps built in ABAP, JavaScript, R, or Python. In the same way, you are free to visualize the data held in SAP HANA using SAP BusinessObjects BI, R, or any open-source product. If you use SAP Business Suite and/or SAP HANA, you may increase your level of dependency on SAP in terms of business applications or databases, but you are not (yet) restricted when it comes to other areas.

- **Reusability of database-level components**
  When merging the application and database layers, SAP goes two steps further than just moving parts of the program logic into SQLScript procedures. On the one hand, functions on the database layer are meant to become reusable building blocks that can be shared between a number of SAP- and non-SAP applications. On the other hand, future application development—for example, using the metalanguage RDL—will then become a matter of describing input and output data rather than defining the logic between them by imperative code.

  SAP has been pursuing this approach for quite some time (for example, with the introduction of SAP EAF) and has already created a couple of related tools (for example, System Landscape Directory [SLD] or SAP process model in ARIS). Such tools are intended to enable the replacement of programming by configuration (which has been one of SAP's key concepts from the company's early days onwards); their market penetration has, however, been limited so far, not least due to substantial deficits in terms of integration and user-friendliness.

  If—in the long run—SAP should indeed be able to establish such tools with a growing number of customers, this would lead to a paradigm shift comparable to the difference between building a house brick by brick and configuring a prefabricated building online. If we stick with this example, there will always be enough builder–owners who have individual wishes and the money it takes to have these wishes put into practice, but the majority of commercial and private building projects might decide to go for cheaper systems of building—especially if, as with some suppliers of more exclusive prefab houses, there is still enough room for individuality.

- **Maximum performance through distributed applications**
  Another advantage of IT systems building lies in the fact that building elements can be fabricated in parallel. The roofers will then no longer have to wait for the walls to be built but can start covering the roof trusses in a factory building right away. Furthermore, if you don't tell the roofers *how* to do it but just specify the roof's structure instead (like you do with RDL), then they themselves can take care of all influencing factors within their domain and optimize their production flow independent of central demands.

  This detail is also why declarative structures are ideal for the efficient allocation of tasks in distributed systems. Programmers do not have to worry about certain performance requirements; instead, they just have to break down the "what" into portions that the system can digest. The underlying systems then automatically generate the "how," autonomously taking care of dispatching the work to individual processes and boxes.

- **Metadata transparency**
  If reusable logic is described via declarative programming and moved to the database, this also leads to higher transparency. With imperative code that has been divided into very small sections, seeing how individual instructions or function modules (SAP's version of a subroutine) fit into the bigger picture is often extremely difficult. With declarative code residing in a database, this is often a lot easier, because intermediate results are stored in persistent or virtual layers sitting between procedures. In SAP HANA, calculation views can be used for this (we are going to explain calculation views in Chapter 4, Section 4.5.2).

  If you know SAP BW, you will know that partitioned data flows with many persistent (for example, DataStore objects) or virtual (for example, InfoSources) intermediate results and without any ABAP code in start or end routines are often easier to understand and to parallelize than complex, monolithical program logic that has been embedded into an extractor's function module. Furthermore, the logic of data flows without embedded program code can be easily translated into metadata. Instead, ABAP code is some kind of black box from a metadata perspective.

Some Terms from SAP BW and ABAP

[«]

DataStore objects (DSOs) are used in SAP BW to persistently store large amounts of data. DSOs consist of a couple of relational database tables and primarily come into play as data targets delivering maximum write rather than read performance. In long, complex data flows, DSOs also often serve as a kind of clipboard, storing intermediary results between two processing steps.

An *InfoSource* in SAP BW acts as a nonpersistent intermediary layer. It therefore sometimes serves the same purpose as a DSO and thus can be considered its nonpersistent twin. InfoSources are used instead of DSOs if there is no reason to persistently store intermediary results.

Transformations modify, delete, or create data records in the course of a data flow. They might also be applied when it comes to enriching or cleansing data. Transformations in SAP BW serve a similar purpose to transformations within SAP Data Services. Both object types are, however, not (technically) identical; SAP BW transformations cannot be used within SAP Data Services and vice versa, and both types of transformations provide different functional ranges.

A *start routine* is a piece of code sitting within a transformation; it modifies data *before* the transformation as such is applied, which means that it modifies the data records before they arrive at the transformation as input data; start routines are sometimes used to generate additional data records (for example, by splitting existing ones). Technically speaking, a start routine is a local ABAP class.

End routines are also implemented within a transformation; in addition to start routines, however, they modify the data *after* the transformation has been executed, working with the transformation's output instead of providing input to it. End routines are sometimes used to check the validity of transformed data. In technical terms, end routines as well are local ABAP classes.

Function modules are ABAP procedures that can be called from other ABAP programs; they are used to encapsulate reusable, generic functionalities (such as calculations in financial mathematics).

Extractors procure data from source systems (in most cases SAP ERP or SAP Business Suite). An extractor mainly consists of a *structure* (describing the data records delivered by it) and a function module reading data from the source system (often via SQL statements) and handing them over to SAP BW.

[»]

Metadata

Metadata are data about data. The term can easily be explained using the example of a library:

- A library contains books.
- If the library was a database, the texts within these books would be its data.

Let's assume one of the library's books is Douglas Adams' *The Hitchhiker's Guide to the Galaxy*, quoted at the beginning of this chapter. There could be a wide variety of metadata about this text:

- In which format (paperback, bound, e-book, etc.) does the library hold the book?
- How many copies are there?
- Which of the copies are on loan and until when?
- At which location (hall, rack) can each available copy be found?
- When did each copy arrive in the library?
- In what condition is each copy?
- When was each copy last borrowed?
- Who has borrowed each copy and during which period?
- What other texts contain quotes from *The Hitchhiker's Guide to the Galaxy* or allusions to it?

Summary: potential benefits

The benefits resulting from moving application logic to the database layer can be summarized as follows: applications built on top of generic procedures within SAP HANA are faster and can be developed more quickly and modified more easily without losing track in an environment that changes at a tearing pace. This might not always lead to brand-new ideas for revolutionary business cases, but it does still facilitate the rapid, cheap, and low-risk implementation of existing ones.

(Meta) Data Integration with SAP's Products

SAP HANA within SAP's product portfolio

SAP HANA is one of many products in SAP's portfolio. Apart from the business applications it is famous for (SAP Business Suite), SAP also offers solutions for data warehousing, business intelligence, enterprise information management (EIM), cloud and mobile computing, and a number of technical platforms and services (hosting, consulting, and so on).

Isolated and looked upon as individual products, many of these solutions are not the cream of the crop in their respective product categories. Experienced SAP users know that the classic user interface of SAP's core application SAP ERP is a child of the 1980s, staying far behind (in terms of user-friendliness and pleasant graphics) the experience offered by most apps on their tablets. SAP itself is well aware of this criticism; SAP Fiori is one of SAP's initiatives to address this issue.

Functional bandwidth and integration

The excellence of individual solutions has never been SAP's unique selling proposition; instead, their strength lies in the bandwidth of their portfolio and in the integration among the various products. In many of the product categories served by SAP, there are competitors that have superior solutions; at the same time, there is no other provider of ERP solutions that can offer such a broad spectrum of products and a comparable level of integration between them. At the best, one might think of Oracle in this context. Figure 2.14 shows the structure of SAP's product space (and the level of integration between these products by the proximity of the planets to each other) from our point of view. We are going to talk about each of the areas within that space in detail in the following paragraphs.

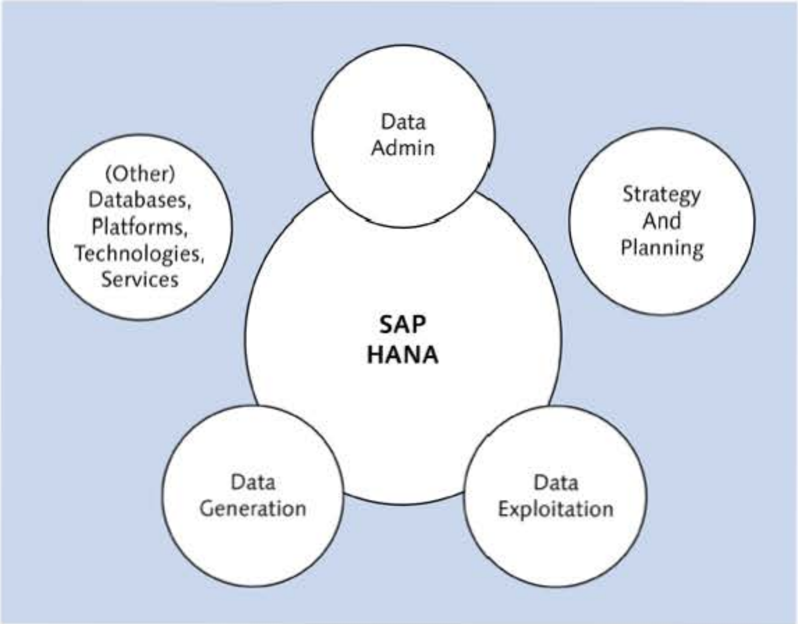


Figure 2.14 SAP's Product Space

Six product categories

SAP is continuously restructuring their product portfolio; the latest version can always be found at *www.sap.com/pc/index.html*. Because SAP's view on the structure of their products has always been subject to ongoing change, we use our own six categories to classify their solutions; in the past, these categories have turned out to be a bit more stable than (SAP) terms such as "financials" or "business intelligence."

Products for Data Generation (Data Collection, Data Processing, and Triggering Actions)

Collecting and generating data, acting on the basis of data

SAP Business Suite (currently assigned to BUSINESS APPLICATIONS at the preceding link to SAP's product website) and the corresponding products for small- and medium-sized enterprises are *products for data generation*. Products for data generation fulfill four tasks:

- Collect data that are relevant for business processes
- Process these data
- Generate more data from these data (by processing them)
- Trigger actions using these data (by sending messages)

Most products for data generation also include reporting functionalities; the great majority of these reports are plain vanilla lists of individual or aggregated data records. Only a few reports in products for data generation support the user when it comes to gaining new insights or even take this job from him.

Let us get back to the library example we used to define metadata in the "Application Logic in the Database Layer" section. If the texts in the books are the data, then data generation means writing these texts, printing and binding the books, and getting them into the library. In the same way as the text in a book is based upon either the author's ideas or texts from other books, products for data generation use external or internal data as input to generate more data.

[+]

SAP ERP is More than Data Generation

In a world of mostly automated business processes, solutions like SAP Business Suite do things that go well beyond collecting, processing, or generating data; they also (more or less autonomously) make decisions and act without human interference.

An ERP system does not have a physical body and cannot walk out of the factory gates and shoot up the neighborhood like the cyborgs in the famous *Terminator* movie series of the 1980s (featuring Arnold Schwarzenegger). It does, however (very much like the computer network Skynet in these movies), send messages and instructions to other systems or even humans and is therefore able to come into effect in the physical world via *agents*. For reasons of simplicity, we have also included this function as part of products for data generation.

Distribution Center [Ex]

In large distribution centers, articles are transferred to stock and picked from stock by automated storage and retrieval systems (ASRS or AS/RS) or people. Even if humans do the job, the decisions about what should be stored where, when and in which sequence articles should be picked from the warehouse, and via which transport route they are meant to be sent to whom aren't usually decisions made by employees on the shop floor.

Instead, warehouse- or transport-management systems, such as SAP Extended Warehouse Management (EWM) or SAP Transportation Management (TM), decide. As a rule, humans working in the warehouse are not authorized to make these decisions and usually do not question the instructions they receive via tablets or other devices.

Theoretically, decisions made by the system are monitored, supervised, and cross-checked by experts in the company's sales and distribution department, much like a crew in the cockpit of a commercial airliner supervises the autopilot. But let's be honest: What's the point of automating such processes if humans still have to review each single step? In practice, having seen the system work, people tend to rely on the computer's higher intelligence, like you do with the navigation device in your car. They follow without question.

If you just landed at San Francisco International on a foggy day, you can be sure that the flight captain at the front of the plane has not performed the landing himself. Today, landings at very low visibility (so-called CAT III approaches) must—with only a few exceptions—no longer be done by humans but have to be left to the plane's autopilot!

You may welcome such trends or fear them as harbingers of the final takeover of machines; for good or ill, big data solutions are going to give this trend—which has been visible for quite a while—far more penetration power and range. Big data is going to create the basis for automating more and also more complex real-time decisions than ever before, at the same time dramatically improving decision quality.

Big data enables flexible rules

The key difference from existing systems (which are also able to decide and act but do so on the basis of fixed customizing or fixed rules) is this: the *decision rules themselves* can be monitored and adapted continuously and in real time. This does not only refer to the setting of switches in a rule but also to the question of which parameters (i.e., switches) a rule is based upon. Clearly, we are finally approaching the old vision of self-programming computers. In this vision, IT experts no longer customize systems but instead are responsible for metacustomizing (driving the control mechanisms, which in return generate new rules, thus customizing themselves).

[Ex]

Source List and Quota Arrangements

Within its purchasing module (MM-PUR), SAP ERP can automatically determine sources of supply for material requirements (such as purchase requisitions). Two of many parameters controlling the automatic determination of sources of supply are the source list and quota arrangements. The *source list* specifies the allowed sources of a material for a certain plant within a certain period. For materials with more than one source of supply, *quota arrangements* can define which share of the required quantity should be purchased from which source.

Building on this, one could imagine a system in which the source list and quota arrangements are no longer maintained manually but automatically. This automatic maintenance would not—and this is the vital point—be based upon static rules but rather upon a self-learning solution that can, for example, do the following:

- Continuously monitor a multitude of key figures generated within vendor evaluation
- Detect dependencies between quantities ordered from a certain supplier and that supplier's performance
- Take into account future material requirements
- Suggest/make changes to source list and quota arrangements to optimize whatever related value drivers might be relevant

One interesting point here: you don't need to know *in advance* whether or not there actually is some kind of link between quantity ordered and supplier performance and what such a link might look like; if the system has been intelligently customized, it is going to find out for itself!

We can't and won't keep quiet about the fact that fully automated systems can—from time to time—get things completely wrong. Algorithmic trading on financial markets has already caused the odd minicrash that costs financial institutions billions. But humans have the same problem; grave mistakes may therefore impede or delay trends but not stop them.

Products for Data Management (Data Logistics and Metadata)

Products for data management take care of data logistics and metadata administration. An organization that uses many different solutions (no matter whether they are all from the same vendor or not) will have to move data back and forth between these solutions (or at least make one solution's data visible to another one). On the way, data are often reformatted, transcoded, enriched, or cleansed. Sometimes, the quality of data is measured or improved as they move from one system to another.

Data logistics

All these tasks are summarized under the term *data logistics* and are often taken care of by ETL tools. Another group of solutions we also rank among data logistics are tools for data replication (for example, SAP Replication Server—previously known as Sybase Replication Server).

[+]

Renaming of Sybase Products

At the beginning of 2014, SAP renamed some products from their Sybase portfolio. In all new product names, the term Sybase has been dropped.

Metadata management

If large data volumes are held, managed, and moved around, then there is often a lot of metadata as well. Where are data stored, where do they come from, and to where are they supposed to flow? Administering these metadata is the second part of metadata management. Metadata management is the job of metadata repositories. SAP's metadata repository is called SAP Information Steward.

A metadata repository is a database containing data that describe other data (such as information regarding their format, origin, or meaning or the use of a field within a database table). Going back to our library analogy, if texts in a library are data, then a library's data logistics (moving books around) would rest upon the shoulders of employees or be based upon library trolleys or networks of chutes and tracks, and its metadata repository would be the library catalogue. On SAP's website, tools related to data management are mainly listed under DATABASE & TECHNOLOGY; some can also be found under ANALYTICS.

[»]

Enterprise Information Management

Enterprise information management doesn't deal with an organization's data but instead with how an organization handles data in general. Typical operational objectives of enterprise information management include the following:

- Data quality and measuring data quality
- Data consistency and measuring data consistency
- Metadata quality and measuring metadata quality
- Metadata consistency and measuring metadata consistency

From a strategic perspective, these objectives serve two purposes:

- Transparency and traceability of data flows and data processing (*compliance*)
- Empowerment of end users in terms of data science and decentralization of data analysis and data exploration

SAP offers a number of solutions for enterprise information management. Details can be found at *www.sap.com/pc/tech/enterprise-information-management/software/overview.html*. There is also an SAP PRESS book on the topic: *Enterprise Information Management with SAP* (Brague et al., 2014).

Products for Data Storage (and Data Preprocessing)

Storing data

Much like a library needs to store its books (usually in racks), data must be stored somewhere (usually in databases). There are various SAP solutions (databases and data warehouses; the latter use databases to store their data) used to keep data; related *products for data storage* can be found under DATABASE & TECHNOLOGY.

Preparing or preprocessing data

One of these databases is SAP HANA, although SAP HANA not only stores but also processes data (like some data warehouses do).

To continue our library analogy, this would be like the library having the technology to display on the shelves, on request, a summary of a book you are interested in reading, its author's name and short biography, its availability, its classification, any hyperlinks and other references that it contains, and a list of other similar books that you might also be interested in. All this would be available from the minute that the book is first put on the shelves.

Products for Data Exploitation

Generating shareholder value from data

We want to keep an eye on shareholder value, which is why we also classify *products for data exploitation* (rather than products for data analysis).

Data exploitation means analyzing data (automatically or by properly preparing them for human users) to generate new insights. In an ideal world, these insights lead to humans (or machines) performing, or refraining from, certain actions. Ideally, such actions have an impact on value drivers, in the end increasing shareholder value.

[Ex]

Manually Exploiting Customer Classification

A sales employee analyzing web logs and sales data notes a high correlation between the retention time of customers on your website and the total revenue realized from these customers throughout the following 12 months. You therefore decide to use retention time to group your customers into 20 customer classes and to store this information in field `KNA1-KUKLA` (maintained via Transaction XD02) in SAP ERP's customer master record. You then use this field to control various marketing measures (such as the distribution of your quarterly newsletter containing individualized special offers).

To update the information in your customer master records, a classification report is executed monthly in SAP Crystal Reports (a reporting solution within SAP BusinessObjects BI). The report delivers the actual assignment of customers to customer classes; the result is downloaded and serves as the basis for a batch input (a method for mass data maintenance within SAP ERP) linked to Transaction XD02.

Laborious data flow

The preceding example corresponds to a data flow that has the following sequence of steps:

1. System to user: A human working on a system is doing some free-style analysis and finds out that there is a correlation between retention time and sales.
2. User to system: A human designs the classification report and implements it in the system.
3. System to user: The system (regularly and automatically) delivers the classification report, and a human downloads it.
4. User to system: A human formats the new classifications, prepares the batch input, and triggers it in the system.
5. User to system: A human triggers a marketing campaign, in the course of which the system uses the classification data to select receivers or to calculate discounts.

The whole thing looks like a fairly exhausting and error-prone can of worms with many interfaces, system discontinuities, and manual activities.

[Ex]

Automating the Data Flow and Follow-Up Activities

Some of the previously mentioned steps could easily be eliminated—even without big data or SAP HANA. The classification report could update the customer master record directly, without any downloads or uploads (class `CMD_EI_API`); marketing activities that are based upon customer classification could be triggered automatically (perhaps monthly) and without any further human intervention.

In our example, once the correlation has been detected and the whole process has been implemented, the process chain would shrink to simply *system to system*. Human intervention would no longer be required and the process would be more efficient, probably even more effective, and definitely less error prone. Admittedly, automating processes is neither new nor rocket science; with the preceding process, doing what we have just explained was possible some 20 years ago (in this case by using function module `SD_CUSTOMER_MAINTAIN_ALL` instead of the class `CMD_EI_API`).

Long latency times and out-of-date rules

But even if the process had been automated that way, it would still have two weaknesses:

- A customer's classification would be updated periodically and via a batch process. If a customer's surfing behavior changes today, this might have an effect on his master data for about a month (at the time of the next update run) and—in the worst possible case—not have an effect on any communication sent to him for a total of four months (at the end of the next marketing cycle after the update of his master record). By that time—if he is driven by personalized special offers—he might have made a purchase from your competitor ages ago. This first weakness is often called *latency* as well.
- Your customers' behavior is subject to continuous change. After a short while, retention time might no longer have any explanatory power for their longer-term propensity to buy. Although your competitors long ago found out that an individual's ordering frequency is a far better indicator for future revenues, your time-honored system

bravely generates its customer classification year in and year out using rules long beyond repair.

Step-change with big data

This is where big data comes into play. As already mentioned in our example about source list and quota arrangements in the "Products for Data Generation (Data Collection, Data Processing, and Triggering Actions)" section, the problem can be solved by harnessing processing speed and challenging algorithms.

Real-time analysis and real-time action

Using SAP HANA's calculation speed, you could have your monthly classification report (which might have taken three hours to complete in the past) executed every 10 seconds—which is pretty close to "constantly"—and instead of changing customer master data the report (which then wouldn't be a report any longer but instead a *bolt* within a *storm cluster*) from the system could send buying invitations with individualized prices on an ongoing basis (as the customer walks past the respective shops in a mall).

All of this also means that latency and the quarterly newsletter are dead.

[«]

Storm Cluster and Bolts

Storm is an open-source event processor released in 2011 and used to build distributed real-time applications. A storm-based cluster hosts one or more topologies of interfaces. A topology is a calculation process that can execute specific tasks and that once started theoretically runs forever (until stopped).

Topologies consist of spouts and bolts. *Spouts* are data streams delivering the input for bolts and are often fed via messaging systems such as Apache Kafka or RabbitMQ. *Bolts* are processes for which there are one or more spouts providing them with input; based upon that input, bolts then return one or more spouts as their output. In this model, bolts are the carriers of computational logic—that is, bolts are where processing takes place.

A typical example for a storm cluster would be a system filtering and processing information relevant for your organization from a messaging service such as Twitter (the Storm project was acquired by Twitter, which made it available as open source). Storm serves purposes similar to those of Complex Event Processing by SAP or SAP Event Stream Processor (ESP).

Real-time verification

At the same time, a t-test that is running over and over all the time could perform ongoing checks to verify whether your assumption of a high correlation between retention time and sales still holds and warn you if a shift is taking place. If you went for the chef's special, you could turn this whole thing into a system that can autonomously modify the classification algorithm and its input parameters on the basis of changing patterns (which is feasible with reasonable effort).

[»]

t-test

The *t-test* is a statistical test used to check whether two characteristics are (statistically, that is, not in terms of cause and effect) dependent. Like other statistical tests for similar purposes, the t-test is part of R's standard repertoire.

The answer to the question of which of many tests should be used under which circumstances is determined by a number of mathematical considerations—for example, whether one can assume that both characteristics are normally distributed.

Analogy: driverless cars

In Chapter 4's case study, we will further expand on the idea of a system monitoring the models it is based upon in real time. The notion of no longer manually defining the rules controlling your business processes might sound scary to you, but as we mentioned with the example of an automated CAT III approach in air traffic, today many processes have simply become so demanding that they cannot be handled by humans any longer. The same applies to modern fighter jets, which can no longer be flown without the help of automation.

How do you evaluate the risks of automated business processes, bearing in mind that you quite happily board a driverless train in airports such as Dusseldorf's, Frankfurt's, or Zurich's or in the city state of Singapore without giving it a second thought and then (together with your family) board a plane that will take off and land automatically, controlled by algorithms?

On SAP's website, most solutions for data exploitation can be found under ANALYTICS. Two examples of products for data exploitation are SAP Smart Meter Analytics (a specialized solution that can detect what is going on in a power grid and use this information to help manage the grid itself) and SAP Predictive Analysis (a generic forecasting solution). In a library, by the way, data exploitation would only take place in the heads of the users.

Products for Strategy and Planning

Developing and implementing strategies

SAP's products for strategic and operational planning can hardly be assigned to the basic categories (data generation/management/storage/ exploitation) defined so far. A planning solution like SAP Business Planning and Consolidation (BPC) is used to generate planning data as well as manage these data (for example, by assigning them to planning versions). The same applies to other, less generic planning solutions, such as SAP Sales and Operations Planning (SOP). On their own or in connection with SAP PAL, planning solutions such as SAP BPC or SAP SOP also come with certain functions for data analysis and exploitation (such as trend analysis, using historical data to generate a starting point for planning). Sometimes, planning solutions are even able to store data (that is, they have their own databases), or they run on another product for data storage (such as SAP BPC on SAP HANA).

A few examples of products supporting strategy and planning are SAP Strategy Management, SAP Profitability and Cost Management, and SAP Spend Performance Management. SAP Profitability and Cost Management and SAP Spend Performance Management are mainly focused on analyzing historical data and learning from them instead of supporting future decisions, in which case one could also consider them products for data exploitation. As we said at the very beginning of this section, planning-related solutions are hard to classify.

[◉]

SAP HANAfyability of Products

The more a product is used to generate, manage, or exploit data that are stored in structured schemata, the closer this product's existing (or future) integration with SAP HANA. When considering products that just store or archive (unstructured) data or help you model strategies (such as SAP Strategy Management), we see far fewer potential benefits from a close integration with SAP HANA.

Other Products (Databases, Platforms, Technology, and Services)

SAP's other products and services

In addition to everything that has been mentioned so far, SAP is still offering quite a few other products and services that are either based upon SAP HANA (like SAP's hosting offering, SAP HANA Enterprise Cloud), or are related to SAP HANA in some other shape or form.

As you can imagine, the more higgledy-piggledy your basket of products is getting, the more difficult it will be to stay on top of things, managing all these solutions for data generation, management, storage, and exploitation plus their respective metadata. Big data is aggravating this problem. Not only are you getting rid of persistent, intermediate layers of data (dealing with more virtual, ephemeral objects than ever before), but also, even worse, at the same time both the volume and the heterogeneity of your (often unstructured) data are increasing.

[👁]

Advantages from Integration with SAP Products

This is exactly where—from the perspective of existing SAP customers—the greatest advantages of SAP HANA (compared to other in-memory databases) are slumbering. SAP's products for data generation, management, storage, and exploitation are attuned to each other and can also (to a growing extent) access each other's metadata.

Furthermore, the high level of integration between all these products reduces the number of interfaces required. This makes your architecture more flexible, enabling you to implement any changes needed in processing structures. The option to change processes without too much ado and a lot faster than your competition (because you are able to quickly change the underlying IT) is an invaluable advantage. Unfortunately, even in relatively homogenous environments, this advantage is often given away for little or no good reason by creating chaotic data architectures.

Level of integration with acquired new additions to SAP's portfolio

Granted, many SAP products (SAP BusinessObjects BI, Sybase, and so on) have been added to SAP's portfolios via acquisitions of other companies, and the resulting product landscape, all under the same label and corporate identity on SAP's website, is all but homogeneous (for example, in terms of user interfaces and administration). SAP is addressing this and investing a lot of work, money, and effort into improving its products' uniformity with each new release. For example, SAP ensures that cross-system solutions (like solutions for data integration or enterprise information management) can communicate with all other relevant SAP solutions by default. This makes data management a lot easier.

Integration saves interfaces

A couple of examples will help explain how this objective is technically implemented. SAP's most important solutions for data generation (SAP Business Suite), data storage (SAP BW and SAP HANA), and data exploitation (SAP BusinessObjects Explorer, SAP Lumira, SAP PAL) can all run on an SAP HANA database in their "powered by SAP HANA" editions. Even other, slightly less common products (for example, SAP BPC) can be implemented on SAP HANA, and once all data you need are in the same SAP HANA database, there is no further need for interfaces.

Migration tools

Migrating existing databases in SAP Business Suite or in SAP BW into SAP HANA is relatively easy and is supported by a number of tools (for example, the Database Migration Option [DMO] of SAP Software Update Manager [SUM]; see *www.saphana.com/docs/DOC-2932*).

(Meta) data integration

There are a number of ways in which SAP HANA can be fed with data, for example, from the SAP Business Suite as follows:

- Real-time integration via SAP HANA Database Shared Library (DBSL), SAP HANA Direct Extractor Connection (DXC), or SAP Event Stream Processor (ESP)
- Near-real-time integration using SAP Landscape Transformation (SLT) Replication Server
- Near-real-time integration with SAP Replication Server (not recommended for SAP Business Suite; SLT Replication Server is the standard solution in that case)
- Periodic extraction of data with SAP Data Services
- Real-time, near-real-time, or periodic extraction via SAP BW's standard ETL functionalities from SAP Business Suite into an SAP HANA-based SAP BW

In addition, there are also ways to transfer pure metadata related to data in SAP Business Suite and from there via SAP Data Services or individual objects from SAP BW directly into SAP HANA.

Integration with metadata repository

Being able to exchange data and metadata between various databases and applications with very little effort is indeed fantastic. Unfortunately, for those of us who are over 40, experience shows that the easier it becomes to model data flows, the more of them will see the light of day in the long run, and the more unmanageable diversity and chaos can potentially be created! Human ingenuity is a wonderful thing, but for positive outcomes it needs to be harnessed and channeled, which is where using a structured approach to data architecture pays dividends.

SAP Information Steward

Quite a while ago, we introduced metadata repositories as a remedy for this problem. SAP's own metadata repository is called SAP Information Steward and does four different things:

- It collects metadata from different sources, merging them into one central directory.
- It produces statistical reports about the data in this central directory and searches the data for relationships/dependencies (among data as well as among datasets).
- It evaluates the quality of the data using predefined criteria and rules.
- It provides information about *data lineage* (for example, in reports); for each individual figure in a report, its data lineage can show you how this figure got there across system borders and how it has been calculated on the way.

Metadata for SAP and non-SAP systems

SAP Information Steward is not limited to digging up metadata from SAP systems (for example, from SAP BW); it can also take data from many other solutions. Data are procured via so called *metadata integrators* that provide you (as a standard) with access to data in the following formats and sources:

- SAP BusinessObjects BI
- SAP Business Warehouse (BW)
- SAP HANA
- Common Warehouse Metamodel (CWM), a multivendor standard developed by the Object Management Group (OMG) to describe and exchange metadata between data warehouses
- SAP Data Services
- The data dictionaries of various relational databases
- Meta Integration Model Bridge (MIMB), a technology developed by Meta Integration Technology Inc. to enable the exchange of metadata between different applications
- SAP Power Designer, a tool used for data and information modeling

Passing on metadata to other metadata repositories

By exporting metadata into Extensible Markup Language (XML), SAP Information Steward can also pass metadata on to user-defined, XML-enabled solutions. Details about SAP Information Steward (and metadata integrators) can be found within the respective administrator guides at *http://help.sap.com/businessobject/product_guides/sboIS42/en/is_422_admin_en.pdf*.

[«]

Extensible Markup Language (XML)

XML is a markup language (for a definition of markup languages, see Section 2.3) that makes it possible to describe hierarchically structured data as plain text.

XML is mostly used for exchanging data between different applications across system and platform limits when exchanging data via the Internet.

SAP HANA's hybrid storage model

Because SAP HANA is well integrated with other SAP products, exchanging metadata between SAP HANA and the rest of the SAP space is obviously easier than using another supplier's in-memory database. Due to its hybrid nature (in terms of how data are stored; SAP HANA is column as well as row oriented), SAP HANA delivers maximum performance with two types of applications:

- Online transaction processing (OLTP)—that is, products for data generation, such as SAP Business Suite
- Online analytical processing (OLAP)—that is, products for data exploitation, such as SAP BW

Many other in-memory databases have been designed primarily for use within the realms of BI and are therefore only column oriented. In contrast, SAP HANA is a habitat in which both OLTP and OLAP can thrive. This special feature enables you to implement solutions representing a self-contained, closed loop consisting of data generation, management, storage, and exploitation; such solutions can then evolve more or less autonomously, setting off an ever-faster upwards spiral. Such an idea goes far beyond the more traditional concept of investing a lot of blood, sweat, and tears into carving big data applications in stone while your competitors have already entered the next level of the arms race.

[◉]

Building Strategic Instead of Operational Applications

Apart from integration-related advantages, such as performance, cost, or risk reduction, this new perspective brings in a strategic dimension; you are suddenly able to develop business cases that are able to deal with more than just today's snapshot of the world out there and that are laid out to satisfy strategic business requirements and not to just attack their current symptoms.

(Meta) Data Integration with Non-SAP Products

Because SAP HANA can be fed with data via SAP Replication Server and SAP Data Services, there is hardly any pool of data that cannot be used as a source for SAP HANA.

Data acquisition with SAP Replication Server

SAP Replication Server can procure data from the following types of sources:

- SAP Adaptive Server Enterprise (ASE), a relational database originally developed by Sybase
- Oracle Database, Oracle's relational database
- DB2, a database system from IBM that is fairly common on mainframes

Data acquisition with SAP Data Services

SAP Data Services is able to cover an even wider array of sources (the following list does not claim to be complete):

- Attunity, a tool for data logistics
- DB2
- Hadoop
- HP Neoview, a data warehouse from Hewlett Packard that was withdrawn from sale in 2011
- Informix, a database system by IBM
- JDEdwards, an ERP solution from Oracle
- Microsoft SQL Server, a database system from Microsoft
- MySQL, an open-source database system from Oracle (there is also an enterprise version with associated costs)
- Netezza, a data warehousing appliance from IBM
- Open Database Connectivity (ODBC), a generic standard enabling access to many databases developed by Microsoft
- Oracle Database
- Oracle E-Business Suite, an ERP solution from Oracle
- PeopleSoft, an ERP solution from Oracle
- Salesforce.com, an SaaS (Software as a Service) customer relationship management (CRM) solution (used to manage customer data and relationships)

- SAP Customer Relationship Management (CRM), SAP's CRM solution
- SAP ERP
- SAP HANA
- SAP BW
- SAP R/3
- SAP Supplier Relationship Management (SRM), a solution used to manage supplier data/relationships and purchasing
- SAP Adaptive Server Enterprise
- SAP IQ, a business intelligence solution
- SAP SQL Anywhere, a solution used to synchronize (mobile) data
- Siebel, a CRM solution from Oracle
- Teradata, a database-management system

Exporting data to third-party applications

Furthermore, even when it comes to passing on data to third-party applications, SAP HANA is an extremely open system. Apart from the option to use SAP Data Services, you also have the following alternatives in terms of accessing data within SAP HANA (mainly intended for reporting):

- Business Intelligence Consumer Services (BICS), primarily designed for SAP BusinessObjects Explorer and other SAP BusinessObjects BI solutions
- Multidimensional Expressions (MDX) for all MDX-enabled reporting and visualization tools
- OLAP Business Application Programming Interface (BAPI) for other applications accessing SAP HANA via Remote Function Call (RFC) (for example, applications written in ABAP)
- SQL for other (non-SAP) programming languages accessing SAP HANA
- SAP HANA XS, a simple application server that can be accessed directly via HTTP and that makes it possible to connect to SAP HANA data from a browser (data are then transferred via the standard OData protocol [*www.odata.org*])

Last but not least, there also is the option to manage non-SAP metadata using SAP Information Steward. This is not a feature of SAP HANA but a functionality of SAP Information Steward. We mention it here because

SAP HANA can be used to either analyze existing metadata or to help gather them (also see Chapter 8). This in turn means that SAP HANA will use metadata from SAP or non-SAP sources or provide metadata to SAP or non-SAP applications.

Advantages of integration with non-SAP solutions

Thanks to solutions from Sybase and SAP BusinessObjects BI—which have now become part of the SAP portfolio—SAP HANA can team with all creatures great and small populating the big wide world of databases, ERP, or CRM solutions. This can make your life a lot easier (in case you were beginning to get a bit worried).

With a big data solution that you have built yourself using open-source components, interfaces (for example, to your ERP system) are not simply configurable but first need to be integrated and implemented. This means you'll need to know more about the logical links between the various relational database tables and the data models of your source systems.

Instead, if you choose to provide data from Salesforce.com to SAP HANA via SAP Data Services, you don't have to implement an interface program—just activate and configure it. If your needs change, adapting a configuration is usually easier and faster than revising source code, even more so if configuration can be done using a graphical, user-friendly modeling tool.

Advantages of SAP HANA Compared to Generic Big Data Solutions

In summary, we can consider SAP HANA to be a kind of subset of big data; in parallel we should, however, also acknowledge that SAP HANA comes with some unique features and advantages:

- SAP HANA comprises ready-to-use functions; with other (open-source) solutions, you would have to implement these solutions separately.
- SAP HANA is well integrated with other SAP applications (both in terms of data and metadata).
- For SAP HANA, there are EIM solutions that are connected to many other products in SAP's portfolio.
- Thanks to SAP Data Services, SAP HANA can easily send information to, or receive information from, other pools of data. A large number of interfaces are turnkey products that need to be configured but don't have to be conceived, designed, written, or implemented.

2.2 Implementation Scenarios for SAP HANA

Three groups of implementation scenarios

In general, SAP differentiates among three groups of implementation scenarios in which SAP HANA plays a major role:

- Replication (side-by-side) scenarios
- Integration scenarios
- Transformation scenarios

Framework architecture

Implementation scenarios can be considered architecture frameworks or templates. Depending on the type of business case you are dealing with, you will pick an implementation scenario (or a combination of scenarios) and animate it. *Animate* means the following:

- Let's assume you have come to the conclusion that the app scenario (one of the replication scenarios) best suits your needs.
- As its name suggests, apps (we will get back to apps later) are a key element in an app scenario, so to implement a solution based upon that template you will have to buy one or more apps on the SAP HANA marketplace or develop your own apps or have them developed for you.
- Once you have decided to use one or more specific apps, the resulting restrictions will—step-by-step—take you to a precise application and data architecture for your business case.

In Chapter 4 through Chapter 11, we will discuss in detail the factors that determine your decision of a certain scenario. We will also review how this fundamental decision limits your subsequent choices; in this context, we will present a couple of generic examples for application and data architectures. Before doing so, however, we would first like to illustrate the differences among the implementation scenarios, so now we will take a closer look at the three scenario groups and at the individual scenarios within each group. In the course of these more detailed discussions, you will notice that the transitions among the scenarios are smooth. Real-world scenarios will usually be hybrid forms representing a mixture of two or more pure scenarios.

The implementation scenarios we defined—and the respective diagrams—are derived from architecture models developed by SAP. We have unified these models and (in some respects) modified them a bit.

2.2.1 Replication Scenarios

Replicating data to SAP HANA

Instead of using SAP's term *side-by-side scenarios*, we are going to call this group *replication scenarios*. With replication scenarios, data from other databases are replicated into SAP HANA, which means that such scenarios have two key properties:

- Data are held redundantly.
- The data flow from (often persistent, relational) databases to SAP HANA is—more or less—down a one-way street.

Depending on what is needed for your specific business case, all data or only a part of them within your source are replicated and therefore held redundantly. Historically (yes, there is history even for in-memory databases), replication scenarios are the oldest way to use SAP HANA.

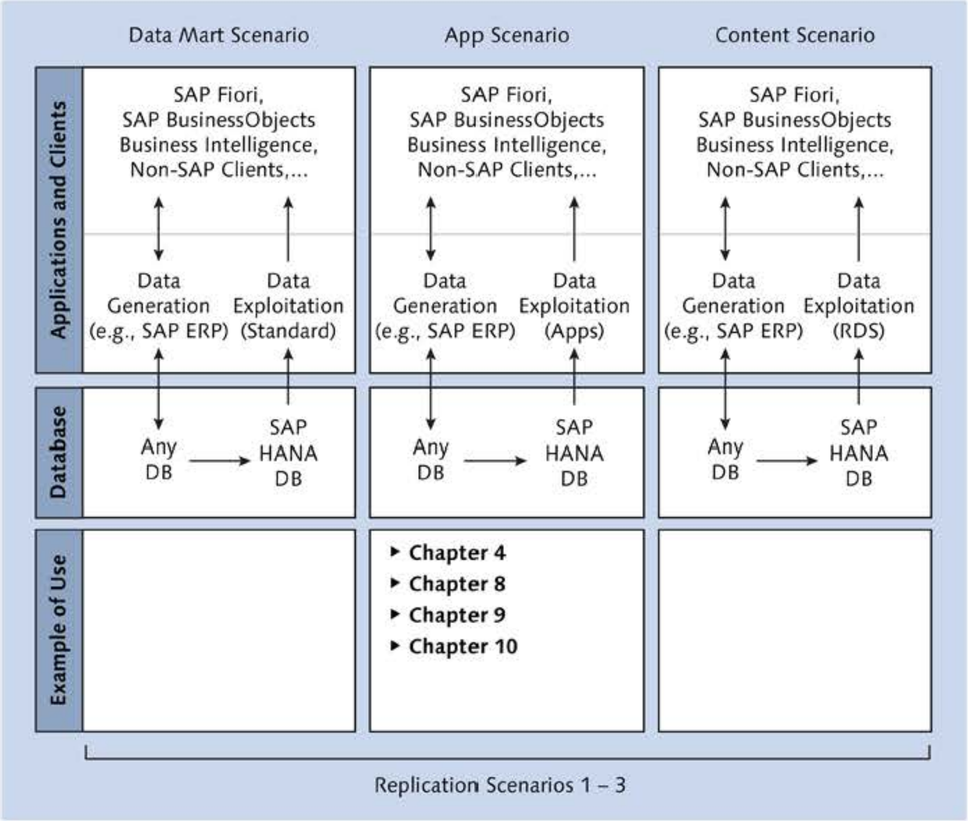


Figure 2.15 Replication Scenarios 1 – 3

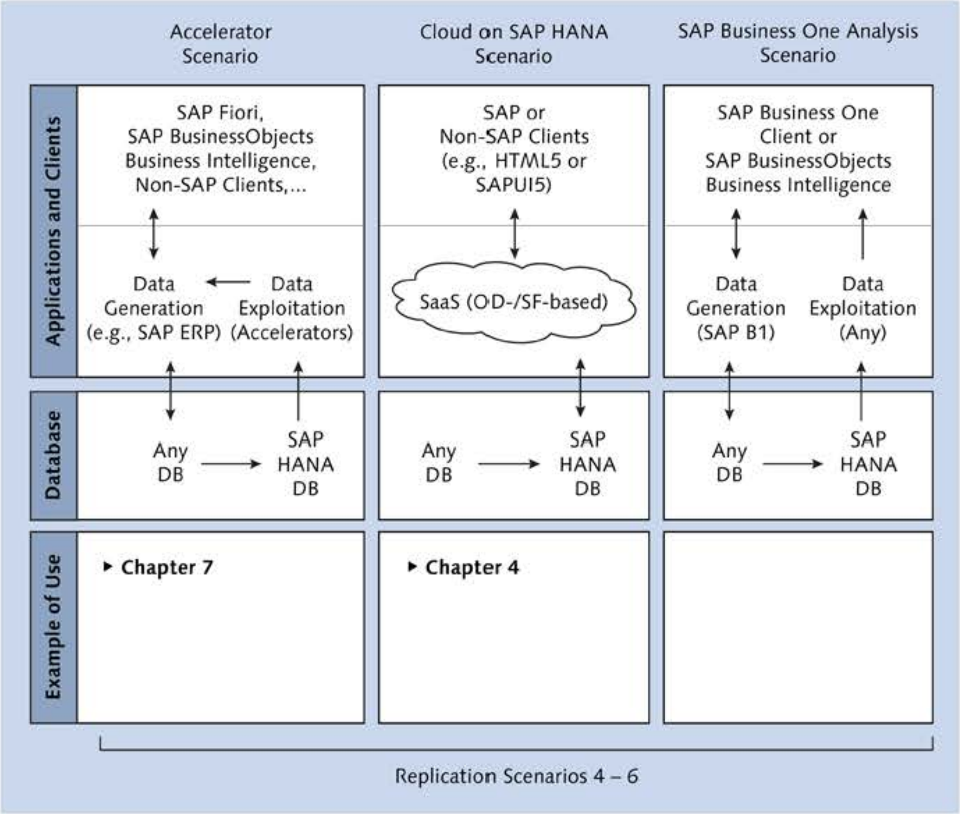

Accelerator Scenario
Cloud on SAP HANA Scenario
SAP Business One Analysis Scenario
Applications and Clients
Database
Example of Use
SAP Fiori, SAP BusinessObjects Business Intelligence, Non-SAP Clients,...
Data Generation (e.g., SAP ERP)
Data Exploitation (Accelerators)
Any DB
SAP HANA DB
▸ Chapter 7
SAP or Non-SAP Clients (e.g., HTML5 or SAPUI5)
SaaS (OD-/SF-based)
Any DB
SAP HANA DB
▸ Chapter 4
SAP Business One Client or SAP BusinessObjects Business Intelligence
Data Generation (SAP B1)
Data Exploitation (Any)
Any DB
SAP HANA DB
Replication Scenarios 4 – 6

Figure 2.16 Replication Scenarios 4 – 6

The replication scenarios group itself consists of six independent scenarios that we are going to examine in detail in this section:

- Data mart scenario
- App scenario
- Content scenario
- Accelerator scenario
- Cloud on SAP HANA scenario
- SAP Business One analysis scenario

Figure 2.15 and Figure 2.16 provide you with an initial overview of these scenarios.

Data Mart

SAP HANA as a data mart for agile BI

In this scenario, SAP HANA functions as a *data mart*. The term data mart usually refers to a subset of a data warehouse representing the interface between the data warehouse and the reports for a group of users. In the data mart scenario, data from SAP Business Suite, SAP BW, or other SAP or non-SAP solutions are brought together and consolidated in SAP HANA. These data are then analyzed or processed further using SAP BusinessObjects BI or other frontend products that serve more specific purposes.

A data mart provides the data basis for reporting, visualization, and analysis and should ideally enable users to play with the data (i.e., agile BI) without having to wait minutes or even hours for the system's response, which is why the data mart scenario is sometimes also called the *agile data mart scenario*. In most cases, standard tools for data exploitation (such as SAP BusinessObjects BI) will suffice.

Risks of this scenario

A major risk of this scenario lies in the fact the end users' enthusiasm created by superfast and user-friendly reporting will make them approach the team managing the data mart with a never-dwindling stream of new requirements. Soon the data mart is going to morph into a shadow data warehouse. This not only leads to redundancies and inconsistencies but also negates all efforts directed toward data safety, security, and quality. Data governance becomes nothing but an illusion, and chaos returns through a back door called the SAP HANA data mart.

Metadata repository as an antidote

There are two ways in which companies can face this challenge:

- They can make sure that SAP HANA is indeed only used as a data mart, getting its data only from the data warehouse and using structures (and metadata) that have been defined there.
- They can implement a strong metadata repository that automatically checks metadata for redundancy and inconsistencies. In the case studies in Chapter 6 and Chapter 8 we will take a closer look at the role of metadata repositories in an architecture including SAP HANA-based elements.

App

Data mart scenario plus real time

Simply speaking, the app scenario can be defined as *data mart scenario plus real time*, which is why SAP sometimes also calls this scenario the *operational data mart scenario*. In the data mart scenario, the data's up-to-datedness is less important. When dealing with a pure data mart that is fed with data via batch data flows that are processed in stages (in a data warehouse), real time would be a challenge anyway. In the app scenario, however, reaction times are key. Organizations use the app scenario to interpret, analyze, and exploit continuous data streams and to react to certain events instantaneously.

Example: sentiment detection

One example for this is sentiment detection:

- A customer posts a negative comment about your services on a portal, in a blog, on a social network, or via a messaging service.
- If you learn about this in real time, your own task force can react immediately, offer a solution to the customer, and immediately stop his statement from going viral on the Internet and producing an extreme backlash against your organization.

This also means that, unlike in the case of the data mart scenario, data exploitation cannot rely solely on standard list reporting or visualization tools but will need more specific (alerting) applications. We are going to have a closer look at such clients in Chapter 9.

Data warehouse not imperative

Including a data warehouse in this scenario isn't usually a great idea for two reasons:

- First, you would lose valuable reaction time. If your data warehouse is updated via daily load processes and then replicated to SAP HANA every 24 hours, in the worst case 48 hours go by between an event and your having any chance to react to it.
- You may want to scan millions or even billions of customer statements on the Internet; on the other hand, you may not want to clog up your data warehouse by storing data you just consider background noise throughout eternity. Not everything that is feasible with Hadoop in principle makes commercial sense.

With the app scenario, it often makes sense to provide data to SAP HANA not from a database but instead directly from an application. (Not everything your applications know at runtime is necessarily written into their database, especially because many of your applications were conceived at a time during which hard disk space was scarce and expensive).

Content

Data mart plus app for SAP Business Suite

From an architectural perspective, the content scenario is a combination of the data mart and app scenarios, which means that the key motive for using SAP HANA in this context could be requirements related to agile BI or requirements related to deciding and acting in real time.

The difference between the content scenario and both the data mart and the app scenarios lies in the fact that, with the content scenario, required data flows and reports are provided by SAP in a ready to use format; this detail limits the content scenario's scope to data acquired from SAP applications (mainly the SAP Business Suite). One example with fully implemented data flows and analysis tools is the SAP HANA Operational Reporting rapid-deployment solution, which comes with both preconfigured data acquisition and reports.

[+]

Differences between Content and Accelerator Scenarios

Some RDSs contain elements that are characteristic for both the content and the accelerator scenarios. This is why, in documents from SAP, content scenario and accelerator scenario are sometimes used more or less synonymously.

Accelerator

In all scenarios introduced so far, data from SAP or non-SAP applications were copied (replicated) to SAP HANA, enabling users to analyze data more comfortably, more flexibly, or more efficiently. The only difference among all three scenarios are the tools that were used to further process these data (what we called products for data exploitation).

Accelerator scenario serves applications not users

With the accelerator scenario, the difference from the other scenarios is, once again, data exploitation. What is special about this scenario, however, is not *how* data are processed but rather *who* is the receiver of information produced by products for data exploitation. With the data mart, app, and content scenarios, data exploitation fed data to users; the accelerator scenario, however, more often sends messages to applications rather than users.

In the accelerator scenario, a number of options can be used to exploit very large amounts of data very quickly, passing on the results of such analyses to an application. These applications consist of the following:

- SAP HANA only
- SAP HANA plus standard tools
- SAP HANA plus apps
- SAP HANA plus preconfigured content from SAP
- Other content providers

The target of this data flow might be the solution that originally sent the data to SAP HANA; there might, however, also be other business cases in which insights or messages end up elsewhere—that is, with other SAP or non-SAP solutions.

By the way: although writing data directly into another application's database (instead of sending it to the application, which would then trigger the necessary write processes) is often technically feasible, in practice such data flows are risky for many reasons and are therefore not a good practice. Whether dealing with SAP or non-SAP products, an application should always exclusively own its database and not have to deal with other products fiddling with it, even for the most noble of reasons.

Accelerator similar to customer exit/BAdI

In practice, the accelerator scenario can be compared to a customer exit that is called by an SAP application at a precisely defined point, inserting additional logic into SAP's standard code without threatening the system's upgradeability. As an example of such additional logic, when creating customer master records, a customer exit could be used to fill the (reserve) field `KATR1` in table `KNA1` (general data in customer master) with an automatically determined customer group.

In the meantime, this very popular option to enhance SAP's products has been replaced (as of release 4.6d) by *classic business add-ins* (*classic BAdIs*) and then (as of release 7.0 of SAP NetWeaver Application Server ABAP) by *new Business Add-Ins* (*new BAdIs*). Nevertheless, customer exits are still alive and kicking and very popular in the SAP world.

Differences from customer exits

There are, however, four important conceptual differences between customer exits and accelerator apps within the accelerator scenario:

- Customer exits represent a kind of bypass for the application. Instead of speeding the application up, they have the unpleasant habit of slowing it down; this is the opposite of what accelerators are meant to do, which is to relieve the application of a part of its workload, producing results that a (nondistributed) application (with limited scalability) might not be able to deliver itself for performance reasons.

- Customer exits are called within the context of a certain program and do not run independently of the transaction calling them (i.e., they don't run asynchronously). Accelerators can theoretically also be called from a transaction (for example, when asking an accelerator to check the customer's credit limit); the application then waits for their result. Asynchronous processing, however, is the more common option with accelerators when the accelerator is running periodically or more or less continuously in the background and providing data independent of whether a certain transaction has been called or not.
- The data generated by customer exits usually end up either in the application's (standard) database tables or in customer-specific tables (also called *Z-tables*) in an SAP database, but are only rarely written into the databases of other, non-SAP applications. Accelerators—in contrast—are not embedded in, nor closely related to, the process that uses the data they generate; they can therefore also play a major role in heterogeneous environments.
- If a customer exit generates data, these data often reflect the situation at the time that the customer exit was triggered—which is more or less identical to the time the calling transaction was running. The data generated by customer exits are therefore consistent with the rest of the database. Accelerators, however, are by definition implemented within a heavily distributed environment; the data they generate are therefore subject to the limitations of the CAP Theorem.

[»]

CAP Theorem (Brewer's Theorem) and BASE Requirements

The *CAP Theorem*—also known as *Brewer's Theorem*—states that in a distributed environment you cannot at the same time have the cake, give it away, and eat it. Or, more technically speaking, you cannot simultaneously guarantee consistency (C), availability (A), and partition tolerance (P).

Consistency in a distributed system would mean that at all points in time all of its nodes see exactly the same data. In this context, consistency not only means consistency between structurally or semantically identical data records (such as the postcode in one customer's master data) but also consistency across data records (such as the postcode in a customer's master data record and the number of all customers in a certain postcode sitting in another data record within another database on another server).

Consistency, according to the CAP Theorem, is therefore fundamentally different from consistency according to ACID requirements (see Chapter 1, Section 1.1.1), which is why CAP consistency is sometimes also called *external consistency* and ACID consistency called *internal consistency*.

Availability means that all requests to the system can always be answered. Partition tolerance means that the system can continue to work even if individual elements (partitions, network nodes, messages, etc.) fail.

Compared with nondistributed systems, the demands made on distributed environments are usually lower:

- In terms of consistency, in distributed systems one often makes do with weaker forms of it—for example, with so-called *eventual consistency* (E). Eventual consistency means that data will at some time—after a sufficiently long period without write requests or errors—be consistent across all nodes.
- In distributed systems, *basic availability* (BA) steps into the shoes of availability. Basic availability abstains from making each individual node fail-safe but instead makes a failure of the whole system (and therefore a partial or complete loss of data) extremely unlikely.
- Finally, one accepts the fact that data are not stored forever but instead have a limited lifespan (for example, because they are only held in main memory) and can (if required) always be restored from backups. This principle is called *soft state* (S).

Basic availability, soft state, and eventual consistency together lead to the so-called *BASE requirements* for distributed systems. Unlike ACID, BASE requirements *are* compatible with Brewer's Theorem and can therefore also be fulfilled in distributed systems.

Example: customer segmentation

One example of an accelerator scenario is SAP's RDS for customer segmentation (mentioned in Chapter 1, Section 1.2.2). Segmenting your customers with hindsight is of limited use, whereas adding continuous customer segmentation using SAP HANA can help you react to changes in customer behavior instantaneously. This does, however, imply that operational (OLTP) systems have up-to-date segmentation results at their disposal.

In distributed (non-ACID) systems, the possibility that two transactions within one and the same application, started at the same time but executed on data stored on different nodes, might behave differently (such as in one case making a special offer and in the other case not offering a discount to one and the same customer) is consciously accepted as the price to be paid for speed. This problem does not occur with SAP's RDS for customer segmentation, because segmentation results are still held in a (consistent) database owned by SAP CRM. However, this means that the assignment of customers to segments by this solution is less real time than it would be with a nonconsistent database.

Cloud on SAP HANA

Software as a Service

The cloud on SAP HANA scenario stands for an architectural variant in which a straightforward software solution reading data from an SAP HANA database is made available in a (public or private) cloud environment. Using such services is sometimes free of charge, or charges may be based either upon usage (on-demand [OD] model) or via a periodic subscription fee (subscription fee [SF] model). In both cases, customers are no longer paying for a license but for a service (which is why this kind of business model is also called *Software as a Service (SaaS)*. You will also find these terms in Figure 2.16.

[Ex]

Recalls Plus

One example of SaaS (in this case one that is free of charge) is Recalls Plus from SAP. Recalls Plus enables parents to access a comprehensive database of product warnings and product recalls via Facebook or an app residing on their tablets. Before buying something for their kids, they can check out the product and the company making it, and if they have bought something in the past they can monitor these products (in terms of future product warnings) via personal watch lists.

The significant characteristic of the cloud on SAP HANA scenario is that one or more applications that are part of your solution are running in a cloud. It is therefore irrelevant whether or not the SAP HANA database itself (on which some or all of your applications might be based) also resides there; this would, however, also be an option. Likewise, the question of whether you are using other sources of data besides SAP HANA is also irrelevant.

Differences from integration scenarios

The cloud on SAP HANA scenario should be differentiated from cloud-based integration scenarios (see Section 2.2.2). The cloud on SAP HANA scenario is based on an architecture in which only part of the SAP Business

Suite or SAP Business One (B1) is running on SAP HANA via SAP HANA Enterprise Cloud or Amazon Web Services. Furthermore, cloud on SAP HANA refers to an architectural variant in which rather slim (SAP or non-SAP) and often analytical solutions are running in the cloud.

In contrast to this, SAP now offers many of its products as a service that is either completely cloud based or comes as a hybrid solution (local installation plus cloud-based services). For example, under the label SAP HANA Enterprise Cloud, you could use SAP Business Suite as well as SAP BW in a scalable and stable cloud-based environment managed by SAP and its partners. (For more information about this, please see the recommended resources in the online appendix for this book.)

Differences from transformation scenarios

You should also clearly separate the cloud on SAP HANA scenario from transformation scenarios (see Section 2.2.3). These also often contain cloud-based or hybrid elements (just think of our example related to flight route optimization in Section 2.2.3). The key differences between them consist of the following:

- With the cloud on SAP HANA scenario, apps running in the cloud only read data from your SAP HANA database but usually do not store data there.
- With transformation scenarios, write access to the SAP HANA database is the normal case.

Compared to all other scenarios discussed so far (data mart, app, content, and accelerator), clients within the cloud on SAP HANA scenario often only provide very limited functionalities and are, as with SAP Recalls Plus, very easy to install or use. This is why such clients are often specialized, HTML5-based frontends rather than, for example, fully-fledged SAP graphical user interfaces (GUIs).

SAP Business One Analysis

SAP HANA for SAP Business One customers

In 2002, SAP purchased an Israel-based company called TopManage Financial Systems and rebranded their solution (originally sold in Israel under the name *Menahel*, meaning *supervisor*) as SAP Business One. Although different from SAP's other ERP products, this solution can also collaborate with SAP HANA.

The SAP Business One analysis scenario serves a purpose similar to that of the content scenario. Reports on data from SAP Business One do not read from the solution's database itself but instead from a mirrored database held in SAP HANA. This again provides you with much better reporting performance.

The key difference between the content scenario and the SAP Business One analysis scenario is the origin of relevant data. With the content scenario, we are talking about data from SAP Business Suite, whereas the SAP Business One analysis scenario is about data from SAP Business One. Very much like with the content scenario, the SAP Business One analysis scenario is about *reading* rather than *writing* data. In a case where data were also to be written and SAP HANA were SAP Business One's primary database, then you would be implementing SAP Business One on SAP HANA (see Section 2.2.2).

Tools for Replication Scenarios

Application's database is taking the lead

With all replication scenarios, the application's database (i.e., not SAP HANA) is taking the lead; SAP HANA just contains an image of some or all data held there. In the data mart, app, and content scenarios, there also is no data flow from SAP HANA to the products for data generation (for example, SAP Business Suite). Data are either replicated from the ERP's database to SAP HANA or are handed over to SAP HANA by the application itself; they do not flow back to SAP Business Suite, however, not even as a basis for reporting.

With the remaining three replication scenarios (accelerator, cloud on SAP HANA, and SAP Business One analysis), there are data flows from SAP HANA to the application due to, for example, the application (the product for data generation) reading data from SAP HANA. Such data are either only used by products for data generation at runtime (that is, not written to the application's database at all) or stored in the database belonging to products for data generation by its own programs (that is, not put there by an external process belonging to products for data exploitation). In a classic SAP environment, this still means that only the (ERP) application (SAP Business Suite)—not any supplementary solutions developed in SAP HANA—is writing data into the application's database.

SAP's toolbox for data replication

Copying (or more precisely *replicating*) data to SAP HANA is handled via specialized tools, such as SAP Replication Server or SLT; if more complex transformations of data between source databases and SAP HANA are required, using SAP Data Services would also be an option. However, the latter will lead to limitations in terms of real-time data provisioning.

If data are supposed to be sent to SAP HANA directly from the application, then SAP HANA Direct Extractor Connection (DXC) might be the right instrument for you. If you would like to update SAP HANA's database directly via a piece of AS ABAP code in your SAP Business Suite, then you could use SAP HANA Database Shared Library (DBSL) to do so. We won't discuss these products in detail here but would just like to mention them so that you can do some further research yourself. A good source of information about writing applications in AS ABAP on SAP HANA is *ABAP Development for SAP HANA* (Gahm, Schneider, and Westenberger, SAP PRESS, 2014).

2.2.2 Integration Scenarios

SAP HANA as the only database

With integration scenarios, SAP HANA becomes the primary database for an ERP system, a data warehouse, or other solutions writing data into it. With non-SAP solutions, writing makes the difference between integration scenarios and the app scenario (for the app scenario, see Section 2.2.1). In an integration scenario, SAP HANA does not contain a copy of relevant data but instead functions as a fully-fledged substitute for a classic, persistent, relational database-management system, such as Informix, Oracle Database, or SAP MaxDB. Data are read from and written to SAP HANA.

There are three individual integration scenarios:

- SAP Business Suite on SAP HANA scenario
- SAP Business One on SAP HANA scenario
- SAP BW on SAP HANA scenario

The first two scenarios are identical in terms of data flows; the only difference between them lies in the applications used as the products for data generation.

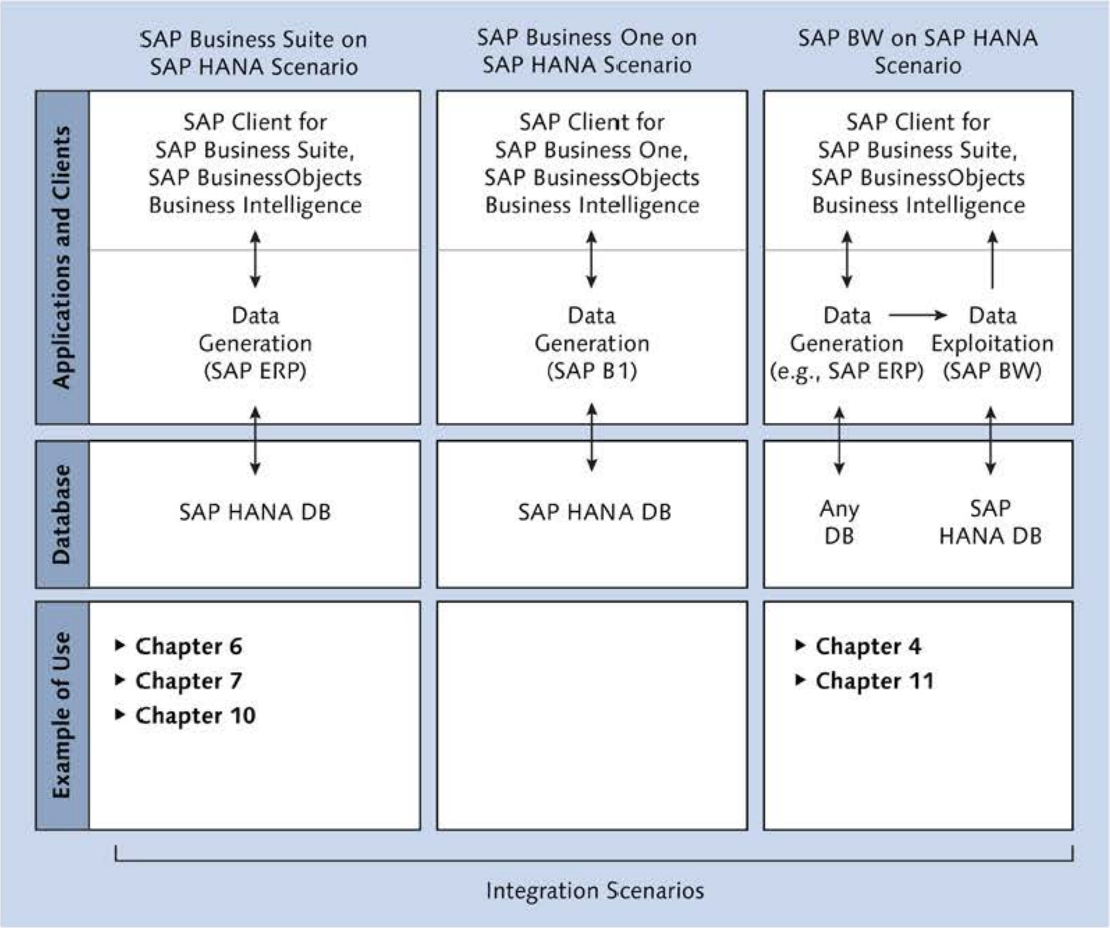

SAP Business Suite on SAP HANA Scenario
SAP Business One on SAP HANA Scenario
SAP BW on SAP HANA Scenario
Applications and Clients
Database
Example of Use
SAP Client for SAP Business Suite, SAP BusinessObjects Business Intelligence
Data Generation (SAP ERP)
SAP HANA DB
Chapter 6
Chapter 7
Chapter 10
SAP Client for SAP Business One, SAP BusinessObjects Business Intelligence
Data Generation (SAP B1)
SAP HANA DB
SAP Client for SAP Business Suite, SAP BusinessObjects Business Intelligence
Data Generation (e.g., SAP ERP)
Data Exploitation (SAP BW)
Any DB
SAP HANA DB
Chapter 4
Chapter 11
Integration Scenarios

Figure 2.17 Integration Scenarios

SAP Business Suite on SAP HANA

SAP Business Suite can remain database agnostic

SAP Business Suite and its core components, such as SAP ERP or SAP Customer Relationship Management (CRM), have always been database agnostic, meaning that they could be implemented on databases from many different suppliers (IBM, Microsoft, Oracle, and so on) without caring about the type of database residing underneath the application layer.

From that perspective, SAP HANA is just another database. Since 2013, customers have the additional choice of implementing SAP Business Suite powered by SAP HANA, meaning that the solution is going to sit on an SAP HANA in-memory database instead of a classic persistent database-management system. SAP calls the corresponding application architecture SAP Business Suite on SAP HANA.

So far, SAP HANA can only be used as a database with SAP ERP and SAP CRM but not (yet) for other building blocks of SAP Business Suite (such as SAP Product Lifecycle Management [PLM], SAP Supply Chain Management [SCM], or SAP Supplier Relationship Management [SRM]). For the sake of simplicity, we are for the moment going to ignore the additional advantages resulting from (standard SAP ERP) application logic on the database layer (see Section 2.1.3).

Combining SAP Business Suite on SAP HANA with other scenarios

Due to the openness of the underlying SAP NetWeaver AS platform, the SAP Business Suite on SAP HANA scenario could easily be combined with other architectural models that we have mentioned. Merging it with some of them, however, doesn't make much sense. Consider the content scenario: if your SAP Business Suite uses SAP HANA as its primary database, just duplicating a distributed, scalable solution does not promise any real improvements in terms of performance. In the end, if your cluster is too slow, then adding additional nodes, memory, or CPU cores to it would be easier than setting up replication data flows. If you see the need for a data mart, then you are facing the question of whether it might make more sense to set up the required structures in your existing SAP HANA database instead of administering another one. The latter option would probably create more problems than it solves; just think of data security and data governance.

SAP Business Suite on SAP HANA without SAP BW on SAP HANA [+]

It would, theoretically, be feasible to implement SAP Business Suite powered by SAP HANA and at the same time still continue with an SAP BW data Fettwarehouse running on a persistent database. Nevertheless, we wouldn't recommend doing that.

First, it would be strange if the reporting performance of your OLTP system was much higher than that of your OLAP solution (don't forget that one of the key reasons for having an OLAP-based data warehouse used to be reporting performance!). Furthermore, migrating an existing SAP BW system onto an SAP HANA database is in most cases a lot easier than migrating your complete ERP environment. If you have already moved your SAP Business Suite to SAP HANA, the more demanding part of the migration path already lies behind you. Why would you be fearful of walking up Ben Nevis if you have already conquered Mount Everest?

Also, keep in mind that one of the key assets of SAP HANA is the fact that it supports columnar as well as row-oriented data storage (see Section 2.1.3)—that is, analytical as well as operational applications. This enables you to set up operational applications that are very closely integrated with analytics and that use the results of analytical services in real time. If both your operational and your analytical systems dwell in one and the same database, then this integration will become even easier.

The third integration scenario in Figure 2.17 (SAP BW on SAP HANA) shows an architecture that is set up the other way around (SAP BW uses SAP HANA, but your SAP ERP is still on a traditional database); this is better than the reverse case, but in an ideal world the ANY DB on the database layer in the SAP BW on SAP HANA scenario would disappear and the SAP HANA DB on the right-hand side would serve both your SAP ERP and your SAP BW.

SAP Business One on SAP HANA

SAP HANA-based ERP for medium-sized enterprises

Using SAP HANA-based solutions isn't the exclusive right of big, multinational conglomerates. Even smaller or medium-sized enterprises can benefit from SAP HANA's extraordinary performance and the embedded statistical tools. SAP's solution for SMEs (small and medium-sized enterprises), SAP Business One, can also be bought as SAP Business One, version for SAP HANA, giving you a fully functional ERP system that can be used by minor organizations (though probably not by every mom-and-pop shop on the corner). The same applies to SAP Business Suite; SAP HANA simply replaces Microsoft SQL Server (so far, the most common database used with SAP Business One). SAP calls this architecture SAP Business One on SAP HANA.

[+]

SAP Business One and SAP BW

Theoretically, an SAP HANA-based SAP BW could still sit on top of your SAP HANA-based SAP Business One. But as SAP BW is not as widely spread among SAP Business One customers as it is with those using SAP Business Suite, we have not considered that option.

Most SAP Business One customers will probably only use reporting components of SAP Business One for data exploitation. As mentioned when defining products for data generation, we would consider such (limited) capabilities part of the product for data generation and not a product for data exploitation in its own right. Another option for data exploitation and frontends would be SAP BusinessObjects BI.

SAP BW on SAP HANA

Data warehousing with SAP HANA

SAP BW can also be implemented on SAP HANA, regardless of whether your products for data generation providing data to it are SAP HANA based or not. The corresponding variant of SAP BW is called *SAP BW powered by SAP HANA*, and the respective implementation scenario is known as SAP BW on SAP HANA.

Migrate, replicate, or copy metadata

As with SAP Business Suite, there are three different ways of making data or structures of SAP's data warehouse available in an SAP HANA environment:

- You could migrate your complete SAP BW into an SAP HANA environment. Once migrated, your SAP HANA-based data warehouse is going to look exactly the same as before, but it will be a lot faster. Migration as such is not a major issue; you only need to consider a couple of peculiarities with an SAP HANA-based SAP BW. One example: aggregates, a persistent, aggregated view on the data within an InfoCube, are unnecessary in SAP HANA and are therefore no longer supported with an SAP HANA-based SAP BW, nor are the process types updating such aggregates. *InfoCube* is SAP-speak for an *OLAP cube* in SAP BW, and *process types* are types of processing steps within a batch job or a job network (a so-called *process chain*) in SAP BW.
- In *SAP HANA Studio*—SAP HANA's administration and development workbench—you can import individual objects from an existing SAP BW system.
- Instead of importing objects, you could also decide to just copy their metadata, in which case you will get copies of the respective objects but not of their content.

When talking about the SAP BW on SAP HANA scenario, we are referring to the first of the above three options.

By the way: in this book we do not consider any kind of technology for data retraction. In SAP's world, *data retraction* refers to a couple of programs that can hand over data from SAP BW to a limited number of applications within SAP ERP. The perspectives provided by either sharing the same database between operational and analytical systems or by using decision tables (see Chapter 10, Section 10.4.2 for details) in SAP HANA go far beyond classic data retraction.

In principle, the SAP BW on SAP HANA scenario is pretty similar to the data mart scenario explained in Section 2.2.1. The main differences between these two scenarios consist of the following:

- With SAP BW on SAP HANA, you (could) build a complete data warehouse, not just a data mart.
- Instead of using any data warehousing solution (as a product for data exploitation), you use the SAP product SAP BW.
- Data are not replicated at the database level but handed over to SAP BW via SAP Business Content—that is, by extractor programs that have been developed by SAP and that take into account dependencies among database tables in SAP Business Suite. To reflect this, Figure 2.17 shows an additional data flow arrow between the products for data generation (SAP Business Suite) and the products for data exploitation (SAP BW). SAP BW then takes care of writing the data it receives via these extractor programs into its own database.

[+]

SAP BW on SAP HANA without SAP Business Suite on SAP HANA

As mentioned earlier, Figure 2.17 shows an environment in which SAP BW has been migrated to SAP HANA while the products for data generation are still running on their old (persistent) databases.

Although this is not ideal, you may want to live with this situation for a limited period; it all depends on the size of your migration project, your portfolio of projects in general, the resources you have, and your requirements when it comes to analyzing operational, granular data.

Impact of SAP HANA on Applications

Making applications leaner

With all three integration scenarios, one can rightly assume that using SAP HANA as the primary database will also lead to major changes to the applications running on top of it. The dissatisfactory speed of the database layer and the fact that the database layer used to be a lot less scalable than the application layer have always been two classic bottlenecks when optimizing the performance of SAP solutions.

Consequently, ABAP programs developed by SAP or their customers contained a lot of extra code, the only purpose of which was implementing workarounds for this issue (such as buffering database tables), trying to minimize its impact on end users. With SAP HANA, most of these workarounds have become obsolete and are now nothing but leftovers from the bad old days. In addition, reusable pieces of application logic are now trickling into the database layer, making the program code within the application layer leaner and faster.

Home-field advantage for SAP

The combination of both factors will dramatically improve the performance of SAP Business One, SAP Business Suite, and SAP BW—not only because they are now living on an in-memory database but also due to the design match mentioned in Section 2.1.3. We are sure that SAP is aware of the fact that taking control of the database layer is going to give them a significant home-field advantage. Sure, even if just for antitrust regulations, many SAP products will continue to run on other databases in the future; however, they might work a lot faster if implemented on SAP HANA instead.

Even if SAP in the long term decides to support other in-memory databases, there will probably still be the odd advantage to be gained from operating your SAP solutions on SAP HANA. Quoting King Edward III, "Shame to him who thinks evil of it." Or instead we could cite one of today's leaders, Bill Gates, who allegedly once said that "Windows is not complete as long as Netscape Navigator is still running on it."

2.2.3 Transformation Scenarios

SAP HANA as a basis for novel solutions

As with integration scenarios, with transformation scenarios, SAP HANA assumes the role of the primary and often the only database. But unlike integration scenarios, transformation scenarios are not intended to provide a habitat for SAP Business Suite or SAP BW or for apps related to these products. And unlike the app scenario, we are talking about apps that are a) totally new and b) can also write to the SAP HANA database.

Applications beyond the horizon of SAP's current portfolio

With transformation scenarios, SAP HANA can become the foundation for what we call *disruptive applications*. Disruptive applications are the software equivalent of *disruptive innovations*. Innovations are called disruptive if they create new markets, change the rules of the game, or have the potential to completely eliminate existing technologies, products, or services. Disruptive innovations or applications often establish new business models and new value networks. For now, there is only a transformation scenario, called *new SAP HANA apps* (see Figure 2.18).

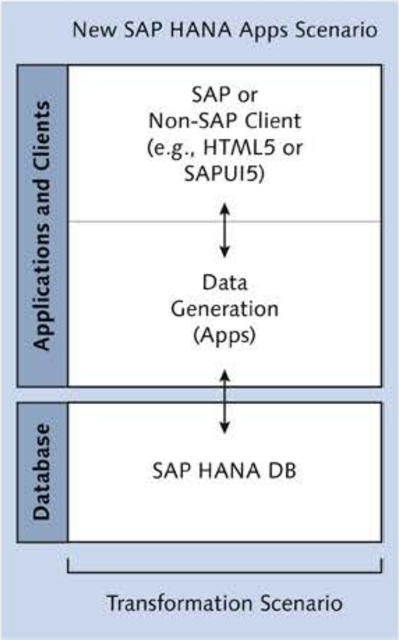


Figure 2.18 Transformation Scenarios

New SAP HANA Apps

Disruptive applications

The new SAP HANA apps scenario hosts one or more self-contained apps (which are not just accelerators supporting an ERP solution) on an SAP HANA database. The apps' application logic might be partially implemented within the database itself (that is, for example, in SQLScript procedures) and partially reside outside of it. Such apps are destined to perform analyses in batch, react to events in real time, hand over information to other apps, or write results into the SAP HANA database.

The special feature of this scenario is the fact that we are talking about new, disruptive apps—that is, apps whose functional scope goes well beyond that of standard solutions in SAP's portfolio. The key idea is that such apps—although potentially supplementing standard solutions—make new territory accessible to a significant extent.

[Ex]

SAP Smart Meter Analytics

SAP considers the app SAP Smart Meter Analytics to fit into this scenario; we, however, still see it as a pretty conventional application.

From our point of view, SAP Smart Meter Analytics would better inhabit the orbit of the replication scenario called *app scenario*. No doubt we are dealing with a sophisticated solution, but it's one that in the end isn't much more than a classic analytics tool.

Due to the openness of this architectural model, we can imagine that applications that are far more groundbreaking are going to surface in the near future; for example, an app that controls a fleet of autonomous delivery drones, thus revolutionizing logistical networks.

Mobile and hybrid architectures

Apps for transformation scenarios could be written in ABAP or Java using SAP NetWeaver AS and also on scores of other platforms and in numerous programming languages. A subset of the apps for a business case implemented on the basis of the new SAP HANA app scenario could be native apps on mobile devices like iPads or Android tablets. Such apps would then be written in Objective-C (an object-oriented extension of the programming language C used by Apple for iOs and MacOS) to run on iOS or in C++ or Java for Android.

We particularly like the idea of creating hybrid applications, the application logic of which is spread across SAP HANA itself, a cloud environment, and mobile devices. Such hybrid architecture would also be our first choice for the previously mentioned optimization of flight routes in real time.

[Ex]

Hybrid Architecture

A commercial airplane carries a large number of sensors that continuously collect data of all kinds. A single plane, however, does not have all data of all planes that are currently in the air at its disposal, nor does it know everything about routing-relevant constraints (such as a volcanic eruption 500 km north of the route, the prevailing wind directions near the volcano, and the resulting ash fall).

Theoretically, it would be conceivable to centrally collect these data and then transmit them to all planes, which could then use them as a basis for further calculations. Unfortunately, such an approach would be impractical and unsafe (has everybody received all data, and have there been no transmission errors?) and also hard to implement. To process all these data, each aircraft would have to have a powerful big data cluster in its freight compartment, which would mean carrying a lot of additional weight around and having to provide all these servers with loads of electrical energy.

A hybrid architecture would make much more sense:

- Some components hosting the data and the required calculation logic are on board, others are in a cloud on the ground (while the plane is in a cloud in the sky), and still others reside in a database system also sitting safely under the ground.
- The plane then sends preprocessed, preaggregated, and filtered data to the nearest ground-control node; at the same time, this node feeds the plane with data about routing options from which the computers on board can make a final selection based, for example, upon rapidly changing local weather conditions.

Strategic objectives with SAP

As we will show in Section 2.3, the fact that SAP's strategic priorities—big data, cloud computing, and mobile computing—are growing together in hybrid architectures is one of SAP's key long-term strengths.

2.3 Trends and Future Developments

Speculations

To be honest with you, we know nothing about what is going on within the top-secret development think tanks, circles, and labs at Walldorf and at Palo Alto. We know even less about what is going on in Hasso Plattner's mind (SAP's chairman of the supervisory board).

In the past, the "father" of SAP HANA (in the early days of SAP HANA, the acronym HANA was lightheartedly interpreted as "Hasso's New Architecture") would have the odd surprise for the SAP community in store from time to time. Even if we had any such knowledge, we would probably not be encouraged to share it with you or even admit we had it. But maybe one can at least come up with some plausible ideas about where big data and SAP HANA, plus some other innovations, are going to take us over the next five to ten years without intelligence operations or major bugging efforts.

2.3.1 Technology Trends

Strategic hotspots for SAP

SAP has often publicly declared that three areas are of particular importance for major organizations from their perspective:

- Cloud computing
- Mobile computing
- In-memory databases

To that list, you may want to add a fourth area that Plattner has put forward in an interview with the German *Wirtschaftswoche* in July 2013: user-friendliness and the customer experience, learning from Apple's example. The first outcomes of such efforts are initiatives such as SAP Fiori and SAP's own version of the markup language HTML5, called SAPUI5.

[«]

Hypertext Markup Language

The term *markup language* (HTML stands for *HyperText Markup Language*) originates from the printing trade. Markups were notes to tell typesetters what certain parts of the text were supposed to look like. Today, markup languages are no longer used just to mark up text; they mainly describe how to handle content.

Hypertext is text that links information via cross references (familiar to most of us due to Internet *hyperlinks*, the functionality of which is based on hypertext).

Trends that mutually amplify each other

Put cloud computing, mobile computing, in-memory databases, and usability into a preheated pan after deglazing it with a bit of accompanying background noise from media and conferences, stir it well, and let it simmer for a while on low heat; in the end, you'll get something that tastes like the concoction described next.

Acceptance for cloud computing increasing

In the future, solutions that are based upon SAP HANA will rarely be provided via pieces of software that have to be installed locally. Thanks to the whistle-blower Edward Snowden, all of us have learned that even German chancellor Merkel's cellphone is far from bug proof. In the light of this knowledge, quite a few top managers have just thrown their hands in the air and given up.

In corporate headquarters, both the desire to seal corporate data off from the rest of the world and reservations about storing data in the cloud are dwindling. Indeed, for those under 30, unveiling their most intimate secrets to the servers of Facebook and Twitter has become a habit. All of that also leads to a growing acceptance of cloud-based services for data storage and processing.

Saving storage space via the cloud

Now, although we (probably due to our ages) are still a bit skeptical in that respect, we have to admit there are strong practical arguments for the cloud. Why should a James Bond movie bought in the cloud absorb five gigabytes of memory on millions of computers at home if it can just as easily be stored once with Apple and streamed to users upon request? Why should each and every company in China store all valid addresses of 1.5 billion Chinese people on its servers to verify delivery addresses of new customers? Wouldn't it be easier to just put them into a centralized directory that is always consistent and up-to-date?

Fair enough; in the first case, Apple will know how often and at what times of the day you feel the urge to meet Daniel Craig or Pierce Brosnan, and when it comes to address data the company providing that service—if it has read and absorbed this book—will know which addresses you are requesting, who might therefore be a new customer for your business, how long they have been buying from you, and how well you are doing in terms of sales.

Extra risky? Not really; your competitors could gain similar insights by posting a webcam just on the other side of the road from your factory gate and making a tally chart about trucks entering your premises to tell them from whom you are receiving goods and how often.

Standard solutions are not trade secrets

Such considerations don't only apply to data but also to application logic. Although we have emphasized that you can generate shareholder value by using superior big data algorithms, this does not mean that 100% of your applications' logic is to be considered strictly confidential. Unless you were adventurous/mad enough to massively change it, the source code of SAP ERP Financials on your machines is probably identical to that on the boxes of your main competitor, so why write algorithms that separate followers from opinion leaders in social networks if you can buy the respective results from Klout?

Because of the extra effort involved, trying to reinvent the wheel could even be a competitive disadvantage, something like a value *diver* instead of a value *driver*. This means that big data solutions are often a combination of confidential and nonconfidential applications and data. With many seminal big data applications, you will be using hybrid architectures; your data and applications will be spread across clouds, smartphones, and tablets or even fridges, toasters, and coffee machines that are hooked up to the Internet.

Core data/ applications are different

This does not mean that we are stepping back from what we said in Chapter 1. Even in the preceding scenario, a killer app that coordinates all these components and generates your competitive advantages will remain as exclusive and proprietary as the recipes of your products, but even this killer app (as well as your recipes, by the way) will probably reside in a cloud for practical reasons—albeit a private and not a public one.

SAP is (once again) a hosting provider

It might not be pure coincidence that SAP—after a long break—has decided to (re) enter the hosting business with SAP HANA Enterprise Cloud. The decision seemed to be right; in 2014, SAP reported dramatic increases in their cloud business. SAP's offerings not only include cloud-based data storage and apps but also cloud-based licensing models for SAP HANA, solutions based upon SAP HANA, and other core SAP products.

Openness is key

In a hybrid world, solutions such as in-memory databases are more valuable the easier and the faster one can harness them to other products. We therefore believe (think/suppose/hope) that SAP will continue to facilitate the integration of open-source big data tools, as it has already done by allowing the inclusion of R code in SQLScript. Enabling SAP HANA to read data stored in HDFS was another essential step, and with other solutions (such as Storm) there are still opportunities for further integration. Whether SAP uses these opportunities or not might also depend on whether it offers competing products (SAP ESP in the case of Storm) and how concerned it is about losing market share for its own products.

Certain innovations can only be implemented in the cloud

If an ever-growing number of applications and devices are a) collecting data and b) forwarding these data into the Internet (*Internet of Things*), then this also leads to a massive increase in the volume of theoretically available data.

In addition to an increasing quantity of data, this also leads to a qualitative leap: the Internet, the cloud, or the pool of data as a whole knows a lot more than every single application/device on its own. This opens the door for new applications, which—from a conceptual point of view—can *only* be built in the cloud (think of our example regarding flight routes). Right now, we are just at the beginning of this revolution.

[»]

Internet of Things

Internet of Things describes the gradual coalescence of physical reality and the virtual world of the Internet. Not just smartphones but a growing number of objects of all kinds and sizes (wristwatches, cars, and so on) are equipped with sensors plus stationary or mobile Internet access.

Smartphones are already able to measure things such as proximity (to your face), movement/acceleration, environmental light, and position in space (via a gyroscopic sensor); driver assistance systems in cars are capturing the world around them via 3-D cameras, radar, and ultrasound sensors.

User-friendly means invisible

A man–machine interface is user-friendly if it serves users without them noticing it any more, comparable to a great British butler disappearing into the background like a shadow during confidential conversations but always back on the spot to top up the sherry. The principle "user-friendly = invisible" not only applies to hardware but to software as well. A speech-recognition or dictation system that needs hours of training or a news website that first needs to be configured according to our personal likings is not very user-friendly. Taking this into account, big data applications will also become increasingly important when it comes to recognizing and forecasting our preferences and to creating systems (be they for the TV, the smartphone, or the fridge) that have the potential to act almost autonomously.

Systems and metasystems

The more complex our world becomes, the more we will have to rely on systems controlling other systems (that is, metasystems). This trend leads to ever more layers or metalayers via which we will often communicate indirectly with solutions doing something for us.

Two potential perspectives within the world of SAP HANA (the first already available, the second still—but maybe only for a little while—a dream of the future) are SAP's new River Definition Language (RDL) for developing SAP HANA-based applications and assistants helping us to select algorithms.

[«]

River Definition Language (RDL)

RDL is a kind of metaprogramming language for developing SAP HANA-based applications. Until recently, solutions in SAP HANA had to be developed using L, R, or SQLScript or—using SAP HANA Extended Application Services (XS)—in JavaScript; RDL is a new and more convenient option.

At the same time, RDL further moves the focus of programmers from "how" to "what." One example: RDL can use a data model (or more specifically an entity relationship model [ERM]) to generate the respective objects (such as tables) within an SAP HANA database plus the code required to access (that is, to read, write, or update) data stored in these objects. In the end, RDL code is compiled to JavaScript or SQLScript, so existing code and the generated code, in both languages, can be reused.

[+]

Assistants Helping to Select Algorithms

We won't tire of pointing out that powerful tools such as SAP PAL also have their inherent dangers, which is one reason that John MacGregor's book *Predictive Analysis with SAP* (SAP PRESS, 2014) extensively discusses how to choose the right algorithm, each algorithm's strengths and weaknesses, and the parameters required for every single one of them.

But quite a few statistical algorithms do not only deliver results; they also provide you with key figures that help you assess the quality or reliability of their output. Examples for such key figures are the Pearson product-moment correlation coefficient with linear regression or the confidence interval with statistical tests. Even better, assumptions that have to be made to use certain algorithms (such as "normally distributed") can yet again be tested using other statistical tools. On top of that, the performance of distributed solutions like SAP HANA also enables you to run algorithms over and over again, using different input data, or to use multiple algorithms in parallel, letting the system find out which one works best.

Corresponding to metasystems and metalanguages, one could also think of *metaalgorithms*. Such metaalgorithms could help you select the algorithm that works best for you here and now and protect you from premature conclusions. With high-frequency stock trading, such metaalgorithms are nothing special any more. In a way, SAP HANA is introducing the spells of the wiz-kids of Wall Street into running everyday business processes; some companies will recognize this opportunity, and others will ignore it and go under.

2.3.2 Ideas Are Becoming the Critical Success Factor

Arms race with hardware and software

To summarize, you face the following challenges:

- Mighty algorithms are made available as instant, preimplemented, ready-to-use solutions.
- They are offered as a service in a cloud; everybody can use them on a pay-per-use basis (or even on a pay-per-benefit basis, as with Google's AdWords web advertising).
- In general, the costs for such services are rapidly decreasing as main memory and processing power become cheaper and cheaper.
- SAP is spreading SAP HANA–related knowhow for free all over the world.
- Brand-new, top-notch algorithms can be developed upon request on idea marketplaces like Kaggle, tapping into an unlimited potential of manpower and skills.
- Your competitors (and companies that are preparing to compete against you that aren't even on your radar yet) have entered into an arms race of ever-more-powerful hardware (number of CPU cores, number of nodes, and so on) and software (meta, meta-meta, meta-meta-metalayers and metaalgorithms).

In such an environment, you have two options:

- If you are sitting on a war chest filled to the rim, you may want to join the arms race, but remember what happened to the Soviet Union at the end of the Cold War.
- If not, you need to have better ideas than competitors, focusing on designing some big data guns while your opponents are still trying to find more efficient ways of cutting down forests that they want to turn into bows and arrows.

Industry definitions are getting blurred

Even for big organizations, the first option is very hard to maintain. Trading giant Amazon, for example, has long ago crossed the boundaries of just being a bookseller; they are also selling groceries and other goods of all kinds. They are also in the service industry (via M-Turk) and a major hosting provider (via AWS).

How long is it going to take until Sauron's (Amazon's, Apple's, or Google's) eye gazes on you? Have a look at *www.relentless.com* and find out who owns that domain name (unfortunately, the owner does not reside at Mordor but in the real world).

Ideas are your secret weapon

If you don't have unlimited financial resources, then your only choice is to come up with better ideas for big data, evaluate these ideas more precisely, turn them into architectures more quickly, and implement them faster than your competition.

The term *competition* should be interpreted broadly: you may not consider Amazon, Apple, and Google to be your competitors, but are you sure they are thinking the same way about you? Some methods for developing and evaluating business cases for big data were discussed in Chapter 1.

The future is about owning data

Although quite a few organizations have yet to understand (we are sure that yours is an exception!) that ideas are the thing via which big data is creating competitive advantage, this is still just a passing phase. In the near future, even the most brilliant ideas won't help you unless you also have (that is, *own*) the data you could exploit with them. Therefore, take care of two things now:

- Get hold of as much data as you can (rest assured, you'll find ways to use them later).
- Do not get stuck only generating ideas that help you make money; also have ideas that help you collect customer- or process-related data.

In this chapter, we spent a bit of time carefully reviewing tools and implementation scenarios that will support your business cases. In a couple of case studies, we are going to bring both aspects (business cases and implementation scenarios) together, enabling you to pick the architecture you need for your specific purposes. Before doing so, however, we would like to discuss a couple of areas in which big data solutions are especially good at quickly and efficiently creating shareholder value. We are going to follow the structure of SAP's solutions.

Viewed from a distance, everything is beautiful.

Tacitus, Annals

3 SAP HANA in Specific Industries and Business Processes

Due to chronic traffic indigestion, Derek had pulled off the motorway near Birmensdorf. For more than an hour, he had been creeping along with the sea of cars through a commercial desert consisting of nothing but workshops, industrial units, gas stations, and fast food restaurants, all of that from time to time punctuated by the neon-red flare of the odd village pub. Previously, when traveling to the technology park at Sophia Antipolis, he had always taken the shorter Autoroute du Soleil, but because it was the end of the school holidays in France and also because he had never been there before he had decided to take a four-hour detour via Switzerland this time. Not a great idea.

Figure 3.1 Lake Stelli and Matterhorn, Canton of Wallis, Switzerland

As a travel destination, he had always found the Swiss confederation too expensive. In his fantasies, however, Switzerland had been an idyll comprised of deep-blue lakes and majestic mountain ranges under which gnomes *secretly watched over legendary treasures. Juicy alpine pastures would lie on the slopes of the hills. Under a crystal-clear sky, lusty dairy farmers were busy turning milk into delicious cheeses, yodeling into the sunset after their daily work was done. The gray faces in the cars in front of and behind him bore no resemblance to either mountain air or folk songs.*

In the next traffic jam, near Koelliken, a big, elongated building next to the motorway had caught his attention. As there was no sign of movement anywhere, he grabbed his smartphone and searched online for "Koelliken." The results at the top of the hit list talked about a hazardous waste landfill; the giant hall housed the decontamination of highly toxic waste, protecting it from rain that would have taken much of it into the ground water, all of it an expensive legacy of Switzerland's chemical and pharmaceutical industry. Koelliken, he continued to read, was located in the Aargau, proudly boasting itself the "nuclear canton"; four of the five nuclear power plants still operational in Switzerland were located here or very close to this canton's borders. One of them, Beznau I, held the dubious record of being the oldest one still running worldwide. And when continuing to search for "nuclear" and "Switzerland," Derek also learned that by the banks of the Broye—a beautiful river he hoped to cross after a few more hours spent in traffic jams—a cavern in the rocks didn't contain silver and gold but instead the contaminated and sealed remains of a test reactor. In 1969—long before Harrisburg, Chernobyl, or Fukushima—the world's first ever first nuclear meltdown had taken place near Lucens. Not really a reason to sing patriotic hymns to the fatherland.

Derek thought about the French comic hero Asterix and the Swiss writer Peter Bichsel. In Asterix with the Swiss, *a Roman legionary was standing at the Swiss border, thoughtfully quoting Tacitus: "Viewed from a distance, everything is beautiful." Bichsel once wrote that his fellow countrymen were the only people in the world that believed the lies of their own tourism brochures.*

Behind him, somebody impatiently honked his horn. Derek had failed to notice that the car ahead had moved on a few inches. Exchanging the business wasteland in the—officially—ugliest canton of Switzerland for the sterile monotony of Sophia Antipolis would take another seven or eight hours.

A bit more than a week ago, Derek had been in the middle of the Namib Desert. Far and wide, there were no houses, no vehicles, and no atomic ruins. Tomorrow, he would have to explain to a French customer's team how all his *new ideas would fit into their organically grown data and application architectures. From afar and in the bright light of Africa, everything had seemed so easy and apparent: big data, brainstorming, benefits, value drivers, and implementation scenarios. However, his customers near Antibes thought in terms of products and modules; their thoughts were tied to tables and the fields within them. It would be anything but easy to percolate even a fraction of his visions into the daily grind.*

Big data and SAP HANA

In Chapter 1, we tackled the question of how (that is, via which mode of action) and where (that is, in which types of business processes) big data solutions can create benefits and shareholder value.

In Chapter 2, we illustrated the—so far, relatively abstract—definition of big data, naming technologies, algorithms, methods, and architectures. We then explained the relationship between big data and SAP HANA. In doing so, we highlighted four items: specific (preimplemented) functions, application logic in the database layer, integration with SAP products, and integration with non-SAP products.

Additional (potential) benefits of SAP HANA

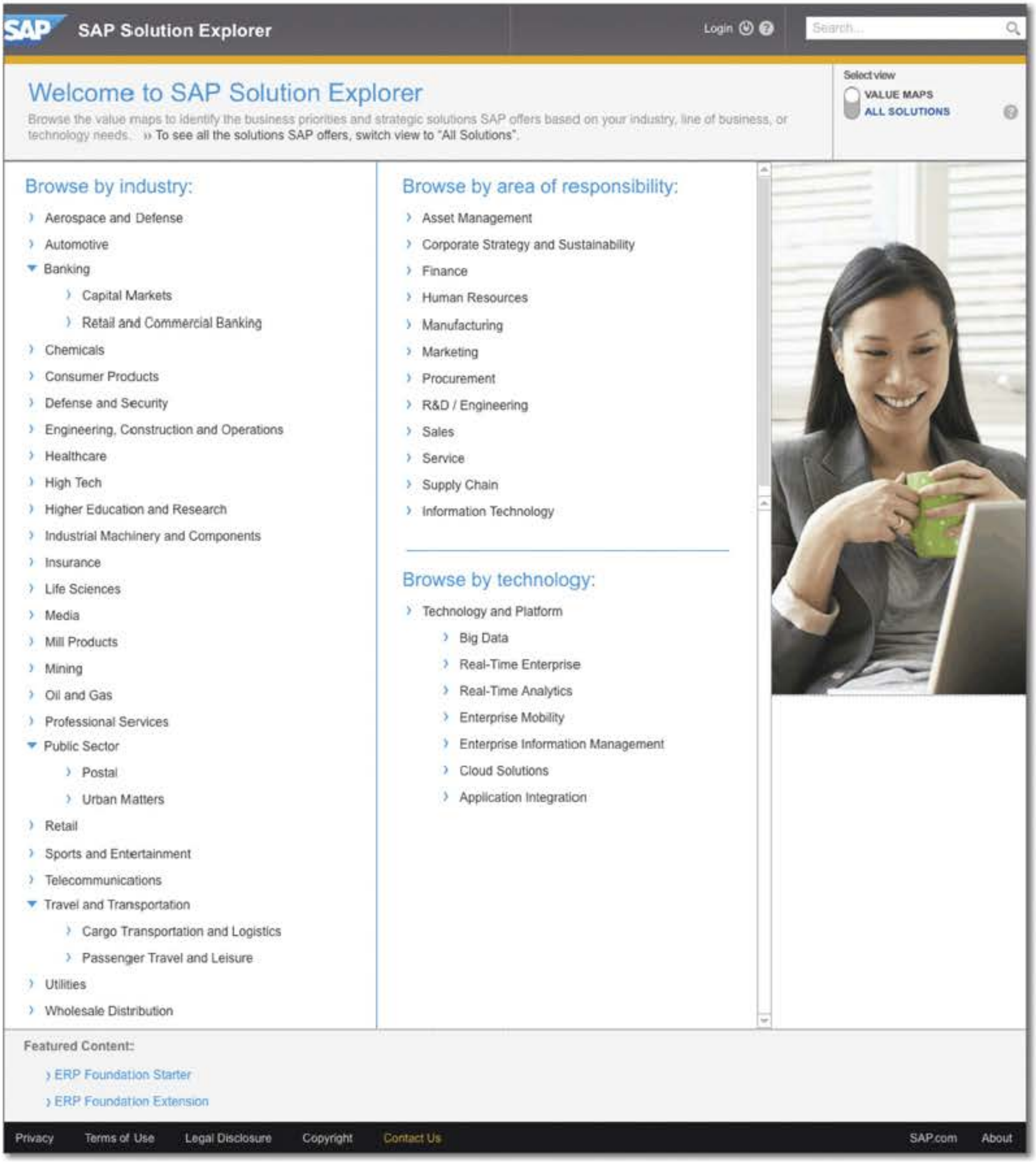
In this chapter, we will extend our generic description of potential benefits of big data solutions by focusing on the potential benefits of SAP HANA. This means that we are first going to ask ourselves what kind of *additional* benefits may result from the four items mentioned previously.

Next, we are going to supplement the two axes of our benefit–value driver matrix, how and where (see Figure 3.2), with a third dimension, industry. We will give you a feel for which potential benefits of SAP HANA are industry-specific or industry-neutral and which of the eight combinations of how and where shown in Figure 3.2 are to be found in which industry. At the same time, we are going to get away from the bird's-eye view, moving in a bit closer and zooming into the SAP world. Our lists do not attempt to be complete; their primary purpose is to help you develop a sense for potential benefits.

Value maps

We also want to take the opportunity to get a bit more SAP specific, which is why we are going to use the value maps within SAP Solution Explorer as a basis for grouping industries.

Figure 3.2 shows the home page of SAP Solution Explorer (*https://rapid.sap.com/se/executive#!/home*); at the top-right-hand side of this page you can choose between structuring your view by VALUE MAPS or by ALL SOLUTIONS. In this book, we will only use the view structured by value maps.



SAP Solution Explorer

Login

Search...

Select view

VALUE MAPS

ALL SOLUTIONS

Welcome to SAP Solution Explorer

Browse the value maps to identify the business priorities and strategic solutions SAP offers based on your industry, line of business, or technology needs. » To see all the solutions SAP offers, switch view to "All Solutions".

Browse by industry:

- Aerospace and Defense
- Automotive
- Banking
  - Capital Markets
  - Retail and Commercial Banking
- Chemicals
- Consumer Products
- Defense and Security
- Engineering, Construction and Operations
- Healthcare
- High Tech
- Higher Education and Research
- Industrial Machinery and Components
- Insurance
- Life Sciences
- Media
- Mill Products
- Mining
- Oil and Gas
- Professional Services
- Public Sector
  - Postal
  - Urban Matters
- Retail
- Sports and Entertainment
- Telecommunications
- Travel and Transportation
  - Cargo Transportation and Logistics
  - Passenger Travel and Leisure
- Utilities
- Wholesale Distribution

Browse by area of responsibility:

- Asset Management
- Corporate Strategy and Sustainability
- Finance
- Human Resources
- Manufacturing
- Marketing
- Procurement
- R&D / Engineering
- Sales
- Service
- Supply Chain
- Information Technology

Browse by technology:

- Technology and Platform
  - Big Data
  - Real-Time Enterprise
  - Real-Time Analytics
  - Enterprise Mobility
  - Enterprise Information Management
  - Cloud Solutions
  - Application Integration

Featured Content:

- ERP Foundation Starter
- ERP Foundation Extension

Privacy Terms of Use Legal Disclosure Copyright Contact Us SAP.com About

Figure 3.2 SAP Solution Explorer

Solution

[«]

Like SAP, we distinguish between *solutions* and *products*. A solution is meant to satisfy the requirements of a complete business case. It provides all the functionality needed for this and often embraces more than one product. In contrast, a product is a software package or a service that is sold individually.

The structure of a solution is therefore determined on the basis of business demands; a product's scope is the result of technical considerations or the vendor's licensing model. Imagine you needed a solution that detects fraudulent credit card transactions. If you were an SAP customer, your solution could consist of the following products:

- Parts of SAP Business Suite (for example, Payment Card Processing in SAP CRM)
- SAP HANA as a database/tool for further analysis (using implementation scenarios like the data mart, app, or SAP Business Suite on SAP HANA scenarios)
- SAP PAL (for example, `ANOMALYDETECTION`, one of PAL's clustering algorithms)
- Libraries of the statistical programming language R (containing other algorithms used for outlier detection)
- SAP BusinessObjects BI (to design reports and to generate alerts)

Depending on your business case, solutions might consist of a combination (a toolchain) of SAP products (such as SAP PAL) and non-SAP products (such as R). In the value maps of SAP Solution Explorer, SAP also defines another option that incorporates more than one solution: the so-called *end-to-end solution*. In addition, solutions and Rapid Deployment Solutions are listed separately; RDSs are based upon solutions but are often already preconfigured and—unlike solutions—are (more or less) ready to be used straight away.

Value Map

[«]

SAP has used value maps to provide customers with a business-centric, solution-oriented view of its products for quite some time. The meaning of the term *value map*, however, has changed and evolved over the last couple of years.

At the moment, value maps within SAP Solution Explorer are a structured representation of end-to-end solutions that are key to creating shareholder value in a certain industry and/or in the context of certain business processes. Value maps, therefore, provide you with a business-driven, process- and solution-oriented view of SAP's portfolio of end-to-end solutions, solutions, and products. In Section 3.2, we will discuss certain value maps more extensively.

In Section 3.3, we will use the value maps provided by SAP Solution Explorer to refine the where dimension of our benefit-value driver matrix.

Classifying the case studies

To conclude—and also as an introduction to the following case study chapters—we are going to locate our case studies within the grid (industry, business process group, etc.) defined by SAP's value maps; we will do this in Section 3.4.

3.1 Creating Shareholder Value with SAP HANA

To answer the question of whether SAP HANA can create greater benefits (and thereby shareholder value) or at least different ones, we are going to return to the four key differentiators mentioned in Chapter 2, Section 2.1.3:

- Preimplemented functions and algorithms
- Shifting application logic to the database layer (plus newly developed declarative or meta programming languages)
- Integration with SAP products (data and metadata); coalescence of OLTP and OLAP
- Integration with non-SAP products (data and metadata)

These four areas can again be grouped into two categories, depending on whether they primarily create quantitative (see Section 3.1.1) or qualitative (see Section 3.1.2) advantages.

3.1.1 Implementing Big Data Solutions Faster and Cheaper

Faster results

If you are considering SAP PAL and R as an integral part of SAP HANA, SAP's in-memory database comes with loads of preimplemented algorithms (see the section "Specific (Preimplemented) Functions" in Chapter 2, Section 2.1.3); furthermore—thanks to supplementary products from SAP—SAP HANA can also be easily integrated with many other products outside the SAP world (see the section "(Meta) Data Integration with Non-SAP products" in Chapter 2, Section 2.1.3). Both facts together lead to two consequences for big data projects:

- **Lower effort**

  The effort needed for implementing solutions is reduced dramatically, which also means that even business cases that would not make sense without preimplemented algorithms suddenly become commercially attractive. This advantage not only impacts implementation resources but also improves your position with almost each and every value driver related to an implementation project: training costs, opportunity costs, occupancy/infrastructure costs, and test or (with company-critical solutions) auditing costs.

  There is no simple answer to the question of whether such savings justify higher entry costs (compared to open-source solutions). In the end, it all depends on how intensively you are going to use such preimplemented functions instead of developing functions yourself.

  But with big data—as with most other things in IT—the proof is in the pudding. If you are able to quickly generate measurable benefits in small projects and with reasonable budgets, then you won't have to wait long for other departments to come up with new requirements.

  For this reason, SAP HANA has advantages compared to open-source products. With preimplemented algorithms, you'll get results a lot faster than if you had to implement all of it yourself.

- **Trial and error is now an option**

  If algorithms galore are available at the touch of a button (as they are with SAP PAL and R), the threshold for trying something will be a lot lower than if you had to kick off a project (and first apply for a budget with your boss's boss, going through the well-known thicket of political intrigue and resistance).

  With very little effort, your data scientists can test whether k-Means or self-organizing maps (SOMs, also called Kohonen maps) are better at analyzing the samples you have or (if k-Means is used) which starting values for *k* and which distance measures will lead to better results in your case.

  Your CFO might be even more pleased to learn that—because you have already migrated your SAP Business Suite to SAP HANA—you could also cash in on the additional benefit of seamless integration with operational data (because there isn't a lot to do in terms of data acquisition).

It is obvious that more attempts for the same cost will increase your chances of detecting patterns in your data. Bear in mind, however, that, as with basic research projects, you don't know in advance what you are looking for, so you can't tell what you are going to find or what benefits your findings might generate and via which value drivers these benefits would finally increase shareholder value. It therefore often remains difficult to exactly (or even broadly) quantify in advance benefits related to trial-and-error experimentation.

Deciding for or against SAP HANA based upon this aspect is to a certain extent a fundamental strategic decision rather than an operational micromanagement one driven by quick wins.

3.1.2 Real-Time Automation

Bringing operational and analytical systems together

For us, one of the most fascinating aspects of SAP HANA is that SAP's new in-memory database is blurring the boundaries between operational and analytical systems. Boundaries that (in the past) have been hard to overcome and that always led to interfaces being built or solutions for data logistics being bought and configured are now becoming more permeable. The option to build columnar as well as row-based structures blurs what has traditionally been the clear line between OLTP and OLAP (see the section "Other Products (Databases, Platforms, Technology, and Services)" in Chapter 2, Section 2.1.3). At the same time, moving procedures into the database also softens the separation between the application and database layers (see the section "Application Logic in the Database Layer" in Chapter 2, Section 2.1.3).

In the long run (and taking into account the tendency to automate decisions), applications might no longer be a jungle of complex, often unfathomable logic. Instead, they might become nothing but a combination of collectors of raw data and analytic/calculation views on these data. Distinctions such as those between OLTP and OLAP or between an application and its database are becoming meaningless in an SAP HANA environment.

This merger will have far-reaching consequences, going way beyond mere technical aspects:

- **Performance**
  If OLTP- and OLAP-related functionalities reside in one single solution, then using the results of complex, strategic analysis (OLAP) for driving operational processes and decisions (OLTP) in real time will become a lot easier and faster.
- **Flexibility**
  If conclusions from analytical applications (OLAP) not only necessitate changes to settings in customizing but also changes in terms of which parameters are available for customizing (OLTP), then such adaptions can be made faster and in a more consistent way.
- **Real-time scenarios**
  Performance and flexibility together are the foundation for the real-time automation scenarios discussed under "Products for Data Generation (Data Collection, Data Processing, and Triggering Actions)" and "Products for Data Exploitation" in Chapter 2, Section 2.1.3.

However, dissolving traditional separations and structures also leads to a couple of challenges for data administration.

Taking a cross-industry view

In the following sections, we are going to address the question of how SAP HANA might be used in a number of industries and with different business processes. In doing so, we first zoom out from looking at *specific* benefits of SAP HANA, once again considering *all* potential benefits of big data (now, however—unlike in Chapter 1—also including those benefits that are SAP HANA–specific as well). Or to put it another way, we are no longer trying to identify *additional* benefits of SAP HANA (as discussed so far in this chapter) but instead including all elements mentioned in Chapter 1 (that is, all eight benefit types shown in Chapter 1, Figure 1.2).

3.2 SAP HANA in Different Industries

SAP's industry classification

Industries can be grouped using all kinds of criteria. In our value driver database (which can be downloaded from *www.sap-press.com/3647*), we are, for example, using the Statistical Classification of Economic Activities in the European Community (*Nomenclature Statistique des Activités Économiques dans la Communauté Européenne* [*NACE*]). But because this book primarily addresses SAP customers and prospects, we are going to use the industry classification from SAP Solution Explorer. You can find this industry classification on the left-hand side of SAP Solution Explorer's home page (see Figure 3.2, under BROWSE BY INDUSTRY).

[+]

Scope of SAP Solution Explorer

The scope of SAP Solution Explorer is not limited to SAP HANA. The tool covers SAP's complete product portfolio (and therefore also SAP HANA), helping you find out which solutions and products could make sense in which industry or business process.

Information within the value maps

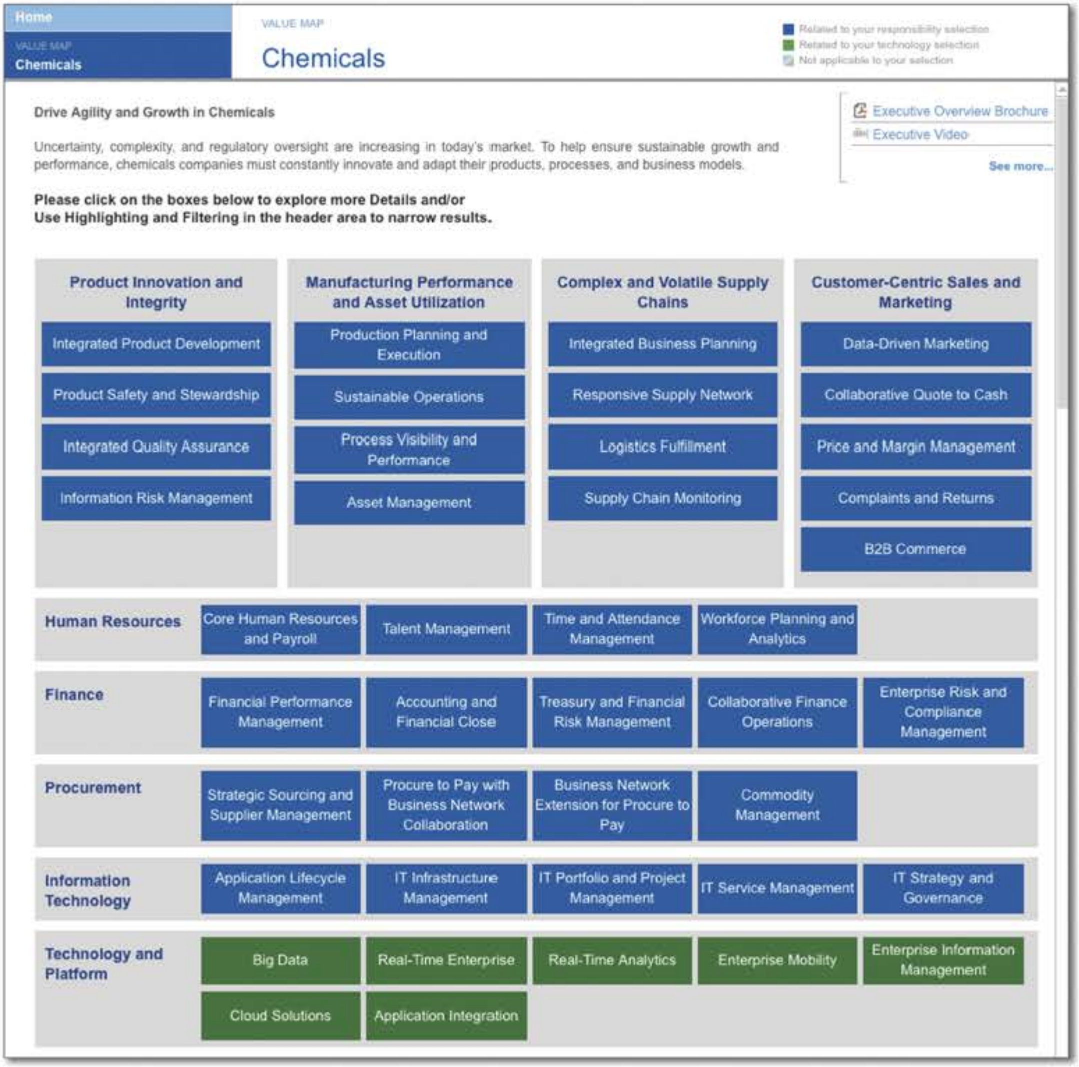
A couple of value maps contain a green rectangle marked BIG DATA under TECHNOLOGY AND PLATFORM (see Figure 3.3). Unlike the blue boxes that represent end-to-end solutions, the green boxes represent business priorities. Please be aware that information furnished under the business priority BIG DATA in various industry value maps is not industry-specific but related to big data (and SAP HANA) in general.

3.2.1 Working with SAP Solution Explorer

For each industry listed on SAP Solution Explorer's home page, you can view that industry's specific value map by clicking on the respective hyperlink. As an example, Figure 3.3 shows the value map for the chemical industry.

Composition of value maps

Value maps include blue and green rectangles that are assigned to either industry-specific business priorities (like PRODUCT INNOVATION AND INTEGRITY or MANUFACTURING PERFORMANCE AND ASSET UTILIZATION in Figure 3.3) or to cross-industry business priorities (like FINANCE or PROCUREMENT in Figure 3.3). Each blue rectangle represents an end-to-end solution that is (from SAP's perspective) particularly suitable for delivering benefits and shareholder value within that industry. Clicking on any of these end-to-end solutions in a value map will bring up further details (success stories, customer credentials, etc.), which also list solutions and RDSs belonging to that end-to-end solution in a column on the left-hand side.



Home

VALUE MAP
Chemicals

VALUE MAP
Chemicals

🟦 Related to your responsibility selection
🟩 Related to your technology selection
▨ Not applicable to your selection

Drive Agility and Growth in Chemicals

Uncertainty, complexity, and regulatory oversight are increasing in today's market. To help ensure sustainable growth and performance, chemicals companies must constantly innovate and adapt their products, processes, and business models.

Executive Overview Brochure
Executive Video
See more...

Please click on the boxes below to explore more Details and/or Use Highlighting and Filtering in the header area to narrow results.

Product Innovation and Integrity	Manufacturing Performance and Asset Utilization	Complex and Volatile Supply Chains	Customer-Centric Sales and Marketing
Integrated Product Development	Production Planning and Execution	Integrated Business Planning	Data-Driven Marketing
Product Safety and Stewardship	Sustainable Operations	Responsive Supply Network	Collaborative Quote to Cash
Integrated Quality Assurance	Process Visibility and Performance	Logistics Fulfillment	Price and Margin Management
Information Risk Management	Asset Management	Supply Chain Monitoring	Complaints and Returns
			B2B Commerce

Human Resources	Core Human Resources and Payroll	Talent Management	Time and Attendance Management	Workforce Planning and Analytics	
Finance	Financial Performance Management	Accounting and Financial Close	Treasury and Financial Risk Management	Collaborative Finance Operations	Enterprise Risk and Compliance Management
Procurement	Strategic Sourcing and Supplier Management	Procure to Pay with Business Network Collaboration	Business Network Extension for Procure to Pay	Commodity Management	
Information Technology	Application Lifecycle Management	IT Infrastructure Management	IT Portfolio and Project Management	IT Service Management	IT Strategy and Governance
Technology and Platform	Big Data	Real-Time Enterprise	Real-Time Analytics	Enterprise Mobility	Enterprise Information Management
	Cloud Solutions	Application Integration			

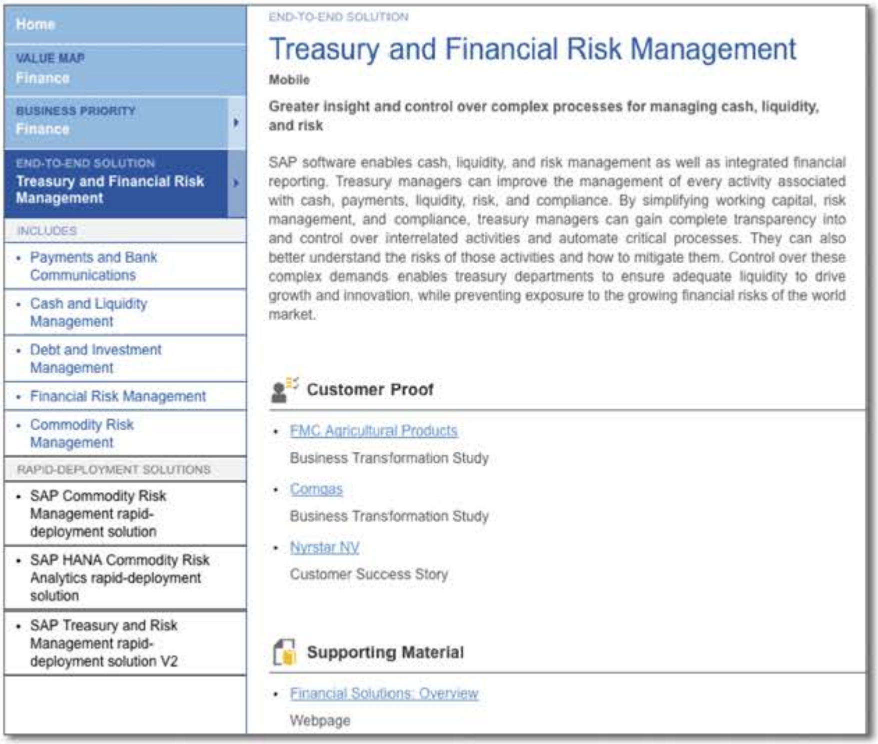
Figure 3.3 Chemicals Value Map

The green rectangles represent business priorities. Clicking on any of the green rectangles brings up a description of the business priority (and sometimes references) and also shows end-to-end solutions assigned to this business priority on the left-hand side.

Detailed view of end-to-end solution

Figure 3.4 shows the view you get when clicking on TREASURY AND FINANCIAL RISK MANAGEMENT. On the top-right-hand side, you will find a link to an information brochure (Solution in Detail Brochure), under Customer Proof there are a few customer success stories, and the link under Supporting Material will take you to the Financial Solutions section on SAP's website.

The navigation bar on the left-hand side lists a couple of related solutions; under Rapid Deployment Solutions you will also find some RDSs, all of which belong to the solution (called a *solution capability* by SAP, because it still takes some effort from your end to turn it into a solution) Commodity Risk Management.



Home

VALUE MAP
Finance

BUSINESS PRIORITY
Finance

END-TO-END SOLUTION
Treasury and Financial Risk Management

INCLUDES

- Payments and Bank Communications
- Cash and Liquidity Management
- Debt and Investment Management
- Financial Risk Management
- Commodity Risk Management

RAPID-DEPLOYMENT SOLUTIONS

- SAP Commodity Risk Management rapid-deployment solution
- SAP HANA Commodity Risk Analytics rapid-deployment solution
- SAP Treasury and Risk Management rapid-deployment solution V2

END-TO-END SOLUTION

Treasury and Financial Risk Management

Mobile

Greater insight and control over complex processes for managing cash, liquidity, and risk

SAP software enables cash, liquidity, and risk management as well as integrated financial reporting. Treasury managers can improve the management of every activity associated with cash, payments, liquidity, risk, and compliance. By simplifying working capital, risk management, and compliance, treasury managers can gain complete transparency into and control over interrelated activities and automate critical processes. They can also better understand the risks of those activities and how to mitigate them. Control over these complex demands enables treasury departments to ensure adequate liquidity to drive growth and innovation, while preventing exposure to the growing financial risks of the world market.

Customer Proof

- FMC Agricultural Products
  Business Transformation Study
- Comgas
  Business Transformation Study
- Nyrstar NV
  Customer Success Story

Supporting Material

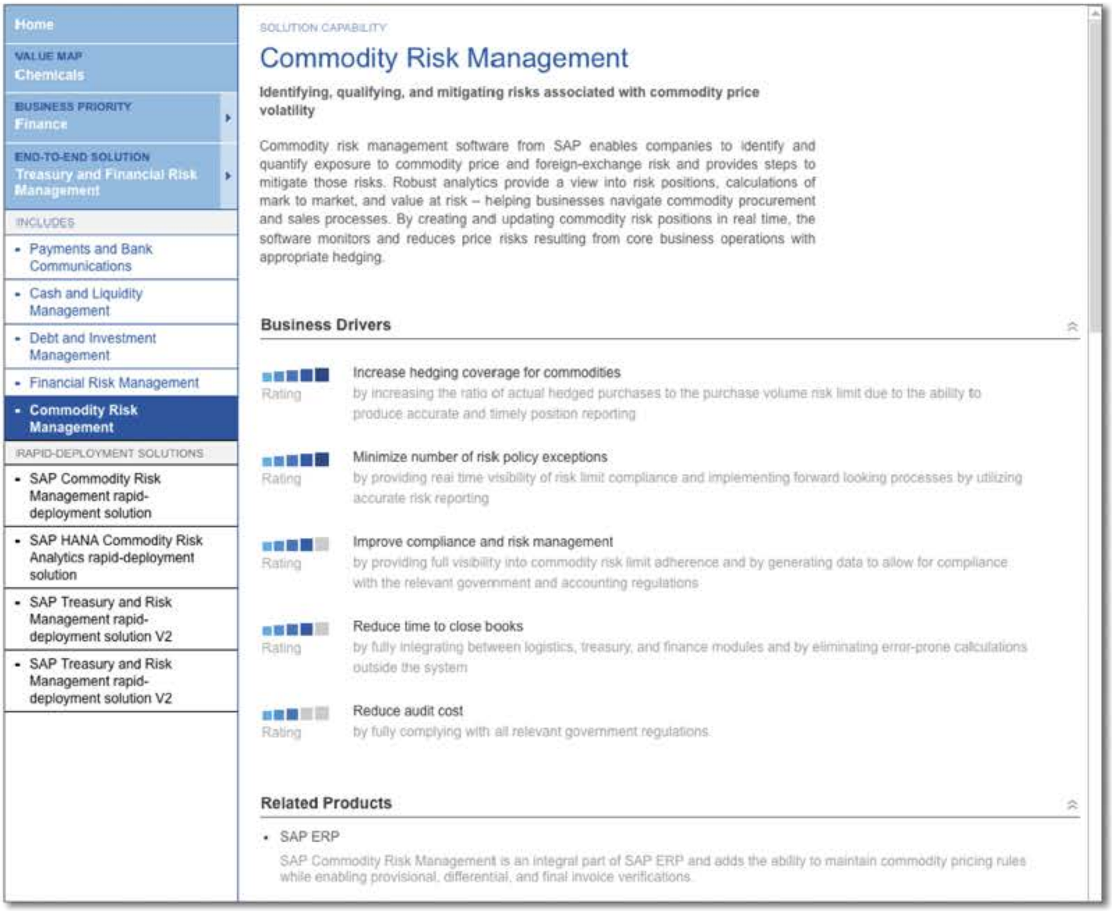
- Financial Solutions: Overview
  Webpage

Figure 3.4 Treasury and Financial Risk Management End-to-End Solution

Solutions (solution capabilities)

If you select any of the solutions (solution capabilities) in the navigation bar, you will find which value drivers (called *business driver* here) could be relevant in their context and which products from SAP would be part of that particular solution.

Figure 3.5 shows a portion of the detailed view for the Commodity Risk Management solution, which belongs to the Treasury and Financial Risk Management end-to-end solution within the Chemicals value map. Figure 3.6 shows a part of the detailed view for the SAP Commodity Risk Management rapid-deployment solution RDS.


Home
VALUE MAP
Chemicals
BUSINESS PRIORITY
Finance
END-TO-END SOLUTION
Treasury and Financial Risk Management
INCLUDES
Payments and Bank Communications
Cash and Liquidity Management
Debt and Investment Management
Financial Risk Management
Commodity Risk Management
RAPID-DEPLOYMENT SOLUTIONS
SAP Commodity Risk Management rapid-deployment solution
SAP HANA Commodity Risk Analytics rapid-deployment solution
SAP Treasury and Risk Management rapid-deployment solution V2
SAP Treasury and Risk Management rapid-deployment solution V2

SOLUTION CAPABILITY

Commodity Risk Management

Identifying, qualifying, and mitigating risks associated with commodity price volatility

Commodity risk management software from SAP enables companies to identify and quantify exposure to commodity price and foreign-exchange risk and provides steps to mitigate those risks. Robust analytics provide a view into risk positions, calculations of mark to market, and value at risk – helping businesses navigate commodity procurement and sales processes. By creating and updating commodity risk positions in real time, the software monitors and reduces price risks resulting from core business operations with appropriate hedging.

Business Drivers

Rating — Increase hedging coverage for commodities
by increasing the ratio of actual hedged purchases to the purchase volume risk limit due to the ability to produce accurate and timely position reporting

Rating — Minimize number of risk policy exceptions
by providing real time visibility of risk limit compliance and implementing forward looking processes by utilizing accurate risk reporting

Rating — Improve compliance and risk management
by providing full visibility into commodity risk limit adherence and by generating data to allow for compliance with the relevant government and accounting regulations

Rating — Reduce time to close books
by fully integrating between logistics, treasury, and finance modules and by eliminating error-prone calculations outside the system

Rating — Reduce audit cost
by fully complying with all relevant government regulations

Related Products

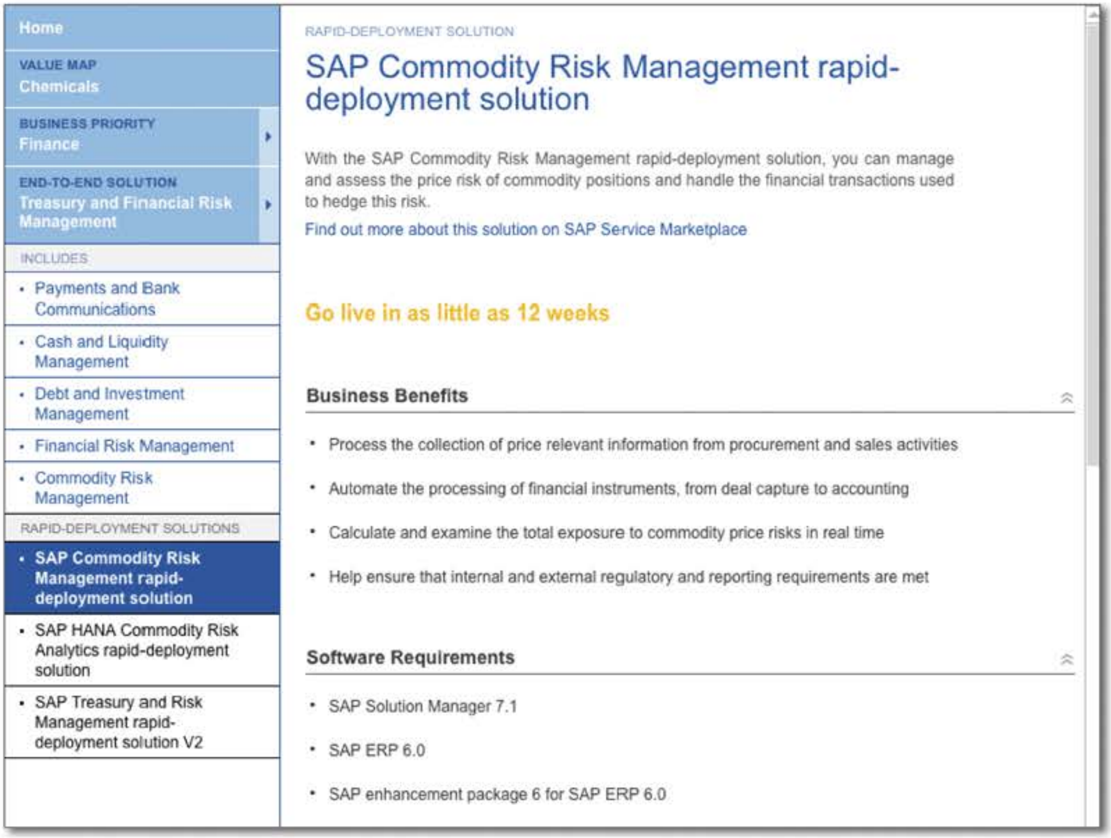
- SAP ERP

  SAP Commodity Risk Management is an integral part of SAP ERP and adds the ability to maintain commodity pricing rules while enabling provisional, differential, and final invoice verifications.

Figure 3.5 Commodity Risk Management Solution

Value(/business) drivers, required solutions, available RDSs

In Figure 3.5, you can see that SAP lists Increase hedging coverage for commodities as an important value driver for Commodity Risk Management, and (under Related Products) you will see that you need the product SAP ERP to implement the Commodity Risk Management solution. In addition, Figure 3.6 advises you that should you also want to use the SAP Commodity Risk Management rapid-deployment solution, you will need SAP Solution Manager 7.1 on top of SAP ERP 6.0.


Home
VALUE MAP
Chemicals
BUSINESS PRIORITY
Finance
END-TO-END SOLUTION
Treasury and Financial Risk Management
INCLUDES
• Payments and Bank Communications
• Cash and Liquidity Management
• Debt and Investment Management
• Financial Risk Management
• Commodity Risk Management
RAPID-DEPLOYMENT SOLUTIONS
• SAP Commodity Risk Management rapid-deployment solution
• SAP HANA Commodity Risk Analytics rapid-deployment solution
• SAP Treasury and Risk Management rapid-deployment solution V2

RAPID-DEPLOYMENT SOLUTION

SAP Commodity Risk Management rapid-deployment solution

With the SAP Commodity Risk Management rapid-deployment solution, you can manage and assess the price risk of commodity positions and handle the financial transactions used to hedge this risk.

Find out more about this solution on SAP Service Marketplace

Go live in as little as 12 weeks

Business Benefits

- Process the collection of price relevant information from procurement and sales activities
- Automate the processing of financial instruments, from deal capture to accounting
- Calculate and examine the total exposure to commodity price risks in real time
- Help ensure that internal and external regulatory and reporting requirements are met

Software Requirements

- SAP Solution Manager 7.1
- SAP ERP 6.0
- SAP enhancement package 6 for SAP ERP 6.0

Figure 3.6 SAP Commodity Risk Management RDS

Having invested only 10 minutes or so of industry-specific research, you already have a couple of ideas for business cases (one per solution capability), a list of applicable value drivers (which should, however, still be cross-checked and expanded a bit), and an idea of the products for which you may want to purchase licenses. As described in Chapter 2, the list of products will also help you select the right implementation scenario. Not so bad considering the very limited amount of time and mouse clicks you had to invest, eh?

3.2.2 Industry-Specific Potential Benefits

What to use SAP Solution Explorer for

Although SAP Solution Explorer sometimes shows only generic big data solutions for business priorities, the tool can still be used for two purposes:

- In addition to our benefit–value driver matrix (or morphological analysis), SAP Solution Explorer can also help you when hunting for ideas and opportunities.
- As SAP Solution Explorer tells you more about required products (for data generation, data exploitation, and so on), it will also help you identify a suitable implementation scenario.

We are going to once again pick up the example from the last section to develop some more detailed step-by-step instructions for generating ideas and selecting scenarios.

Generating Ideas Using SAP Solution Explorer

Scenarios within the value maps

Let's assume your company is operating within the chemical industry. You open SAP Solution Explorer. You then select CHEMICALS for the chemical industry (see Figure 3.3) and (under TREASURY AND FINANCIAL RISK MANAGEMENT) SAP COMMODITY RISK MANAGEMENT (see Figure 3.5). At the very bottom, you click on the (higher-level) COMMODITY MANAGEMENT, which brings up a list of demos, one of which is SPEED DEMO OF SAP COMMODITY RISK MANAGEMENT FOR COMMODITY CONSUMERS OF ALUMINIUM AND WHEAT. This *speed demo* (an SAP term) features two business transactions:

- A company is buying aluminum at a variable (market) price (starting with entering a purchase order). If goods are not bought at a fixed price but on the basis of the selling price on the day on which they are going to be delivered, there is the risk of price change for both the seller and the buyer.
- A company processing wheat is hedging the price-change risk of wheat to be purchased during the first half-year of 2013 via a future contract, with one of its house banks as the counterparty. Unlike the first transaction in this demo, the price-change risk does not result from a purchase order that has already been entered into the system

but from forecasted material requirements. These material requirements have not yet been entered into the company's SAP ERP but are uploaded from a Microsoft Excel file as aggregated monthly figures.

Examples for required solutions

In terms of products, both transactions in the preceding demo are using functions within SAP Treasury and Risk Management (TRM), a component belonging to Financial Supply Chain Management (FIN-FSCM) in SAP ERP.

Translating the preceding example to your own specific situation

Even if your company buys neither aluminum nor wheat, by examining these examples you may well think of similar issues in your organization. You may know that the purchase price of many of the materials you are buying depends on the price of crude oil. Hence, you could, for example, hedge the resulting exposure by buying call options on crude oil. Or your production might be pretty energy intensive, which is why your corporate treasury department has been negotiating futures at the European Energy Exchange (EEX) at Leipzig to protect you from price fluctuations with electrical energy. Unfortunately, such deals are still administered and evaluated in an Excel worksheet.

To process such transactions within SAP ERP in the future, you will need access to certain products (either by buying a license or by renting the respective service in a cloud environment). In the navigation bar on the left-hand side of SAP Solution Explorer, you may have spotted an RDS for commodity risk management; this RDS has SAP ERP 6.0 EHP 6 and SAP Solution Manager 7.1 as prerequisites. If you implemented this RDS you could—according so SAP—go live with commodity risk management in SAP in about 12 weeks.

Links to further information

A link to the SAP Service Marketplace at the very top of the RDS's description (FIND OUT MORE ABOUT THIS SOLUTION ON SAP SERVICE MARKETPLACE) takes you to a presentation (direct link: *service.sap.com/rds-commodity-risk*; once there, select SOLUTION DISCOVERY • SAP COMMODITY RISK MANAGEMENT SOLUTION DETAILS). Via this presentation, you could make yourself familiar with which business processes your system would be able to support in the future (for example, trading commodity swaps, commodity futures, commodity options, etc.). These business processes may have been on your agenda for quite some time. Coordinating with your CFO, you may have already developed detailed process models in Microsoft Visio that you can now once again dig out (perhaps even exhume) and review.

Analyzing the potential of SAP HANA

If your company already migrated its SAP ERP to an SAP HANA database a few months ago, you might be asking yourself whether you might be able to approach these business processes in an entirely different way, designing a state-of-the-art big data solution. Furthermore, you also may be a bit skeptical about the data flow shown in the demo with SAP Solution Explorer; to you, manually uploading commitments from an Excel spreadsheet doesn't really seem like a cutting-edge way of doing things.

Showtime for the benefit–value driver matrix

At this point, you will (hopefully) remember this book and pull out the benefit–value driver matrix (Chapter 1, Figure 1.2). This matrix will help you discover new potential benefits in existing business processes (regardless of whether you are talking about processes already implemented at your organization—the ones in your Visio flowcharts—or processes in new pieces of standard software like SAP TRM). If you remember the discussion in Chapter 1, Section 1.5, then you know that the matrix might even help you come up with new business processes not yet on your radar. In both cases (existing and new processes), SAP HANA could open the door to new insights, better decisions, sophisticated tools, and acting faster.

Examples for value drivers

In terms of business/use cases and benefits or value drivers, your train of thought could go as follows:

- **Knowing your exposure**

  To hedge this price-change risk (and to also avoid costly and potentially dangerous overhedging), you need reliable, up-to-date information about your commitments and your hedging transactions. One option to collect such data might be to use SAP HANA with the data mart scenario (see Chapter 2, Figure 2.15).

  A sample value driver could be "gains/losses from price changes" or "gains/losses from hedging transactions." In our benefit–value driver matrix, these value drivers would sit in the two upper-left quadrants (EXISTING PROCESSES/NEW INSIGHTS and EXISTING PROCESSES/BETTER DECISIONS).

- **Predicting your commitments**

  Knowing past or present commitments/exposures is only half the battle. The earlier you have reliable data about *future* commitments, the better (and the cheaper) hedging the resulting exposures will be. SAP PAL (used as a product for data exploitation in the data mart scenario; see Chapter 2, Figure 2.15) might be the right instrument to improve the forecasting quality when predicting commitments (such as predicting the amount of wheat to be bought in our example).

Using SAP PAL in this way would act on the same value drivers (gains/losses from price changes or gains/losses from hedging transactions). As SAP PAL also provides you with a couple of sophisticated algorithms, you will also start to move into the EXISTING PROCESSES/ SOPHISTICATED TOOLS area of the benefit–value driver matrix.

- **Early warning system**

  Breaking news can steamroll even the best forecasts and decisions. Hence, an early warning system telling you that—due to an increased volatility of commodity prices or exchange rates, for example—your hedging strategies will no longer work as planned might come in handy.

  This once again affects the same value drivers (gains/losses from price changes or gains/losses from hedging transactions), but as we are now talking about an application that will warn you in real time, the respective architectural model would be the app scenario. In terms of the benefit–value driver matrix, we are now in the lower two quadrants on the left-hand side (EXISTING PROCESSES/SOPHISTICATED TOOLS and EXISTING PROCESSES/ACTING FASTER).

- **Evaluating risks with a combination of hedging methods**

  With most exposures, there is more than one way of hedging any resulting financial risk. With the example used in the demo (purchasing aluminum), a commodity swap is not your only option; you could instead buy a call option on aluminum or enter a future contract. Having a human trader evaluate all theoretically possible alternatives plus expecting this poor guy to analyze the resulting risk profiles would be asking too much of him.

  Depending on whether you can, or would like to, use existing functionalities in SAP TRM or not, you could go for a solution based upon the SAP Business Suite on SAP HANA scenario (integrating with SAP

TRM as your product for data generation/exploitation) or the new SAP HANA apps scenario (building something entirely new).

In both cases, however, we would still be talking about the same value drivers (gains/losses from price changes or gains/losses from hedging transactions), but because finding the best combination of hedging methods automatically would probably be a new business process we are now in one of the following three quadrants on the right-hand side of the benefit–value driver matrix:

- New Processes/Better Decisions
- New Processes/Sophisticated Tools
- New Processes/Acting Faster

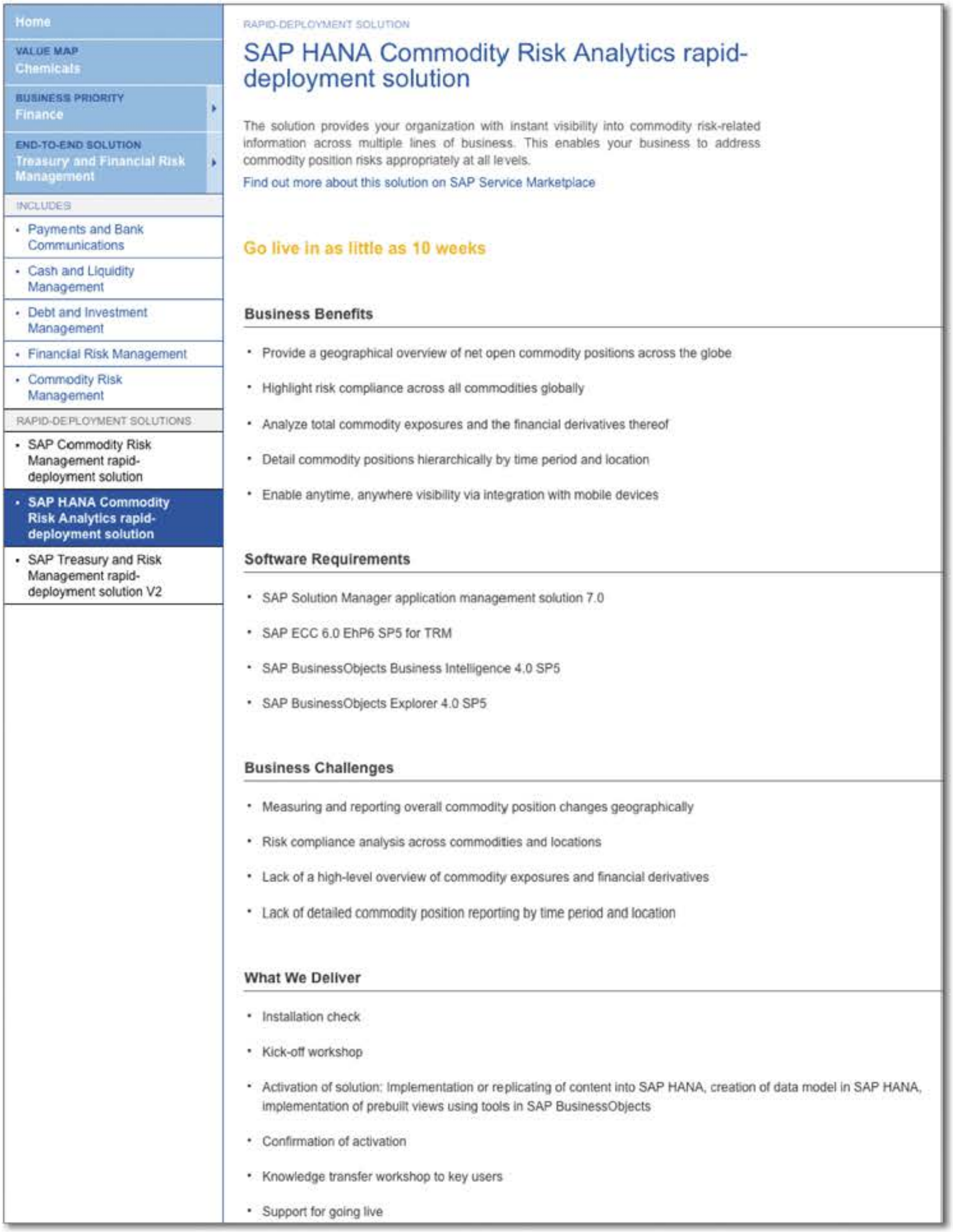
Your key benefits could be any one of these three:

- Better decisions (by factoring in more options)
- Sophisticated tools (by developing your own selection algorithms or by having a cutting-edge one developed via crowdsourcing)
- Acting faster (by continuously checking your options in real time on the basis of up-to-date prices, quotes, and volatilities)

Depending on your individual needs, one or two of the preceding benefits could be more important for your company than the other one(s). The value drivers would, however, remain unchanged.

SAP HANA-specific RDS

By carefully reading the information on SAP Solution Explorer, you may even have noticed that SAP offers an SAP HANA–specific RDS for commodity risk management (SAP HANA Commodity Risk Analytics rapid-deployment solution; see Figure 3.7). For this RDS, there is once again supplementary information on SAP Service Marketplace (direct link: *https://service.sap.com/rds-cra*; click on Solution Discovery and select any of the presentations there). The core purpose of the RDS is to present exposures in a comprehensive, prompt, and clear manner. It provides you with a database (plus some reports) for efficient commodity risk management. This fits in nicely with the preceding thoughts about the benefit–value driver matrix. The RDS might therefore take you a bit closer to the visions described there and save you substantial implementation costs.


Home
VALUE MAP
Chemicals
BUSINESS PRIORITY
Finance
END-TO-END SOLUTION
Treasury and Financial Risk Management
INCLUDES
Payments and Bank Communications
Cash and Liquidity Management
Debt and Investment Management
Financial Risk Management
Commodity Risk Management
RAPID-DEPLOYMENT SOLUTIONS
SAP Commodity Risk Management rapid-deployment solution
SAP HANA Commodity Risk Analytics rapid-deployment solution
SAP Treasury and Risk Management rapid-deployment solution V2

RAPID-DEPLOYMENT SOLUTION

SAP HANA Commodity Risk Analytics rapid-deployment solution

The solution provides your organization with instant visibility into commodity risk-related information across multiple lines of business. This enables your business to address commodity position risks appropriately at all levels.

Find out more about this solution on SAP Service Marketplace

Go live in as little as 10 weeks

Business Benefits

- Provide a geographical overview of net open commodity positions across the globe
- Highlight risk compliance across all commodities globally
- Analyze total commodity exposures and the financial derivatives thereof
- Detail commodity positions hierarchically by time period and location
- Enable anytime, anywhere visibility via integration with mobile devices

Software Requirements

- SAP Solution Manager application management solution 7.0
- SAP ECC 6.0 EhP6 SP5 for TRM
- SAP BusinessObjects Business Intelligence 4.0 SP5
- SAP BusinessObjects Explorer 4.0 SP5

Business Challenges

- Measuring and reporting overall commodity position changes geographically
- Risk compliance analysis across commodities and locations
- Lack of a high-level overview of commodity exposures and financial derivatives
- Lack of detailed commodity position reporting by time period and location

What We Deliver

- Installation check
- Kick-off workshop
- Activation of solution: Implementation or replicating of content into SAP HANA, creation of data model in SAP HANA, implementation of prebuilt views using tools in SAP BusinessObjects
- Confirmation of activation
- Knowledge transfer workshop to key users
- Support for going live

Figure 3.7 SAP HANA Commodity Risk Analytics RDS

SAP Solution Explorer does not always consider SAP HANA

With this example, SAP Solution Explorer even builds the bridge to SAP HANA, but this is not always the case. It will often make sense for you to develop your own applications or solutions. SAP Solution Explorer will only show you what SAP has thought of, and your competitors are able to use the Internet and buy RDSs as well.

We therefore recommend using SAP Solution Explorer to identify business processes that are a) key within your industry and b) already supported by classic SAP solutions, rather than trying to find SAP HANA solutions right from the start. Much like in our example, you can then use the methods presented in our discussion of morphological analysis (see Chapter 1, Section 1.5) to scan these processes for potential big data/ SAP HANA benefits. If you should—as in our case—happen to come across an SAP HANA-based solution or an SAP HANA-based RDS in the course of that process, then that's even better!

RDSs and underlying solution

With RDSs, there are often no proposed value drivers. RDSs belong to solutions (in our case, the RDSs shown in Figure 3.6 and Figure 3.7 belong to the solution in Figure 3.5); therefore, the value drivers of that solution can be applied as a starting point. Compared to the underlying solutions, RDSs often come with additional or more specific product requirements than the solution (in our case: SAP ECC 6.0 EHP 6 SP5 for TRM, SAP BusinessObjects BI 4.0 SP5, and SAP BusinessObjects Explorer 4.0 SP5).

[+]

Price-Change Risks in Other Industries

Quite a few industries encounter problems that are similar to the ones presented our example. If your company processes agricultural produce, then you may face the issue that not only the price but also the quality of your raw materials are fluctuating. This raises additional questions. As a brand manufacturer of cigarettes, champagne, or whisky, you want to make sure that your products taste more or less the same every year, regardless of the crop yield in certain growing areas, which means that predicting and even managing harvests might be a fascinating option to consider.

Selecting Implementation Scenarios Using SAP Solution Explorer

SAP Solution Explorer lists products

With all the examples mentioned so far (the ones from SAP's RDSs as well as the ideas we developed using the benefit–value driver matrix), as we went along we also came up with provisional ideas about appropriate implementation scenarios. With solutions from the SAP Solution Explorer, the required products (for data generation, exploitation, and so on) are often listed explicitly, helping you narrow down potential scenarios.

Products help select implementation scenarios

With business cases and solutions that you have developed, the implementation scenario can often be determined by asking yourself what your big data–specific requirements are (for example, whether you need a real-time solution or not). After answering that question, you can do the following:

- Scan the market for products for data generation/management/storage/exploitation and so on
- Check your list of products against SAP's product portfolio or against anything else you'll find outside of the SAP world

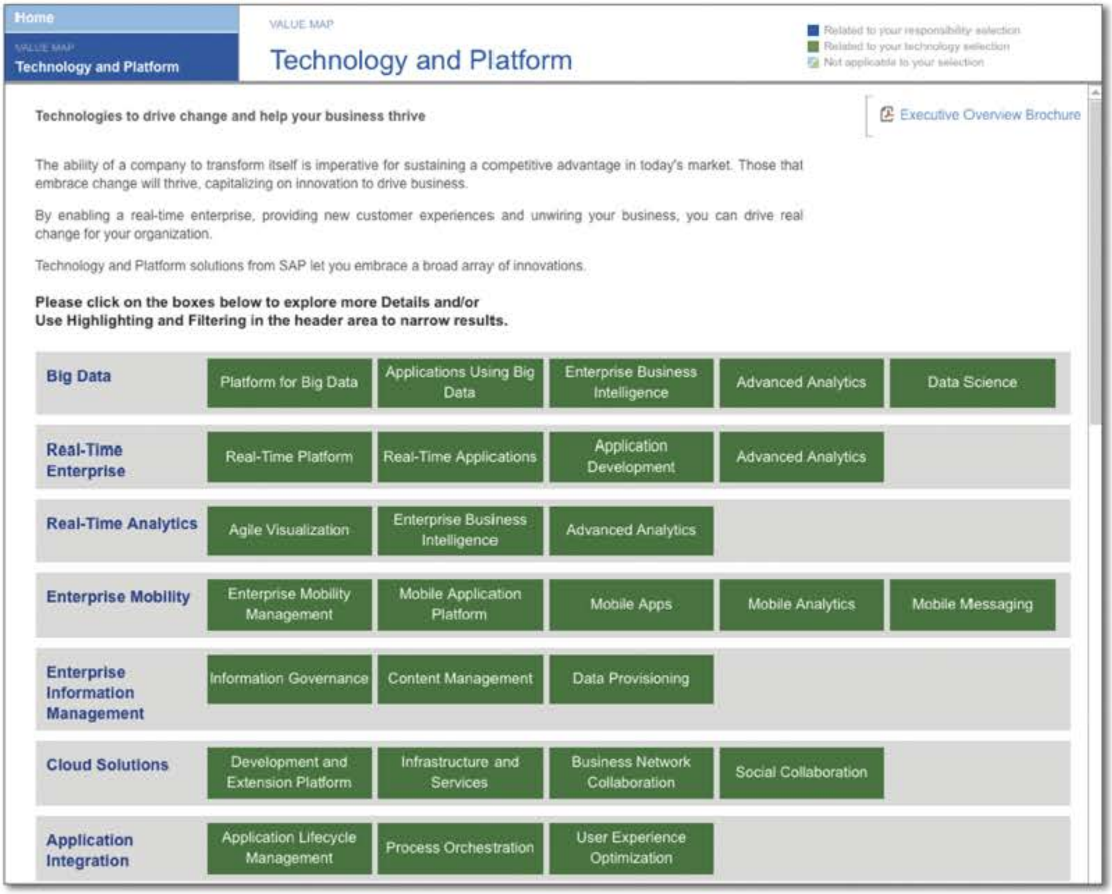
After completing these two steps, you'll have a tentative list of the SAP and non-SAP products you need, which will once again lead you toward a specific implementation scenario. Further considerations in terms of selecting the right implementation scenario can be found in Chapter 2, Section 2.2.

3.2.3 Cross-Industry Potential Benefits

In addition to industry-specific value maps, SAP Solution Explorer also contains quite a few cross-industry ones. These value maps are structured by functional areas, areas of responsibility (Browse by area of responsibility), or technologies (Browse by technology) and can be reached via the respective links on the right-hand side of SAP Solution Explorer's home page (see Figure 3.2).

Technology and Platform value map

Although there are separate value maps for each area of responsibility, all hyperlinks under Browse by technology are going to take you to one and the same value map (see Figure 3.8).



Technology and Platform

🟦 Related to your responsibility selection
🟩 Related to your technology selection
▨ Not applicable to your selection

Technologies to drive change and help your business thrive

The ability of a company to transform itself is imperative for sustaining a competitive advantage in today's market. Those that embrace change will thrive, capitalizing on innovation to drive business.

By enabling a real-time enterprise, providing new customer experiences and unwiring your business, you can drive real change for your organization.

Technology and Platform solutions from SAP let you embrace a broad array of innovations.

Please click on the boxes below to explore more Details and/or
Use Highlighting and Filtering in the header area to narrow results.

Big Data	Platform for Big Data	Applications Using Big Data	Enterprise Business Intelligence	Advanced Analytics	Data Science
Real-Time Enterprise	Real-Time Platform	Real-Time Applications	Application Development	Advanced Analytics	
Real-Time Analytics	Agile Visualization	Enterprise Business Intelligence	Advanced Analytics		
Enterprise Mobility	Enterprise Mobility Management	Mobile Application Platform	Mobile Apps	Mobile Analytics	Mobile Messaging
Enterprise Information Management	Information Governance	Content Management	Data Provisioning		
Cloud Solutions	Development and Extension Platform	Infrastructure and Services	Business Network Collaboration	Social Collaboration	
Application Integration	Application Lifecycle Management	Process Orchestration	User Experience Optimization		

Figure 3.8 Technology and Platform Value Map

Demand Signal Management RDS

One example of a cross-industry solution is SAP's RDS for demand-and-supply network planning, found under Browse by technology • Technology and Platform • Real-Time Enterprise • Real-Time Applications • SAP Demand and Supply Network Planning rapid-deployment solution.

Predicting demand and resulting material requirements is relevant for practically each and every manufacturing business, which is why the RDS is listed there. Even service organizations (such as hosting providers) want to know more about future demand, and although they are not shipping physical products, demand still drives their requirements in terms of, for example, server capacity.

SAP HANA as an option

Once you start looking at data from millions or billions of transactions with thousand or millions of customers, demand-and-supply network planning also becomes an interesting area for big data. This is why SAP's respective RDSs can also include SAP HANA Live for SAP SCM (SAP HANA CONTENT HBA SAP SCM 100 SP005) and SAP HANA Appliance Software SPS07 (SAP HANA Server, SAP HANA Client, and SAP HANA Studio). For this particular RDS, there is a detailed list of required products.

3.3 SAP HANA in (SAP's) Business Processes

Level of detail in SAP Solution Explorer

As we approach the end of this chapter, we will look at the where dimension in our benefit–value driver matrix, showing with which SAP business processes SAP HANA might help you generate additional shareholder value. We are not going to drill down into individual processes here. The SAP Solution Explorer (as a tool freely available on the Internet) shows end-to-end solutions and solutions based upon a variety of SAP products. It does not, however, present any SAP- or customer-specific business processes (in the form of, for example, process models).

Modeling business processes

Business processes can be modeled in a variety of solutions that are either ERP/SAP specific or totally independent from the kind of product you want to use for implementing them; some examples are: Microsoft's Visio, Software AG's ARIS, Oracle's Business Process Management Suite, SAP Solution Manager, or SAP Business Process Management (BPM).

SAP Process Orchestration

SAP offers its modeling tool SAP BPM as part of a package called SAP Process Orchestration (PO). This package also contains SAP Business Rules Management (BRM) and SAP Process Integration (P). All three of these products are seamlessly integrated with SAP Solution Manager and its reference models.

Books about business process modeling with SAP

Within the scope of this book, we can't discuss the functionalities of these products in detail (we've probably used up our quota of trees as it is). If you are interested in business-process modeling with SAP, you will find more information in the following four SAP PRESS books:

- *SAP Process Orchestration* (Bilay and Viana, 2015)
- *Practical Workflow in SAP* (Dart, Keohan, Rickayzen, et al., 3rd edition, 2014)

- *Applying Real-World BPM in an SAP Environment* (Chase, Rosenberg, Rukhshaan, Taylor, and von Rosing, 2011)
- *SAP Solution Manager* (Melich and Schäfer, 3rd edition, 2012)

We'll just add one remark in this context: to an increasing degree, SAP is orienting itself on the standards defined in SAP Enterprise Architecture Framework (EAF) and therefore on a service-oriented way of representing processes.

[«]

SAP Enterprise Architecture Framework (EAF)

SAP Enterprise Architecture Framework is a collection of methods, procedures, tools, and templates based upon the Open Group Architecture Framework (TOGAF), applied to developing (enterprise) architectures.

EAF extends TOGAF's toolbox by allowing for standard software and SAP-specific building blocks and tools. For example, TOGAF doesn't contain any reference models (of business processes), whereas SAP EAF comes with such reference models, reflecting SAP's approach to handling business processes.

[«]

Service-Oriented Architecture (SOA)

A *service-oriented architecture* is an architectural pattern that is aligned with business processes and often used when designing distributed systems. Service-oriented architectures consist of services that serve other services. Services are loosely coupled, self-contained units of functionalities.

Looked upon from the outside, a service is something like a black box that bundles related functions and makes them available via clearly defined interfaces. Key characteristics of a service consist of the following:

- Is clearly defined
- Offers business functions
- Is functionally self-sufficient

One example for such a service would be a function to request the credit rating of a prospective customer from a credit-rating agency. Via a precisely defined interface, the service is called (for example, from an ERP application), providing it with the data needed to identify the customer (with the credit-rating agency). It then responds to the application with a credit-worthiness level (that is again clearly defined as, for example, "A," "B," or "C") for this customer.

A service made available within a network is often called a *web service*. With big data, web services are becoming more and more important. Instead of evaluating the influence of a follower on Twitter yourself, you could just request this information from *https://www.klout.com* via a clearly defined programming interface.

How do you proceed when trying to identify non-industry-specific fields of application for SAP HANA by scanning business processes instead of industries? Follow these steps:

1. Define your scope—that is the business process(es) you are considering. Quite often, you will have to take into account political aspects. Improvements—no matter how much benefit they might create—are not always enforceable, and within your company some business processes are usually in focus while others (though more important from your perspective) are not.
2. From these business processes, select the one(s) with the best value proposition. Which business processes are the most important ones for shareholder or (in Six Sigma, for example, a set of tools for customer-focused process improvements) customer value? Improving which business process will create maximum value with minimum effort? Remember that if SAP HANA is new in your organization you will need to show results as soon as possible.

   Of course, to make such a selection, you need to know which business process is influencing shareholder value via which value drivers. This is, however, not a big data–related question but a general strategic task for your organization; if you can't answer that question, making value-driven decisions in general is by definition impossible.
3. Have a closer look at the selected business process(es) via your preferred modeling tool(s). Focus on those steps within the process at which decisions are made (by humans or by systems). Identify associated value drivers (to get some inspiration, use the value driver database we have made available at *www.sap-press.com/3647*) and proceed as described in Chapter 1, Section 1.5.

Decisions are often made based on forecasts and models, and big data solutions are quite good at improving both forecasts and models. This is why the potential benefits listed in the rows of our benefit–value driver matrix (Chapter 1, Figure 1.2) often occur when decisions have to be made, such as in the following situations:

- **New insights**
  New insights about your business processes often lead to a revised version of that process. Maybe you can do away with some decisions (because new insights tell you they don't add value), maybe more potential outcomes of decisions have to be taken into account, or maybe a decision should be split into a series of individual decision steps.

  One example could be a credit check when relatively small amounts are involved and therefore little risk. If you find out (via an appropriate analysis) that—under a certain amount threshold—potential damages are lower than the costs of the credit check, then you may decide to do without the check at all, which in the end means making two decisions instead of one: first checking the amount (to decide whether a credit check is necessary) and then (if yes) checking the credit worthiness of your creditor.
- **Better decisions**
  You may want to use a big data solution to automate decisions that were until now made by humans or (alternatively) supply the decision maker with better (more exact or up-to-date) information. You could, for example, monitor tweets to learn at an early stage whether shortages are to be expected with raw materials that are essential for your business. Such shortages could, for example, be caused by bad weather in one of the producing countries; in this case, the information is probably on Twitter months before the supply shortfalls hit your production targets.
- **Sophisticated tools**
  If decisions are already automated (following clearly defined, exact rules), then better and more sophisticated algorithms could help you make better ones. Many systems used to prevent credit card fraud are using proven and tested rules to detect and report cases of suspected fraud.

  But human behavior changes constantly, and criminals are often particularly good at adapting to changing environments. You may therefore want to take two countermeasures:

  - Constantly monitor the performance of your rules (via certain key performance indicators)
  - Keep updating your system by continuously analyzing fraudulent and innocent transaction patterns, employing self-learning algorithms instead of static rules

- **Acting faster**

  Whenever humans are making decisions, you may lose valuable time; humans need to sleep and are not infallibly vigilant. Therefore, there may be no point in asking yourself whether to implement an early warning system to enable users to act faster or to automate the decision process altogether; automation wins hands down.

  One example: expectable delays with procurement. If you are able to track the flow of goods with your suppliers as well as with your distributors and enrich the information about a shipment's current location with data about the traffic or the marine weather on its planned route, then you may be able to react to bottlenecks hours, days, or even months before they occur.

3.4 The Case Studies in this Book

Case studies from different industries

By taking you through eight case studies, we would like to help you more thoroughly deepen and extend the knowledge you have acquired in the first three chapters of this book. All case studies are fictitious but are based upon a couple of decades of experience within the SAP world, which is why one or another scenario will probably look familiar to you.

Table 3.1 shows the industries in which our cases studies reside. The first column is based upon SAP Solution Explorer's classification of industries. We are using this classification as the basis because we are (where applicable) going to have a brief look at the respective value maps with each case study. Furthermore, the last column of Table 3.1 provides you with the respective NACE classification. This is because the value driver database supplied on *www.sap-press.com/3647* is SAP-neutral and therefore based upon NACE assignments.

Industry (According To SAP Solution Explorer)	Chapter	Sample Company	Industry (According To NACE)
Automotive	4	Tire Manufacturer (RunFlat Tires)	C, Manufacturing
	8	Automotive Industry (Maneki-neko K.K.)	
Healthcare	9	Residential/Nursing Home (Venerable Villas)	Q, Human Health Activities
Mining	10	Mining (Iron Bug)	B, Mining and Quarrying
Professional Services	5	Business Consultancy (Walk-on-Water Associates)	M, Professional Activities
	11	IT Service Provider (SAP Tuk-Tuk)	J, Information and Communication
Retail	6	Consumer Electronics Store (Sell-your-Soul)	G, Wholesale and Retail Trade
	7	Food Trade (Leech Oil)	

Table 3.1 Assigning Case Studies to Industries

Planning means replacing randomness by errors.

Attributed to Samuel Goldwyn, cofounder of Metro-Goldwyn-Mayer

4 Flexible Planning

The Gorges de Véroncle was some three miles away from the monastery; by the time he arrived there, Derek had trudged up the bed of the dried-out stream for almost two hours. He walked on, and when he reached the sixth mill, the shade of the evergreen oak looked like it was the perfect place for a picnic. Derek pensively gazed at the set of stones: the bedstone worn off to its steel ring, the remains of the broken, moveable runner stone, and the rusted spindle that still sat on its shaft in the eye of the millstones. In his walking guide, he had read that a certain Aymar d'Astouaud had first dammed the water of the Véroncle; in the course of the centuries to follow some ten mills had been established, the last of which had been in operation until 1910.

Figure 4.1 Ruined Mill in the Gorges de Véroncle, Département Vaucluse, France

But like the Sénancole, the stream that once supplied water to the abbey he was staying at, the Véroncle had fallen dry a long time ago. Small earthquakes had changed the course of subterranean rivers in the karst and made it increasingly difficult to keep the mills up and running. In their despair, the millers had tried to handle the frequent water shortages by installing a number of small dams and weirs. They had also replaced horizontal shafts with vertical ones that operated with a reasonable degree of efficiency even with low quantities of water. But in the end, all the effort and creativity invested had been to no avail; water had disappeared from the canyon, and with the water the craft of millers had vanished from this part of Provence.

Letting go wasn't hard just for individuals but for companies as well. An aphorism of the Dakota Native American tribe said that one should dismount from a dead horse, but Derek's experience in IT consulting had taught him that most organizations first try to solve the problem by optimizing the harness or by reviewing the job profiles of horse and rider.

Nevertheless, the water hadn't suddenly vanished from here; the process had taken centuries. A perceptive observer could have read the signs and might even have discovered the link between earthquakes and water volume. The first miller to notice that early enough could still have got a good price for his business and invested that money in a modern steam mill.

To put it another way, if you are able to continuously monitor the validity of your planning assumptions (forecasts) and models, then you can spot forecasting or planning errors a lot sooner than others and handle such errors in a more timely, laid-back, and wise manner. However, getting this message through to the head of operations would be a different story; and in all other areas of Derek's company, there were many entrenched experts who wouldn't want to hear about such strange concepts and would rather pin their hopes on re-engineering dead horses or on believing that the rivulet in front of their mill would once again turn into a mighty river.

"And they will always find consultants to encourage them to do so," Derek grumbled to himself.

Planning and big data

Planning plays an important role in our lives; most of us—consciously or unconsciously—plan before we act. Short-, mid-, and long-term planning is also an important function in most organizations. In this chapter, we will review a planning-related business case for big data/SAP HANA.

Before we do this, we will clarify what is meant by *planning*; we will also examine the relationship among planning, forecasting, and modeling. To explain all of that, we are going to use a very simple example: making a cup of tea. We will then work out what this simple example could teach you in terms of your professional environment.

Sample scenario

After discussing planning, forecasting, and modeling, we will introduce a more business-focused—fictitious, but realistic—planning scenario. The lessons learned from this scenario will help us come up with requirements for a (SAP HANA-based) solution and decide which tools could be best used to satisfy our needs. Of course, we won't forget to take into account how such a solution could create shareholder value.

Which implementation scenario?

Toward the end of this chapter, we will discuss which of the implementation scenarios introduced in Chapter 2, Section 2.2 could be the best option, which specific products from SAP could be used to fill that architecture model with life, and the trip wires you should keep an eye out for when building the solution.

Because this is the first case study in this book, we are going to spend a bit more time discussing the scenario and resulting insights than we will for later case studies. The structure of the following chapters will then be more or less in line with this one. Furthermore, as mentioned at the very beginning of the overview, we will also use this opportunity to introduce a couple of key terms (plan, forecast, and model) and concepts (such as process-oriented instead of content-oriented data models, also called *semantically neutral* models in later chapters); doing all of that now will make this chapter a bit longer than the others but save time and space later.

4.1 What Does "Planning" Mean?

Cycling from the North Cape to Catania in Sicily, dining at our favorite French restaurant, or just having a cup of tea at the office: we cannot act without having planned first. A bicycle trip of nearly 2,500 miles certainly requires some planning. We must decide on the route, check timetables for buses, trains, or ferries, consider the weather and what kind of clothing we may need, and book hotel rooms. Similarly, a cup of tea will not find its way to our lips without a sequence of activities that—at first glance—are not related to the desired result (having a cup of tea).

Planning as defined by Wikipedia

You need to stand up, leave the office, walk over to the kitchen, find and fill the electric kettle, switch it on, get a clean cup and spoon, find a teabag and maybe get some sugar from the cupboard and milk from the fridge, put the teabag into the cup, fill it with hot water, let it draw for 90 seconds, add milk and sugar, and then walk back to your desk (where you can finally spill it all over your keyboard and your important documents).

Planning is based upon models

But why do we open the cupboard and the fridge (after all, we don't usually expect to find a freshly made cup of tea in there)? Why do we turn on the kettle? In both cases, we are relying on a *model* that leads to certain assumptions about the consequences of activities under certain circumstances. We open the cupboard because our model tells us that we need certain ingredients for making tea and we are going to find them in there. We turn on the kettle because we assume—based upon past experience or hearsay—that flipping the switch will make the water boil (which is only going to work if the device is fully functional, if it has been plugged in, if the electricity bill has been paid, if there is no power outage in the area, etc.).

Forecasts are the input data for models

Why do we believe there will be some teabags left in the cupboard and that the electricity bill has been paid? We simply rely on *forecasts*, which are specific predictions about what will happen in the future. Forecasts provide us with input data for models and are therefore (alongside models) the second pillar upon which planning rests; only if we have an idea about what the future will hold in store will we be able to decide which actions may or may not serve the purpose of reaching our goals.

4.1.1 Planning, Modeling, and Forecasting

Deciding what to do for the best

Hence, planning implies that we have an idea (forecast) about the future. We use this forecast, and models, to tell us which actions would make sense in terms of achieving our objectives. Consequently, our meditation about a cup of tea will lead us to three important insights about planning:

- **Planning is critical**
  Without the ability to make forecasts and to anticipate the effect of our actions, we would not even be able to survive (as we would not have the faintest idea of where to go and what to do to get hold of food and drink), let alone be in a position to undertake more complex adventures, such as traveling nearly 2,500 miles on a bicycle.
- **All our actions are based upon forecasts and models**
  Forecasts are related to certain environmental parameters (there will be milk in the fridge and sugar in the cupboard). We use these environmental parameters to feed models that in essence represent cause–effect relationships (switching the kettle on will make the water boil). As an output, our models deliver (theoretical) results, telling us how our own activities will change the course of things in the world around us. We then use these results to make up our minds on what to do/not to do.
- **Incorrect forecasts and assumptions should be corrected as early as possible**
  Our assumptions about environmental factors and about cause-effect relationships may be right or wrong. If we are right, that's fine. But if some of our forecasts are wrong, well, the sooner we learn about it, the better.

  Had we known that a rat fink of a colleague sitting in the cubicle across the aisle just took the last teabag five minutes ago—there actually was some kind of evil look in his eyes when he smiled and raised his mouth-watering cup to us a few minutes ago, wasn't there?—then we could have gone to the floor above and stolen the last teabag from the kitchen there. But now the receptionist has found that one before us, and we (why does it always have to be us?) are the ones who have to do the shopping!

Hence, there are two things that *can* go wrong (and according to Murphy's law *will* go wrong) in planning: our forecasts might be flawed, and our models might not represent reality but instead just wishes, fantasies, or delusions.

4.1.2 Business Planning

Core function in business

Planning is not limited to our own personal world. Most of us are very familiar with planning within a business environment, and some of us (those who know Scott Adams' cartoon character Dilbert, for example) have come to the conclusion that major organizations spend most of their time on nothing but making and changing plans. But without some kind of concept regarding what the future will hold (forecasts) and which actions will have what kind of effect on it (models), businesses—much like individuals—would have a hard time surviving.

Commonalities between personal and business planning

Among other plans, every business needs to make decisions about which products or services to provide in what quantities and to which markets to supply these products. As in our personal life, our insights about planning apply to businesses as well.

4.2 Scenario: Sales and Results Planning with a Multinational Tire Manufacturer

SAP HANA as an early warning system

In this chapter, we are going to explain how you can (via an appropriate data architecture) properly separate forecasts and models from plans and how you could use an SAP HANA-based big data solution as an early warning system for forecasting and modeling mistakes.

In the following case study, we take a closer look at a company whose production and financial plans rest upon—among other factors—exchange rate forecasts. Using a numerical example, we demonstrate the far-reaching effects of getting these exchange rate forecasts wrong and (even worse) of noticing this error too late. Using this setting, we are going to explain how and why multilayered, flexible planning models in SAP HANA can help you identify incorrect forecasts sooner and initiate adequate measures faster.

RunFlat Tires Inc. (RFT) is a (fictitious) US-based manufacturer of tires for all types of vehicles (cars, trucks, tractors, dump trucks for the mining industry, motorcycles, bicycles, wheelbarrows, etc.). Apart from selling to the makers of such vehicles (original equipment manufacturers [OEMs], such as the automotive industry), they also provide their products directly to end customers via a variety of distribution channels (garages, DIY markets, online shops, etc.). In the context of this case study, we are only considering one of RunFlat's products: the new winter tire, Super X7700.

Although they are based in the United States, RFT mainly serves European markets (France, Germany, Poland, Switzerland, and the UK). Their products are made in factories in Western Poland and Northern England and are sold into their target markets via a marketing organization in Switzerland. The marketing organization buys the tires from the two factories and then sells them to local distribution centers within the individual countries.

Why exchange rates are important for RFT

One of the key decisions RunFlat has to make and review on an ongoing basis is where to make the tires it is planning to sell. For a couple of reasons, exchange rates are outstandingly important in this context:

- **Revenue in USD and exchange rates**
  RFT is a US-based company, so they'll have to report the profits or losses of their Swiss-based marketing organization in US dollars (USD), even though they originally arrive in Swiss Francs (CHF). Hence, their revenues in USD are exposed to some currency risk.
- **Revenue in CHF and exchange rates**
  In the same way, revenues generated in Switzerland are subject to exchange rate risk. As tires are not only sold in Switzerland but also in France, Germany, Poland, and the United Kingdom—that is, against Euros (EUR), Polish Zlotys (PLN), and British Pounds (GBP)—revenues in CHF are not simply *Price* × *Quantity* but instead *Price* × *Quantity* × *(Exchange Rate EUR/PLN/GBP:CHF)*. Revenues in USD are then *Price* × *Quantity* × *(Exchange Rate EUR/PLN/GBP:CHF)* × *(Exchange Rate CHF:USD)*.
- **Costs and exchange rates**
  Both European plants are in countries that are not within the European Monetary Union (i.e., countries that don't use the Euro as their currency). This once again means that RFT might gain or lose from currency fluctuations on the purchasing side—even within Europe itself. This risk results from the fact that the Swiss-based marketing organization reports profits or losses in CHF but has to pay for tires from the factories in GBP or PLN.

Planning cycles with RFT

RFT's planning cycle is built upon three revolving plans that are reviewed on a regular basis, going through different versions from draft to final approval:

- During each fiscal year (January – December), RFT has four quarterly planning cycles. In the course of these planning cycles, RFT updates its annual, month-based sales, production, and financial plans.
- In the course of the last of these quarterly planning cycles (happening during the last two weeks of December), RFT's long-term (five years, annual) plan will be reviewed and extended by one year.
- The long-term plan also serves as the basis for RFT's mid-term (three years, quarterly) plan. This mid-term plan for the following three years will then be broken down into months and become the new annual plan for the following year.

RFT uses SAP BPC

All these planning activities take place within SAP Business Planning and Consolidation (BPC) for SAP NetWeaver. RFT's BPC system runs on SAP Business Warehouse (BW) on SAP HANA and receives data—for example, historical sales quantities—from RFT's ERP system (SAP Business Suite on SAP HANA). To report planned and actual data, RFT uses various business intelligence solutions (SAP Crystal Reports, SAP BusinessObjects Dashboards, etc.).

SAP BPC components with RFT

The decision to implement SAP BPC for SAP NetWeaver had been made quite some time ago. The main reason was that RFT's complex and entwined planning processes could be modeled as business process flows and thus be monitored easily and systematically. Business process flows, the Process Monitor, and associated reports facilitate the coordination of global planning activities and provide a framework to centrally monitor planning status. On top of that, Consolidation Central and more specifically Currency Translation drastically simplify planning at a group level.

SAP HANA was implemented a short while ago, and although RFT does benefit from faster reporting performance, it has not yet implemented any use cases that would only be realizable with SAP BW or SAP Business Suite on SAP HANA.

4.2.1 Forecasts and Models with Sales, Results, and Cost Planning

RFT's long-term sales plan

The diagram in Figure 4.2 shows an extract from RFT's five-year sales plan (as of December 2011). Figure 4.2 shows only planned sales for RFT's new Super X7700 tire in some key markets throughout the year 2012.

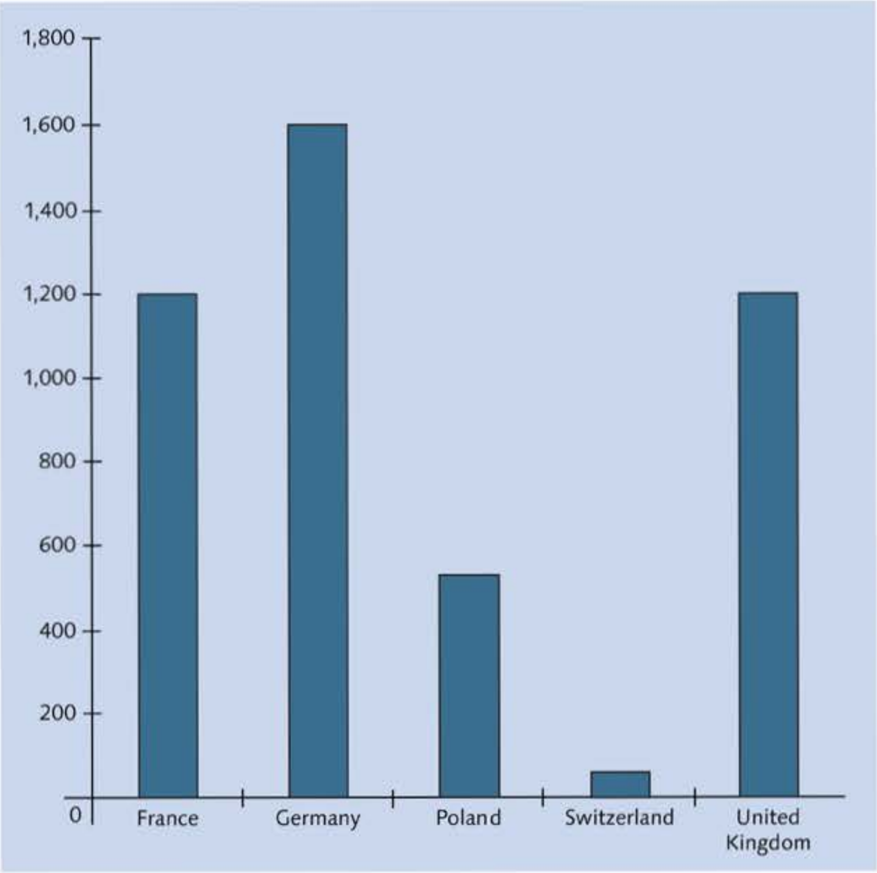


Figure 4.2 Sales Plan 2012, Annual Values

The basis of RFT's sales plan

A specialized market-research firm supplying forecasting services to RunFlat has calculated the numbers in Figure 4.2 via a complex economic model (polynomial regression) designed on SAP Predictive Analysis (PAL).

This model uses a number of market-specific input parameters (forecasts), including the following:

- Forecasted gross domestic product (GDP) growth rate
- Forecasted inflation rate
- Income distribution
- Long-term trends about available net income
- Long-term trends about RFT's market share in various product groups
- Long-term trends about how customer demand is split between products within the same product group

[+]

Forecasting Errors, Modeling Errors, and Solution Architecture

We mention the model's input parameters because this knowledge would enable RFT to challenge not only the forecasted sales figures but also the model that produced them. If all relevant input parameters have developed as predicted but sales figures still stay behind expectations, then calling the validity of the model into doubt is very reasonable.

If instead input parameters deviate from their forecasted values but the model—fed with actuals instead of forecasts—would have delivered sales figures that are very close to actuals, then one should instead ask how the forecast for the relevant input parameters (like the future GDP) could be improved.

Hence, when designing the solution architecture you need to know whether you are working with a plan, a plan plus forecasted input values, or a plan plus forecasted input values plus the models the plan has been generated with. If you only have a plan and/or forecasts, you could build an early warning system to detect plan/actual deviations. If you also have the model, you can test its accuracy and discover which parts of it need to be reviewed.

Sales per quarter and per month

Based upon the sales forecast shown in Figure 4.2, RFT has broken down the numbers for 2012 to 2014 into quarters using *Seasonalization* (another standard functionality in BPC). These quarterly figures were then (again using Seasonalization) apportioned to individual months (see Figure 4.3). Because the Super X7700 is a winter tire, it is predominantly bought in autumn and winter.

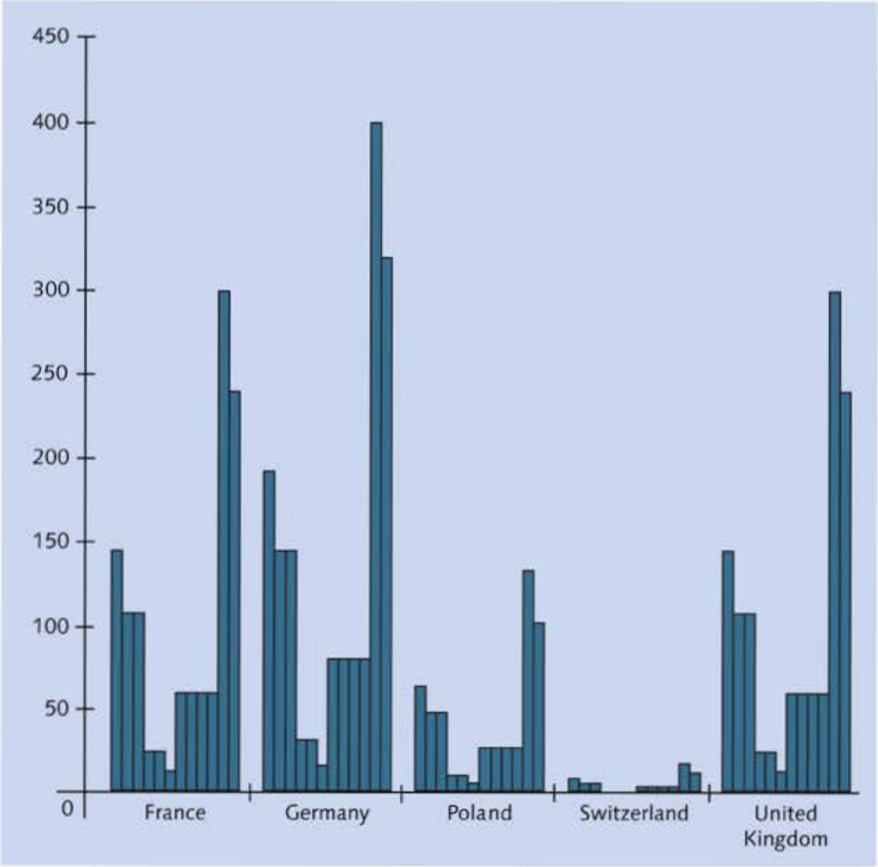


Figure 4.3 Sales Plan 2012, Monthly Values

Sales and results plan

Because sales prices are fixed (they are agreed upon with distribution partners in advance), planned revenue can be derived from planned sales. Furthermore, the internal transfer prices that the marketing organization in Switzerland is going to pay to the plants in Poland and England for the imported tires are also fixed in advance. This means that not only planned revenue but also planned costs for the marketing organization in Switzerland (assuming a certain distribution of production volumes onto the plants) can be calculated from planned sales.

When calculating revenues and results from sales, we also use models. These models, however, do not use any statistical algorithms. The simple reason for that is that all relevant parameters (sales and transfer prices) are known, and therefore—mathematically speaking—there are no random variables. These models are therefore deterministic; one can easily build them in SAP BPC without using any statistical tools (such as SAP PAL or R).

4.2.2 Exchange Rate Forecasts with RFT

Up to this point, RFT can only determine revenues and costs as amounts in foreign currency. To convert revenues (for example, in GBP) and costs (for example, in PLN) into one common currency (for example, CHF) and then add them up, RFT needs information about (future) exchange rates. Furthermore, exchange rate forecasts are required to convert the Swiss marketing organization's results into USD and thus to produce a results plan for the group as a whole.

Exchange rates forecasts supplied by partners

For this reason, exchange rate forecasts are essential for RunFlat. When it comes to exchange rate forecasts, the company calls on the help of recognized external experts. These experts also use models (of varying degrees of sophistication) to generate their forecasts. However, RFT knows nothing about the inner workings of these models nor about their input parameters (unlike with the sales forecasts). If actual exchange rates should deviate from the forecast, RFT is therefore not able to find out whether this deviation is caused by an incorrect forecast of the input parameters or by a lousy model.

Exchange rate forecasts as of December 2011

For planning purposes in December 2011, RFT was using the exchange rate forecasts shown in Figure 4.4. For the sake of simplicity, we assume the following:

- Buying currency A for currency B can be done at the reverse rate of buying currency B for currency A (that is, we don't differentiate between bid price or ask price).
- There are no transaction costs.
- There is only one relevant exchange rate applied to transactions.

According to Figure 4.4, RFT's external consultants are expecting the EUR and the USD to rise against the CHF by about 24% and the GBP to rise against the CHF by about 29%, whereas the PLN is going to lose some 28% of its value against the CHF. They also assume that these changes will apply evenly over the year.

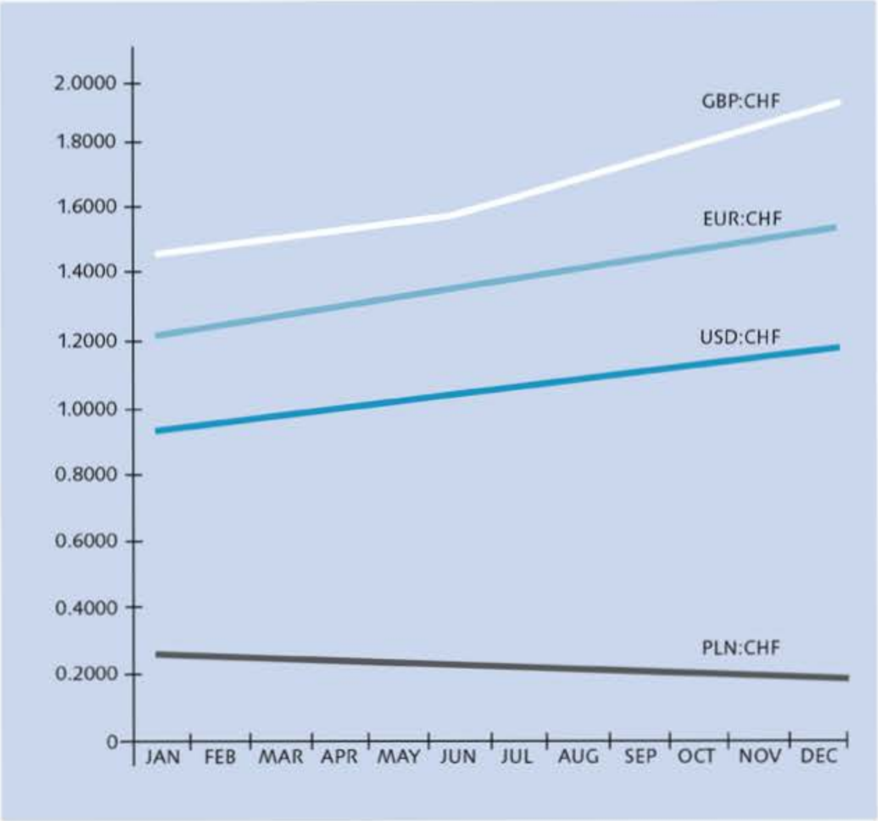


Figure 4.4 Forecasted Exchange Rates 2012, Monthly Values

Exchange rates determine production plans

Although RFT needs exchange rates to produce its results plans in CHF and in USD, its planning activities do not end there. Depending on the forecasted exchange rates, it makes sense for RunFlat to produce more or less product in certain countries or to not even supply specific markets.

4.2.3 Models for Production, Results, and Financial Planning with RFT

Based upon the sales plan in Figure 4.3, RFT is planning to sell a total of 4.6 million tires in Germany, France, Poland, Switzerland, and the United Kingdom. Because they now also have exchange rate forecasts, they can run through different production scenarios.

Making everything in Poland

On receipt of a very thorough analysis using a (deterministic) optimization model, RFT's top management has decided to completely satisfy expected demand with Super X7700 tires made in Poland and at the same time stop making tires of this type in England. This does not, however, mean that the factory in England will stop making anything. Over there, the company is going to focus on products for which the plant in England has got a comparative cost advantage or products that cannot be made in Poland for technical reasons.

Other follow-up decisions

Having made this commitment in terms of production planning (plus corresponding production plans for other tires), RFT has also come to a number of other production-related decisions:

- The planning of required production capacities for 2012 and decisions about short-term investments needed to make these capacities available
- The planning of running times, service intervals, and maintenance downtimes of machines
- The planning of head count throughout the year, aligning existing agreements with temporary employment agencies
- The purchasing of other external services (such as storage capacity, transport services, and so on)

Results plan determines financial plan

Once production planning is defined, results planning is frozen. Based upon these (and other corresponding numbers for other products and product groups), further financially relevant plans and publications are prepared. These include the following:

- A cash-flow plan that serves as a basis for determining which amounts RFT will have to secure in 2012 as additional liquidity in the United States or how much extra cash could be invested in the short- or midterm
- A consolidated balance sheet plan for the headquarters in the United States and the group as a whole
- Formal announcements to stock markets in Europe and the United States, disclosing RFT's current business outlook

4.3 Planning Errors: Costs, Risks, and Opportunities

Dependencies within the planning chain

The planning chain described in the previous section contains a number of dependencies:

- Certain macroeconomic parameters—together with a model—generate planned sales volumes.
- Those sales volumes per definition lead to revenues and costs in foreign currency and—together with exchange rate forecasts—to a production and a results plan in company currency (CHF) and group currency (USD).
- Production planning is significant for quite a few production-related decisions.
- In addition to production planning, results planning also has an impact on important decisions/measures in finance.

Ultimately, the accuracy of incoming forecasts and the correctness of the models used are of extreme importance for RFT's future success. Let us therefore take some time to find out what could go wrong, using a fictitious numerical example to determine what that would mean for RFT.

4.3.1 Problem: Risks with Forecasting and Modeling

All of the forecasts and models that RFT's plans rest upon could either be right or wrong. If individual ones or all of them are wrong, then a beautiful plan could collapse back upon itself like a failed chocolate soufflé.

Forecasts and models with RFT

Keeping an eye on certain facts (that is, forecasts and models) might therefore make a lot of sense for RunFlat. To stay on top of things, we would like to categorize this collection using two criteria:

- Which plan (sales, revenues, costs, exchange rates, production, financial, etc.) is each fact related to?
- Is the fact a plain vanilla forecast, or is it a statement about something more complex—that is, a model?

Matching up forecasts and models

Table 4.1 provides you with an overview. Our list is probably not complete; its sole purpose is to raise your awareness of how important forecasts and models are in planning and of how often they are overlooked or at least not explicitly named. We have ignored some more exotic scenarios: in the future, aliens could land on earth, start selling flying vehicles, and as a consequence eradicate the demand for tires altogether.

Plan	Forecast	Model
Sales	GDP Growth, Inflation, Income Distribution, Net-Income (Trend), Market Share (Trend), Customer Demand Split, Competition (Behavior), Weather	Sales Forecasting Model (Stationarity of Stochastic Process)
Revenues	Sales Prices	Payment Terms, Payment Behavior
Costs	Material Prices, Bills of Material/ Routings (Waste)	Linear Increase
Exchange Rates	Unknown (Black Box)	Unknown (Black Box)
Production	Downtimes, Capacity Sufficient, Industrial Action	Routings, Capacity Planning Model
Finance	Interest Rates (Borrowing/ Lending), Financial Risk	Free Capital Flow, Capital Markets

Table 4.1 Categorizing Facts (Forecasts and Models)

Nevertheless, the list is still quite impressive:

- **Sales planning (forecasts)**
  - Actual figures for GDP growth, inflation, income distribution, net income, and market share in all relevant countries might differ substantially from RFT's expectations.
  - RFT's competition could spend more or less money than anticipated on advertising articles that are positioned to compete directly against RFT's Super X7700 or could even launch new competitive products.
  - New players from emerging economies like China or Brazil could enter the market.
  - Winter could start a lot milder than usual, which could time shift demand (making people buy winter tires later). With less snow

than normal, customers could refrain from buying new winter tires altogether.

- **Sales planning (models)**
  - Customers might still buy the planned number of tires from RFT but may want to buy different (cheaper or better) products.
  - Although all of the preceding parameters may have been forecasted correctly, something could be wrong with the model that derives customer demand from these figures. There might be no such link at all, or maybe there was one in the past few years, but in the meantime some of these factors have stopped influencing customer behavior. Mathematically speaking, this would mean that the underlying *stochastic process* (random process) might be nonstationary. Or in plain English, the rules of the game have changed.
  - Because RFT gets its sales forecasts from an external service provider, RFT itself doesn't really know how the forecasting model works. RFT would, however, be able to recognize if something was wrong with that model (this would be the case if sales or sales volatility were not within the expected margin of fluctuation even though all input parameters were behaving as expected).
- **Revenue planning (forecasts)**
  - Although RFT's management assumes that sales prices are fixed, the company might experience unexpected pressure to reduce them in the short run.
  - New Asian competitors might offer their products for dumping prices, or parallel imports might appear on the market, forcing RFT to react. This could reduce revenue even if sales volumes stayed the same.
- **Revenue planning (models)**
  - Although prices might remain untouched, a difficult economic environment could drive some of RFT's distribution partners out of business. This could leave RFT with a longer cash cycle or a higher than usual share of uncollectible bills.
  - From an accounting perspective, this would not affect revenues (bad debts still count as revenue) but would lead to allowances and

thus eat into RFT's contribution margins. Furthermore, our simplified model assumes that 100% of all sales lead to cash in the same month; thus even delayed payments would have an impact—if not on net operating income then on cash flow.

- **Cost planning (forecasts)**
  - Prices for raw materials, services (e.g., electricity and transport), or labor could change, leading to higher or lower production costs.
  - Even if prices stayed the same, input quantities (per tire) could be higher than put down in bills of material (BoMs; for materials) and routings (for services). A new, improved, but yet untested production process in one of the plants could lead to more waste, higher downtimes, or more need for unplanned manual problem fixing. All of these factors could lead to higher variable costs.
- **Cost planning (models)**
  - RFT's cost planning is based upon the assumption that costs increase linearly with production volume. This is usually only the case within a certain volume bandwidth.
  - For very low or very high production volumes (that is, once the plants are approaching their minimum or maximum capacity), marginal costs for each additional tire made could be lower/higher than previously experienced (exponential decrease/increase).
  - If demand turns out to be substantially less or more than expected, then the model would no longer be applicable, and variable costs would change.
- **Exchange rate planning (forecasts)**
  - With the sales forecast, RFT knows which input parameters their advisors use but not *how* they use these input parameters.
  - With the exchange rate forecast, they know nothing about the input parameters. RFT is therefore unable to monitor any input parameters or to keep an eye on whether or not they behave as expected.
- **Exchange rate planning (models)**
  - As with the sales model, RFT knows nothing about the inner structure of the exchange rate model that generates the forecasts it uses.

  - Also as with the sales model, RFT would still be able to recognize if something was wrong with the model's output (that is, if exchange rates or exchange rate volatility was not within the expected margin of fluctuation).
  - With exchange rates, however, RFT would not be able to tell whether this failure was due to incorrect forecasts regarding input parameters or incorrectly assumed cause–effect relationships; if the latter was the case, they would have even less of an idea about what the problem might be related to inside the model.
- **Production planning (forecasts)**
  - New equipment might give RFT teething problems.
  - Machines that are well beyond their design lifespan might fail more often.
  - The overall capacity of plants might also suffer in the wake of major accidents or industrial action.
  - All of that would lead to higher downtimes and would not only result in higher per-unit costs but also limit the plant's production capacity. As a seasonal business, this might endanger RFT's ability to satisfy customer demand (especially at peak times) and hold its market share.
- **Production planning (models)**
  - The production-planning model might not be correct—that is, the relationship between input factors (for example, machine hours) and output volume as defined in routings (a *routing* is the definition of a production process used to manufacture materials/products or the steps and actions involved in delivering a service) might not reflect the plant's real ability to make tires.
  - When push comes to shove, more time, more machines, and more staff than expected might be required. As most of these factors cannot be adapted in the short term, output might be lower, once again leading to RFT being unable to sell planned quantities.
- **Financial planning (forecasts)**
  - RFT's financial plans are based upon planned revenues and costs and would therefore lead to certain amounts being invested or

  borrowed throughout the fiscal year. Borrowing costs or income from these investments would be a component of RFT's liquidity planning but might deviate from plan if interest rates for borrowing or lending or related risks change.
  - New financial risks could lead to RFT's lenders no longer being willing or able to supply cash or—even worse—RFT's debtors no longer servicing bonds that RFT invested in.
  - RFT's lenders might become concerned about money and so refuse to roll over or extend existing credit lines.
- **Financial planning (models)**
  - The model assumes that money can be moved freely and without restrictions between sales markets, distribution centers, and production sites.
  - Past experience, not only in developing countries (remember the Cyprus crisis or the more recent sanctions against Russia), has shown that this could change overnight.

We worry about abrupt changes for the worse

Most of these items could theoretically lead to RunFlat doing either better or worse than planned. In most organizations, management is usually more concerned about the latter.

Some events (such as a strike) are quite likely to catch management's attention anyway, even without a monitoring solution in place. Others, however (such as several short downtimes on the shop floor, a gradual deterioration in a customer's payment behavior, or higher exchange rate volatility), might well go unnoticed for a while; it's these ones we are going to have a closer look at in this chapter.

Black swans

Apart from an alien invasion, there could be other, more likely but still unforeseen, events that could have a significant impact on the preceding parameters and cause-effect relationships. Such events are not captured by any of the lists discussed so far (as one usually doesn't even think of them as a theoretical option) and are often called *black swans.* Classic examples of a black swans are the Russian currency crisis around 1998, the worldwide financial crisis with Lehman Brothers' insolvency (2008), and the 2012 to 2013 Cypriot crisis (and the resulting confiscation of bank deposits there).

What is going wrong? And why?

If something goes wrong, we would also like to know whenever possible why it has gone wrong. If production output goes down unexpectedly, we would like to get an alert about the symptom, suggestions about possible causes, and some recommendations for therapy or resolution of the issue. Quite ambitious, isn't it?

Root cause analysis

Maybe, but SAP HANA can help. The challenge sounds far less scary if you imagine that it isn't overworked and underpaid employees who have to keep an eye on hundreds of parameters (on top of their everyday jobs). Imagine, instead, that a never-tiring appliance (like SAP HANA) that can digest vast amounts of data in the twinkle of an eye could handle such an exhausting mission. You could then continuously scan an almost limitless number of values for deviations from normal patterns or—in statistical terms—for outliers.

[«]

Outliers

Outliers are observations, measurements, or values that are unexpected, unusual, or (in a graphical representation) distant from the rest of the data. Outliers may occur for a number of reasons, for example, pure chance, measurement errors, or a flaw in the forecast or assumption model.

However, before hunting down a diagnosis and medication we will have a closer look at the disease itself and get an idea of how much damage a black swan—even if it is just a swan's cygnet—can cause to RunFlat. We will use forecasted exchange rates as an example for this.

4.3.2 Numerical Example

Relevant exchange rates hardly changed in 2012

In 2012, actual exchange rates developed as shown in Figure 4.5 (source: *www.oanda.com/currency/historical-rates*).

Other than as forecasted, there was very little movement on currency markets. Most exchange rates stayed more or less where they were at the beginning of the year. This rate pattern (all other things being equal) would have pruned RFT's net operating income from 323.9 million USD to 115.5 million USD (that is, by about 64%; see Figure 4.6).

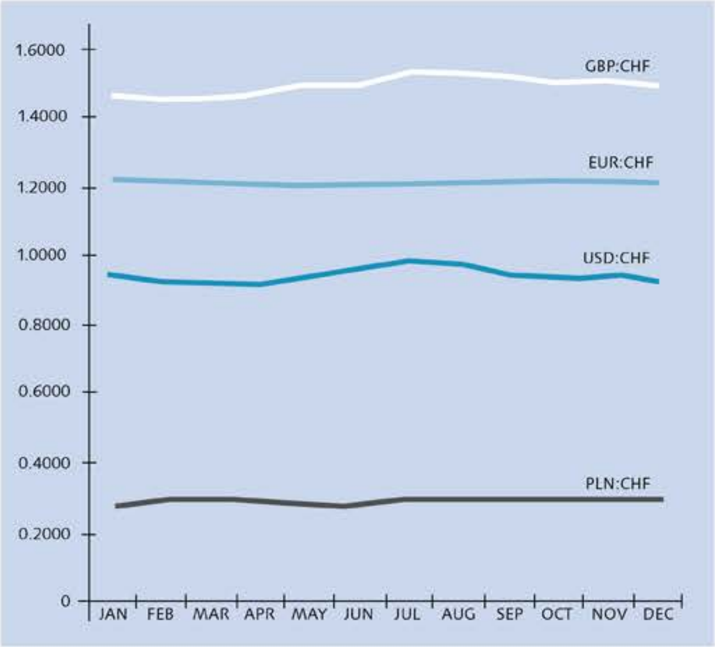


Figure 4.5 Actual Exchange Rates 2012, Monthly Values

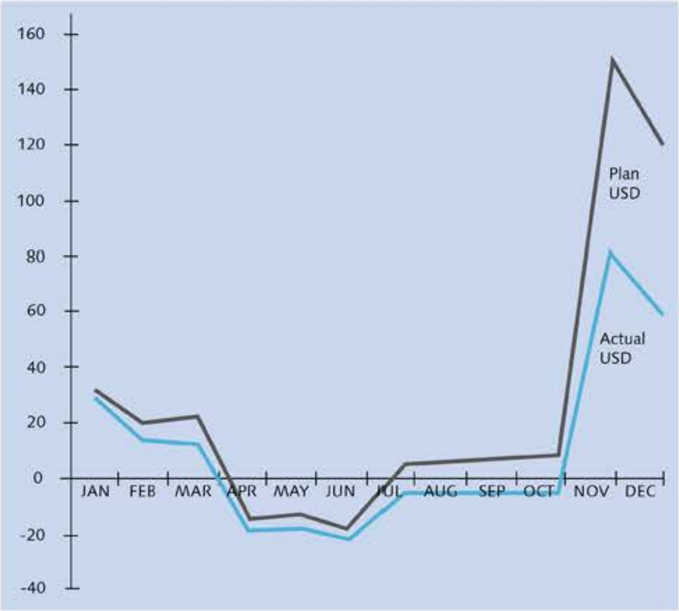


Figure 4.6 Planned versus Actual Net Operating Income 2012 (USD), Super X7700, Monthly Values

Although still dire, things look just a little bit brighter in USD than in CHF, because the USD:CHF exchange rate developed favorably from RFT's headquarters' perspective.

The prince and the geranium

But things could have been worse. Let's imagine that instead of delivering an heir to the English throne, as was their duty, Prince William and his Kate got into a terrible fight at the Queen's birthday party in April 2012. They announced their divorce on the first of May. At the same time, William—after consulting with his father—declared his intention to marry a geranium in the near future. English nobility and wealthy royalists were beside themselves with rage, and fled the kingdom in great numbers, moving their belongings into what seemed to them a haven of decency in these chaotic times: Roman Catholic Poland.

The resulting shifts in financial markets sunk the British Pound; at the same time they let the Polish Zloty soar to unprecedented heights, as shown in Figure 4.7.

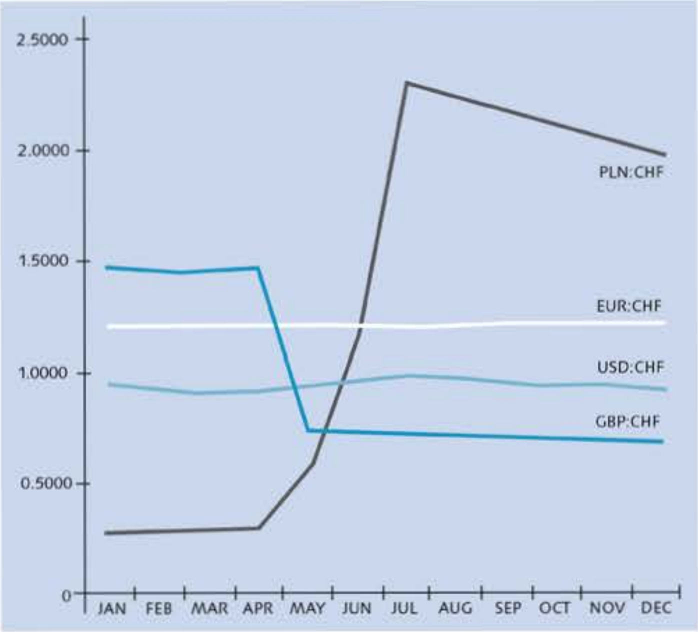


Figure 4.7 Actual Exchange Rates 2012, Geranium Scenario, Monthly Values

A perfect black swan

With the actual exchange rates discussed at the beginning of this paragraph, RFT's future already looked a bit gloomy (as shown in Figure 4.6).

This new exchange rate scenario would now be a real disaster for RFT. The company would suddenly have to swallow a loss of almost 2.5 billion USD (see Figure 4.8), which would almost certainly take them into bankruptcy. It would be a perfect black swan (which would even catch the Queen—the official owner of all mute swans in British open waters—by surprise).

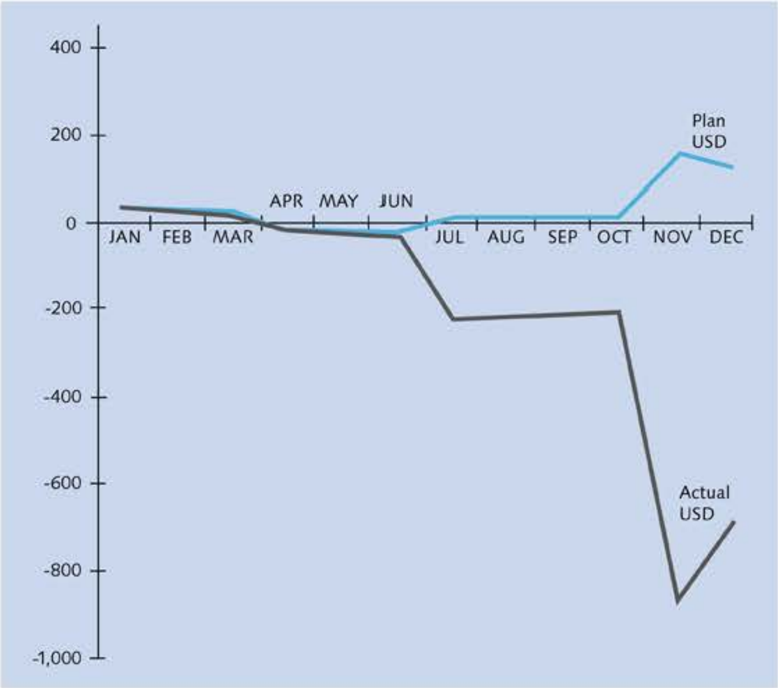


Figure 4.8 Planned versus Actual Net Operating Income 2012 (USD), Super X7700, Geranium Scenario, Monthly Values

You might argue that Prince William loving a potted plant more than his wife is not a very likely thing to happen, but if you pause for a moment you could imagine quite a few events that are less exotic but not less surprising and that can lead to major distortions in currency markets. It doesn't therefore take a lot of imagination to understand that it can be very hard to determine the true costs of planning errors. Usually, these costs can only be known in hindsight, and even in hindsight a calculation is anything but simple. You would have to know how RFT would and could have reacted had it known that its exchange rate forecasts were fundamentally wrong at an early stage. In the end, all these thoughts will lead to pure speculation. In our example, the costs of getting the forecasts wrong could therefore be anything between about 200 million and almost three billion USD. They could have caused a veritable dent in RunFlat's net operating income but could just as easily have driven the company into bankruptcy.

Planning errors are expensive

Although determining the true costs of planning errors is difficult, one thing is for sure: they can be substantial. Even without the black swan—which came along masked as a geranium in our case—we are talking about some 200 million USD lost for RFT. Avoiding even only 10% of that damage would probably have justified more or less any imaginable big data implementation project.

All of this raises a question: What exactly could RFT have done differently by using SAP HANA?

4.3.3 Conclusion: Keeping an Open Mind

Lessons learned from case study

Let's once again go back to Figure 4.6 and Figure 4.8. In combination, both diagrams teach us what matters when monitoring planning models:

- **A small leak will sink a great ship**
  Even relatively small planning errors could have dramatic effects on results, especially if such errors are not due to pure chance but instead indicate a systematic error in forecasts or some kind of paradigm change.
- **Aggregated numbers can be deceptive**
  Look at the lines in Figure 4.6 and Figure 4.8 (which probably represent the kind of information you would find in a management dashboard), and you'll realize that everything seemed to be in apple-pie order until about June/July 2012. This slightly strange effect is caused by two factors:
  - On the one hand, a key performance indicator (net operating income) behaved more or less as expected up to and including October 2012. Figure 4.6 shows that RunFlat was expecting losses in April, May, and June anyway, and although losses were some 19% higher than planned in May, the pattern net operating income

  showed in the course of the year was not too different from what the company expected it to be. Even worse, as things started to get better in July, a dog in the manger driveling that a storm was brewing would hardly have been taken seriously. As long as everything seems quiet, we simply don't see the need to dig deeper and might therefore simply fail to recognize problems that are developing just below the surface.

  - As shown in Figure 4.2 and Figure 4.3, most tires are sold and made in winter—that is, around the beginning and the end of the year. But in the beginning of the year actual exchange rates still remained relatively close to forecasted values, which means that the forecasting error (the difference between forecasted and actual exchange rates) was relatively small. Throughout the year—as the gap between forecasted and actual exchange rates was widening, theoretically making the forecasting error more visible—RunFlat was making and selling fewer tires. Hence, the error's effect on net operating income remained manageable. A system that is working fine in calmer waters might suddenly and, it seems, unpredictably collapse when put under stress.

- **Being aware of the present**
  The problem would probably remain unnoticed for much longer in many companies we know. Exchange rate gains or losses are often not posted immediately but in the course of closing activities (for example, using Transaction F.05), and if books are not made up every month or quarter such effects tend to pop up when bills are being settled and foreign currency is being bought or sold.

- **The earlier, the cheaper**
  If RFT had been able to recognize that exchange rates seem to remain more or less stable instead of showing clear upward or downward trends, it could have reviewed its production plans and perhaps moved certain quantities from Poland to England. This could have been done well before RFT's critical months (in terms of sales and production volume). In the geranium scenario, making a tire in England in July 2012 would have been 93% cheaper instead of being 21% more expensive than a tire made in Poland! When it comes to

developing countermeasures against unexpected developments or black swans, time is money.

- **SAP HANA as a driver assistance system**

  Maybe we are inclined to look backwards instead of forwards when making decisions because looking ahead simply means asking too much of us poor souls. What lies behind us is familiar; we know which action leads to which consequences.If so, we may—in the interest of overall transport safety—want to consider buying a car with very sophisticated driver assistance systems, a car that can not only detect other cars or pedestrians but also brake or steer autonomously.

  Maybe SAP HANA could play just that role for you.

4.4 Solution: Monitoring Forecasts and Models in Real Time

Your early warning system: some ideas

A fully fledged, implementation-ready, detailed specification for an early warning system to monitor your plan data, forecasts, and models would not fit between the covers of this book. Instead, we are going to provide you with suggestions in terms of which algorithms and products could help you build such a solution and show you where to find these inside or outside the SAP world.

Later, in Section 4.5, we will discuss implementation scenarios and basic considerations related to the solution and data architecture of such a system.

4.4.1 Related Value Maps in SAP Solution Explorer

Planning isn't industry specific

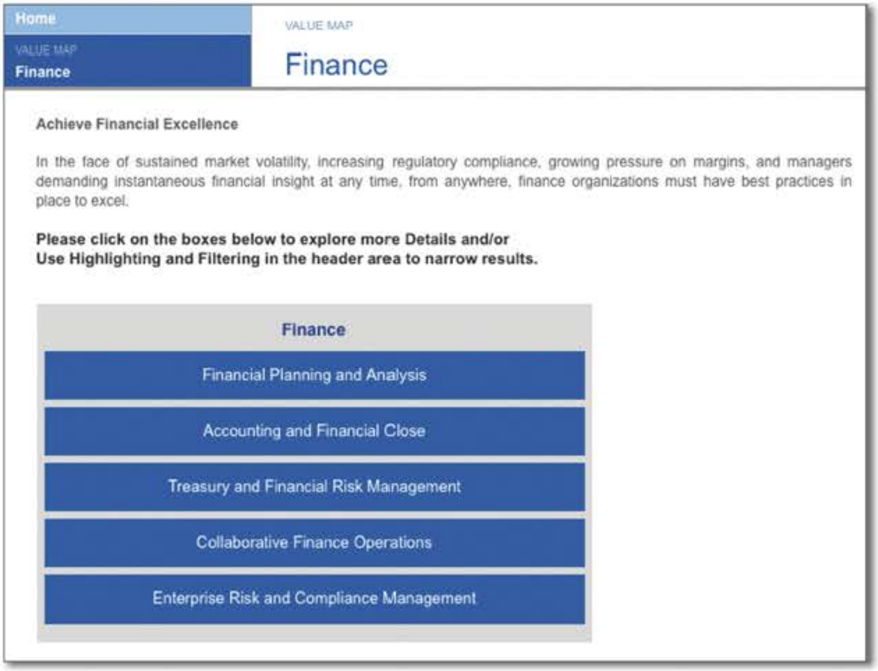
Planning takes place in each and every organization or industry. We have set up our example with a tire manufacturer, but the primary reason for doing so was to help you create a vivid mental image drawn from life. The processes we looked at are not specific to this (or any other) industry.

This is also why you will find the respective value maps in SAP Solution Explorer not *exclusively* in their industry-specific area under AUTOMOTIVE, but also with cross-industry functions (under Browse by Area of Responsibility). The planning activities we have talked about are mainly related to the Finance, Manufacturing, and Supply Chain areas on the website.

End-to-end solutions in finance

In the Finance area, we focus on the following end-to-end solutions (see Figure 4.9):

- Financial Planning and Analysis
- Treasury and Financial Risk Management
- Enterprise Risk and Compliance Management



Achieve Financial Excellence

In the face of sustained market volatility, increasing regulatory compliance, growing pressure on margins, and managers demanding instantaneous financial insight at any time, from anywhere, finance organizations must have best practices in place to excel.

Please click on the boxes below to explore more Details and/or
Use Highlighting and Filtering in the header area to narrow results.

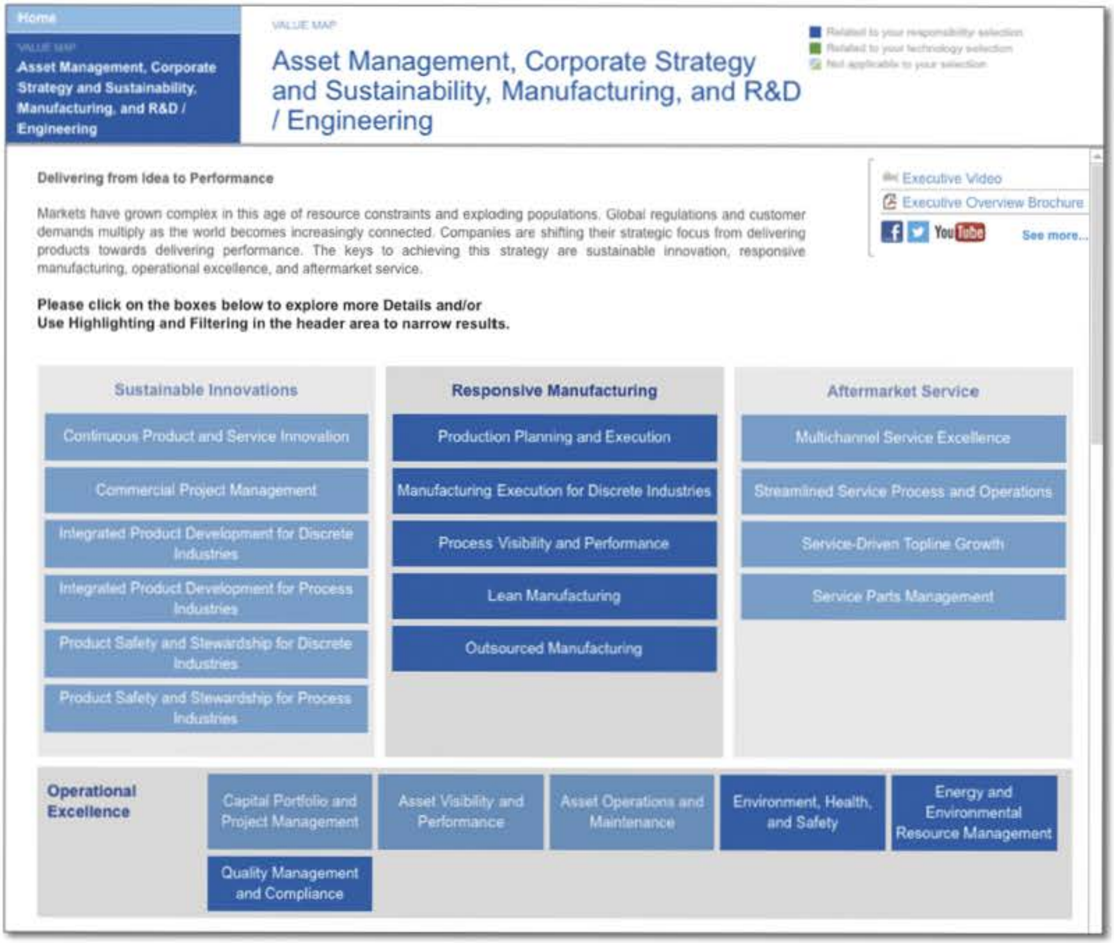
Finance

- Financial Planning and Analysis
- Accounting and Financial Close
- Treasury and Financial Risk Management
- Collaborative Finance Operations
- Enterprise Risk and Compliance Management

Figure 4.9 Finance Value Map

End-to-end solution in manufacturing

When it comes to Manufacturing, our case study aligns with SAP's Production Planning and Execution end-to-end solution (see Figure 4.10).



Home

VALUE MAP

Asset Management, Corporate Strategy and Sustainability, Manufacturing, and R&D / Engineering

VALUE MAP

Asset Management, Corporate Strategy and Sustainability, Manufacturing, and R&D / Engineering

🟦 Related to your responsibility selection
🟩 Related to your technology selection
▨ Not applicable to your selection

Delivering from Idea to Performance

Markets have grown complex in this age of resource constraints and exploding populations. Global regulations and customer demands multiply as the world becomes increasingly connected. Companies are shifting their strategic focus from delivering products towards delivering performance. The keys to achieving this strategy are sustainable innovation, responsive manufacturing, operational excellence, and aftermarket service.

Executive Video

Executive Overview Brochure

See more...

Please click on the boxes below to explore more Details and/or
Use Highlighting and Filtering in the header area to narrow results.

Sustainable Innovations	Responsive Manufacturing	Aftermarket Service
Continuous Product and Service Innovation	Production Planning and Execution	Multichannel Service Excellence
Commercial Project Management	Manufacturing Execution for Discrete Industries	Streamlined Service Process and Operations
Integrated Product Development for Discrete Industries	Process Visibility and Performance	Service-Driven Topline Growth
Integrated Product Development for Process Industries	Lean Manufacturing	Service Parts Management
Product Safety and Stewardship for Discrete Industries	Outsourced Manufacturing	
Product Safety and Stewardship for Process Industries		

Operational Excellence

Capital Portfolio and Project Management

Asset Visibility and Performance

Asset Operations and Maintenance

Environment, Health, and Safety

Energy and Environmental Resource Management

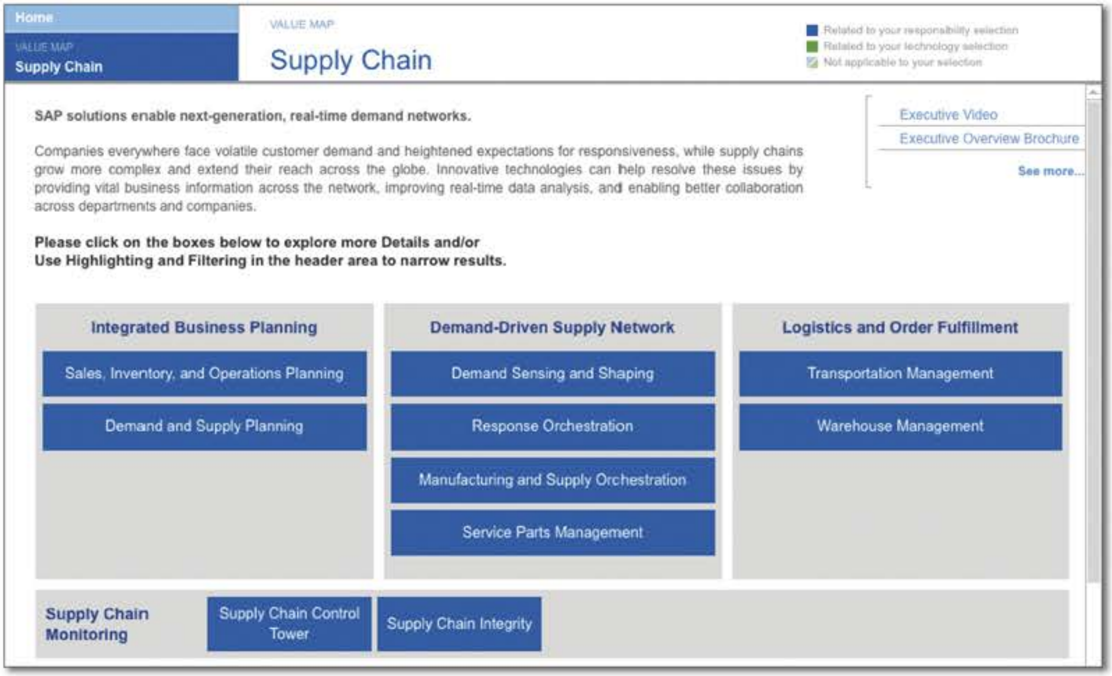
Quality Management and Compliance

Figure 4.10 Manufacturing Value Map

For SUPPLY CHAIN, we have addressed functions that are covered by the following SAP end-to-end solutions (see Figure 4.11):

End-to-end solutions in supply chain

- Sales, Inventory, and Operations Planning
- Demand and Supply Planning


Home
VALUE MAP
Supply Chain
VALUE MAP
Supply Chain
🟦 Related to your responsibility selection
🟩 Related to your technology selection
▨ Not applicable to your selection
Executive Video
Executive Overview Brochure
See more...
Integrated Business Planning
Sales, Inventory, and Operations Planning
Demand and Supply Planning
Demand-Driven Supply Network
Demand Sensing and Shaping
Response Orchestration
Manufacturing and Supply Orchestration
Service Parts Management
Logistics and Order Fulfillment
Transportation Management
Warehouse Management
Supply Chain Monitoring
Supply Chain Control Tower
Supply Chain Integrity

SAP solutions enable next-generation, real-time demand networks.

Companies everywhere face volatile customer demand and heightened expectations for responsiveness, while supply chains grow more complex and extend their reach across the globe. Innovative technologies can help resolve these issues by providing vital business information across the network, improving real-time data analysis, and enabling better collaboration across departments and companies.

Please click on the boxes below to explore more Details and/or
Use Highlighting and Filtering in the header area to narrow results.

Figure 4.11 Supply Chain Value Map

Using the preceding assignments to value maps/end-to-end solutions/solutions in SAP Solution Explorer, you can start with your own research, find out more about SAP's perspective on the processes discussed, and get an idea of which data within your SAP Business Suite might be affected and might have to be replicated/analyzed.

4.4.2 Functional Requirements

RFT's wish list

In an ideal world, we would want a system that can satisfy the following requirements:

- **Detecting outliers**

  We need a solution that can detect significant deviations from expected or observed patterns, regardless of how small or unimportant these deviations might seem to be at first sight.

  We defined the term *outlier* at the end of Section 4.3.1; the same section also contains a list of facts or parameters that RFT might want to watch. In real life, such lists tend to be a lot longer. Anyway, feel free to keep scanning all facts and forecasts plus the input and output parameters of all your models for outliers; it will be worthwhile.

- **Monitoring forecasts and models separately**
  RFT needs a solution that is able to distinguish between deviations in forecasts (as with forecasted exchange rates) and model-induced deviations between planned and actual values:

  - Deviations between the forecasted and actual values of environmental parameters can be analyzed without further complications. If there is a significant deviation, this deviation may indicate a systematic error in our forecast.
  - Deviations between planned and actual values of a model's output are a bit more complicated to interpret. They could be caused by the fact that planned data were based upon incorrect forecasts (that is, the input data of the model were different, and hence the model produced different output data). On the other hand, such deviations could also indicate some kind of flaw in the model. Even worse, both could be the case at the same time.

  To be able to evaluate the model's quality, we first need some target values, that is, planned values that the model would have delivered had it known the actual values of forecasted parameters (for example, actual exchange rates).

- **Analyzing root causes within your models**
  Planning models are often complex and consist of many interdependent building blocks. If we come to the conclusion that our plans are wrong and our models are flawed, then we want the system to tell us which forecast and/or model might have to be reviewed, reconsidered, or redefined. This is very much in line with what we expect from a warning light in our car's dashboard or from a useful error message in a computer program (let us know if you ever come across one!).

- **Evaluating risks**
  If a cog seems to be creaking or if there might be a wrench in the works, then we want to know what that means to the system as a whole before going full steam ahead. Not only do we need to understand whether the business would break once it is operating at 100% capacity, but we also need to know which of its functions are affected by the suspected problem. A red light in a plane's cockpit might indi-

cate a minor problem that can be taken care of during the next scheduled maintenance. On the other hand, that light could mean that an immediate emergency landing might be mandatory.

- **Analyzing impacts**
  If we know that something is wrong with a forecast for an environmental parameter, then we might also want to know what exactly that means for our plans. We might, for example, want to find out how RFT's net operating income would behave if exchange rates fluctuated twice as much as expected.
- **Being constantly vigilant**
  The system RFT wants to design is supposed to keep an eye on all of the previously noted elements at every second of every minute of every day. It should work quietly in the background most of the time, but escalate alerts immediately as and when required. The objective is to give us as much reaction time as possible to limit the damage or to exploit opportunities, thus helping to focus valuable management resources on real problems (management by exception).
- **Acting faster**
  Finally, RFT needs a solution that is able to react at lightning speed. If they had to wait for some four weeks after the end of each quarter to gain relevant insights, then they would be flying blind for the four months before that. RFT might then have very deep insights into what lies behind them, but would at the same time know very little about their current business environment and even less about what they will meet next.

4.4.3 Building Blocks of the Solution

Which algorithms do you need?

In this section, we will provide you with a couple of ideas about which tools or algorithms could help you meet all these expectations. We are not going to restrict our considerations to components within SAP's product portfolio, but instead think outside the box.

Detecting Outliers

One of the key problems with detecting outliers is that the term *outlier* is itself not clearly defined. Whether a dataset is an outlier or not often lies in the eye of the beholder. Therefore, visualization is often the first tool of choice when checking data for outliers.

Algorithms

Using graphical data representation, there are four groups of algorithms that can help you detect outliers; we will take a closer look at each one of them in the following sections.

Visualizing Data

Detecting outliers in diagrams

When it comes to detecting outliers, the box-and-whisker plot is one of the most popular diagram types. A box-and-whisker plot (also called boxplot) visualizes how data are spread around their so-called median and helps you to identify outliers at a glance. There are also a couple of other display formats that—depending on the issue in question—can serve you well. One example is the *scatter plot*, which shows your data in the form of a two- or a three-dimensional cloud of dots.

[«]

Median

In terms of its intended information value, a *median* serves the same purpose as an average (a.k.a. an arithmetic mean), although it is less sensitive to outliers. If you sort a set of values in ascending order, the median is the numerical value separating the higher half from the lower half.

One example: the people in a group of three have annual incomes of $10,000, $20,000, and $10,000,000. In this case, their median income is $20,000 and their average income is $3,343,333. The one guy making ten million dollars a year would create a biased picture of how wealthy people within the group are.

As the median is less sensitive than the arithmetic mean, it is often considered a better measure of the data's location. With measurement errors, the median still delivers a reasonable idea of the data's mean as long as not more than 50% of the data are contaminated. It is therefore also called a resistant measure (or a resistant statistic) and plays a major role in *robust statistics* (a collective term for statistical methods used to deal with data that are not normally distributed or highly contaminated by outliers and measurement errors).

Define Distance Threshold

How far away is far away?

You may also want to treat all data that are far away (that is, further away than a certain distance threshold) from the rest of your data as outliers. The trouble with this approach is determining how you measure distance (how far is a dataset where the gender is male from one in which the gender is female?) and what kind of distance threshold you want to use.

Both decisions are in the end arbitrary and are based upon subjective experience; there is no (fool-proof) way of cross-checking these assessments using statistical algorithms. A common way of defining distance thresholds is calculating the so-called interquartile range (IQR).

Testing Data for Outliers

Detecting outliers via statistical tests

There are a whole slew of statistical tests (most of them named after their inventors—for example, Baarda, Grubbs, Dixon, Hampel, Nalimov, Pope, and Walsh) with which you can check whether a certain dataset is (probably) an outlier.

All the tests in our list (with Walsh's outlier tests being the only exception) are based upon the fairly restrictive assumption that your data are normally distributed; the tests then check whether a certain data point in question would be likely to occur in a normally distributed population or not. Some of these tests are contained in R's `outliers` package (for example, `dixon.test` and `grubbs.test`); for the other ones you will find instructions and formulas on the Internet.

[»]

Probability distribution and random variables

Probability Distribution/Random Variable

A *distribution* or more precisely a *probability distribution* comprehensively describes the behavior of a *random variable*—that is, a parameter whose values are (at least to a certain extent) unpredictable. A probability distribution assigns a probability to each value or set of values of a random variable.

Example: uniform distribution

With a six-sided die, you can roll six different numbers, and any of these numbers is equally likely—that is, each number has a probability of 1/6. As only six outcomes are possible (with a normal die, you cannot roll 2.5 or 3.3), the corresponding probability distribution is called a *discrete* distribution. In our special case (if the die isn't loaded, then each number is equally likely), we are dealing with a discrete *uniform distribution*.

If you had a die with which you could roll *each* number between 1 and 6 (including numbers like 1.76343789), you would be dealing with a *continuous* (uniform) distribution instead. In this case, each possible outcome would still have the same probability attached to it, but as there is an indefinite number of imaginable results (not just 6 of them) the probability for each outcome would be infinitely small (1.763437891 and 1.763437892 and every number between them would have equally likely outcomes). Hence, with continuous distribution one usually doesn't look at the probabilities of individual values but at the probability of an outcome lying within a certain range (for example, throwing a number between 1.5 and 1.6).

Normal or Gaussian distribution

The normal or Gaussian distribution (shown in Figure 4.12) is one of the most commonly used probability distributions; you may still remember it from school or university. With a normal distribution, extreme events are less likely than with a uniform distribution. Once you know its expected value and its standard deviation/variance, a Gaussian distribution is completely defined.

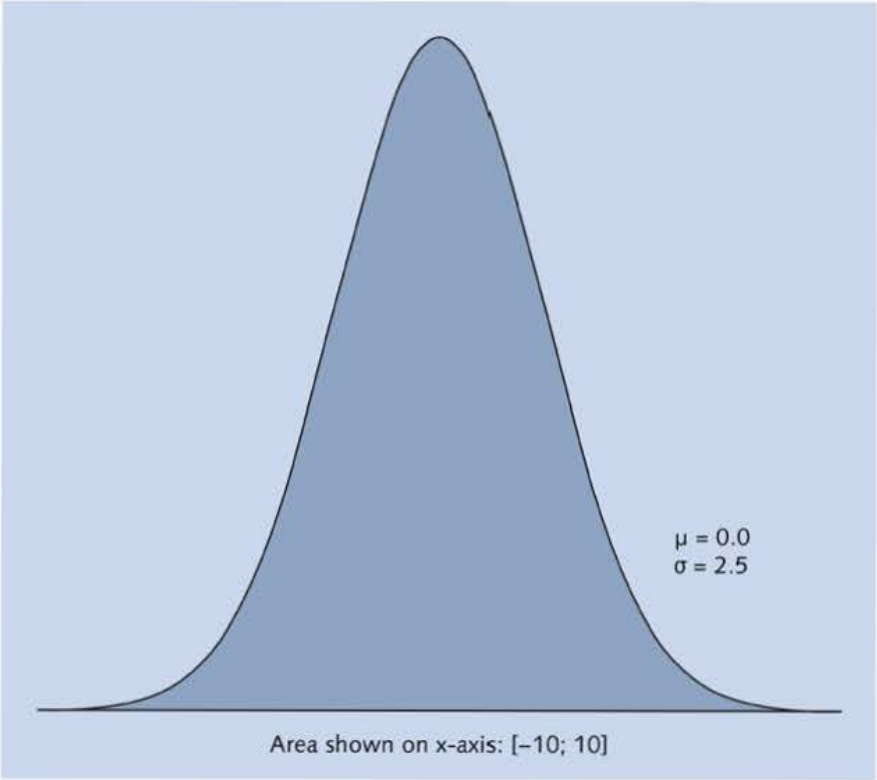


Figure 4.12 Normal Distribution

Expected Value

The *expected value* (sometimes incorrectly also called mean value) is the (expected) average value of a random variable having repeated the underlying stochastic process infinitely often. A bit more exact (in mathematical terms), the expected value is also called a distribution's first raw moment.

After an indefinite number of rolls of a nonloaded die showing (with equal probability) the numbers 1 to 6, one would (on average) roll $(1 + 2 + 3 + 4 + 5 + 6) / 6 = 21 / 6 = 3.5$. Therefore, 3.5 would be the expected value for the random variable "number of pips when rolling the dice." Note that the expected value—the arithmetic mean—can also be an impossible outcome with discrete distributions, such as when throwing a die.

Standard Deviation/Variance/Skewness

The *standard deviation* is a measure of how much a random variable fluctuates around its expected value; it can be considered something like the "expected value of the deviation from the expected value" and can thus also be defined as the square root of a distribution's *variance* (the distribution's second central moment).

In the case of our die, the variance would be $((1 - 3.5)2 + (2 - 3.5) 2 + (3 - 3.5) 2 + (4 - 3.5) 2 + (5 - 3.5) 2 + (6 - 3.5) 2) / 6 = 17.5 / 6 \approx 2.92$ and the standard deviation $\sqrt{2,92} \approx 1,71$. If the die was loaded, showing a tendency for small or large numbers, variance and standard deviation would be higher, and the distribution would be *skewed*.

Skewness is a measure of the asymmetry of a distribution-curve based probability. Take, for example a perfectly symmetrical bell-shaped distribution curve. If it is positively skewed by its data, it will lean to the left, and if negatively skewed it will lean to the right (a bit like the hats of the seven dwarves, but less extreme).

This might represent a value-creation business opportunity for you, particularly if your competitors are using tools that assume a perfect normal distribution (and many of them do). Change that assumption and you may gain insights to your own unique business advantage.

If expected value, variance, and standard deviation are not determined on the basis of theoretical considerations but instead (as usual) estimated on the basis of a sample, one often calls them (for the sake of clarity) sample arithmetic mean, sample variance, and sample standard deviation.

Using statistical tests

Statistical tests are used to check (reject) a hypothesis about a population using only a sample of the population's data. Based upon a sample of 100 randomly drawn visitors of a shopping mall, one could challenge the hypothesis that 80% of all visitors are female (in this case, the population one would make a statement about would be "all visitors of the mall"). If the hypothesis could not be rejected, management may decide to focus 80% of the mall's advertising budget on women or charge higher rents from shops selling ladies' wear than from those selling men's wear.

Statistical tests *never* deliver definitive conclusions, but can only be used to reject (accept) statements (hypotheses) with a certain probability (the so-called *statistical significance*). You could, for example, test the hypothesis that 80% of your mall's visitors are female using a statistical significance of 5%; this would mean that you are willing to accept a 5% chance that the results of your sample are a random anomaly, leading you to reject your hypothesis (80% female visitors) even though it is correct.

You need to know what you are doing

This statement is, however, only true if your sample is a piece of solid workmanship and really matches your hypothesis. If the data in your sample, for example, have been collected in a very narrow time slot, you may have instead tested the hypothesis that 80% of your visitors on Monday July 31, 2014, between noon and 2 p.m. were women. Because you actually wanted to say something about a different population (that is, all visitors of your shopping mall during the whole week and in the future), the sample is not representative, and your test result is worthless.

Even worse, many statistical tests rest upon assumptions regarding the population's probability distribution (they often assume that the population your sample is drawn from is normally distributed). If this assumption is not correct (one tends to forget to verify it), you will get distorted results, and your big data solution might lead you astray, giving you a head start in the wrong direction and generating competitive disadvantage.

Testing a Distribution Hypothesis

A slightly more sophisticated way of detecting outliers is checking whether not only individual data points but also all actuals you can get hold of are behaving as expected. No (serious) external consultancy would claim they are able to forecast exchange rates dead-on to four decimal places one year in advance; a consultant able to do that would no longer toil at his desk but would instead enjoy life as a multibillionaire on his own yacht or island (probably both).

Distributions as forecasts

Instead, a reputable consultant would put forward a hypothesis such as "the exchange rate in July 2012 will presumably be normally distributed and fluctuate around an expected value of μ with a standard deviation of σ" (also providing you with specific figures for μ and σ). Such a hypothesis contains three claims:

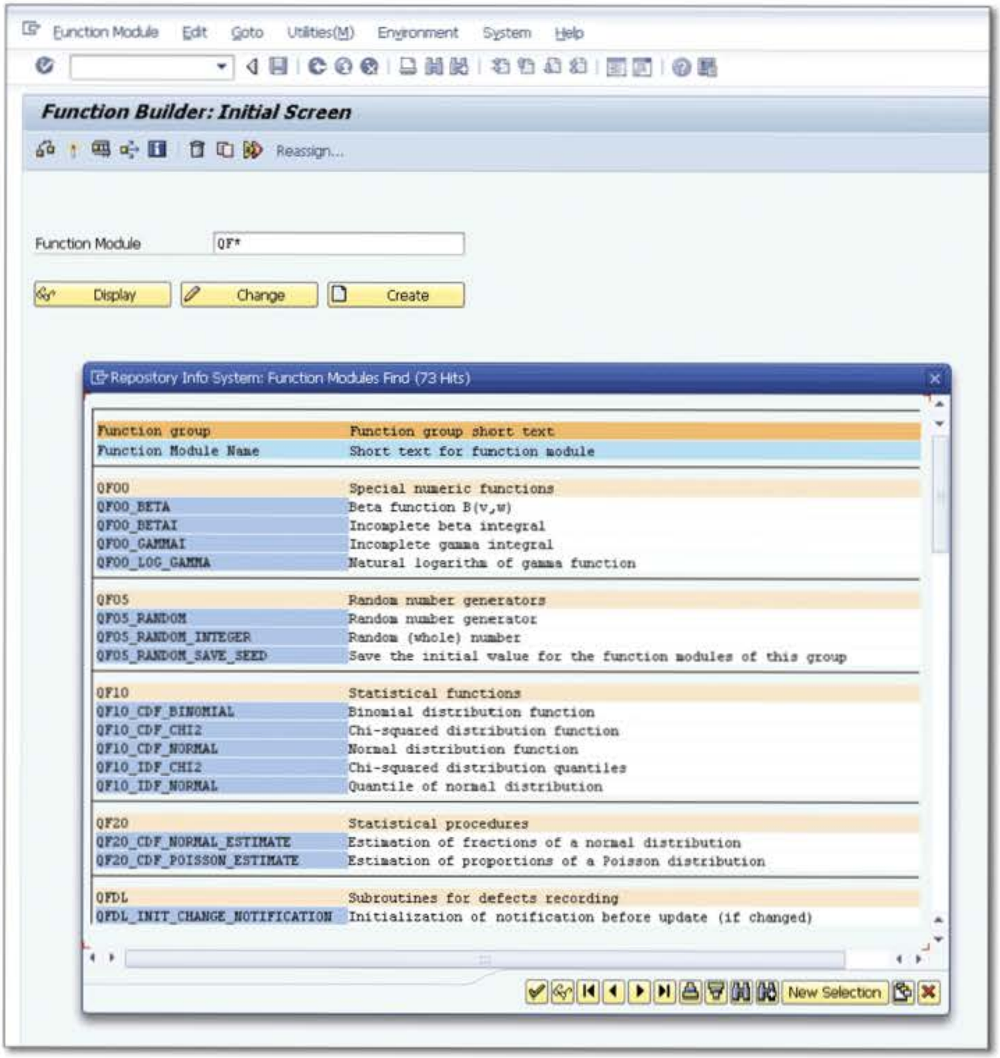
- Normally distributed
- Expected value is μ
- Standard deviation is σ

Any of these three claims can be challenged using statistical tests, or you can test all of them at once—for example, by using Pearson's chi-squared test for goodness of fit.

Statistical tests in software products

There are a wide variety of statistical tests that can be used to test hypotheses (like the preceding ones about exchange rates). When it comes to using or implementing such tests, you have several choices:

- Even commonplace spreadsheet software, such as Microsoft Excel, comes with a multitude of built-in statistical tests that can be used without additional programming (such as Excel's function `CHISQ.TEST` for the chi-squared test for goodness of fit).
- Furthermore, you could use good old ABAP to build statistical tests. SAP's traditional programming language comes with function modules that can be used for statistical testing (such as `QF10_IDF_NORMAL` to calculate the quantiles of a normal distribution or `QF10_IDF_CHI2` to calculate the quantiles of a chi-squared distribution; see Figure 4.13).
- SAP PAL also provides you with the tools for conducting a chi-squared test. The respective function there is called `CHISQTESTFIT`.
- As of the time of writing, the programming language R, which can be used within SAP HANA, comes with a total of six statistical tests (Shapiro-Wilk test, Anderson-Darling test, Cramer-von-Mises test, Lilliefors test, Pearson's Chi-squared test, and Shapiro-Francia test) used to challenge distribution-related hypotheses in R's standard set of functions (for example, `shapiro.test`) or via add-on packages.



Function Module QF*

Function group	Function group short text
Function Module Name	Short text for function module
QF00	Special numeric functions
QF00_BETA	Beta function B(v,w)
QF00_BETAI	Incomplete beta integral
QF00_GAMMAI	Incomplete gamma integral
QF00_LOG_GAMMA	Natural logarithm of gamma function
QF05	Random number generators
QF05_RANDOM	Random number generator
QF05_RANDOM_INTEGER	Random (whole) number
QF05_RANDOM_SAVE_SEED	Save the initial value for the function modules of this group
QF10	Statistical functions
QF10_CDF_BINOMIAL	Binomial distribution function
QF10_CDF_CHI2	Chi-squared distribution function
QF10_CDF_NORMAL	Normal distribution function
QF10_IDF_CHI2	Chi-squared distribution quantiles
QF10_IDF_NORMAL	Quantile of normal distribution
QF20	Statistical procedures
QF20_CDF_NORMAL_ESTIMATE	Estimation of fractions of a normal distribution
QF20_CDF_POISSON_ESTIMATE	Estimation of proportions of a Poisson distribution
QFDL	Subroutines for defects recording
QFDL_INIT_CHANGE_NOTIFICATION	Initialization of notification before update (if changed)

Figure 4.13 Function Modules for Calculating Quantiles in ABAP

Monitoring Forecasts and Models Separately

Is it a forecast or a model?

The trick with telling forecasting errors from modeling errors is to properly separate forecasts from models. Whether a fact is a forecast or a model is not set in stone; it depends on your perspective:

- From RFT's perspective, future exchange rates are a forecast; the company has no idea how these data have come about.

- Planned revenues instead are—from RFT's point of view—the output data of a deterministic model; RFT itself has created this model (in SAP BPC), which uses the sales forecast, the exchange rate forecast, and the sales prices as input data. Within this list, sales prices are not a forecast but something that is known for sure (these are fixed before the start of the year).
- For RFT, sales figures are the output data of a statistical (or stochastic) model. Although RFT knows nothing about the formulas and algorithms within that model, the company still knows which input data have been used.

Conditional and unconditional facts

From RFT's perspective, forecasts deliver unconditional statements about the future. Such statements will later prove to be right or wrong. In contrast, models come up with *conditional* statements: if input data are like this, then output data will be like that.

The only thing you can do with a forecast is monitor whether deviations between forecasted and actual values (such as actual exchange rates) can be considered normal random fluctuations or whether they are so high that the forecast itself should be questioned. This can be done by statistical tests.

Target values needed to check models

When testing a model, you need to proceed in a slightly different way. Using the actuals that have become available in the meantime as input data, you need to recalculate. Or to put it another way, you need to find out which output data (target values) your model would have delivered had it known the actuals for its input data (for example, actual exchange rates). These new target output data (for example, target revenues) can then be compared to the corresponding actual output data (for example, actual revenues). At this stage, you once again need to ask yourself whether the delta could be a random deviation, and you could once again do this using statistical tests.

Data models should be based upon algorithms

It would, however, be pretty time-consuming and exhausting to execute these tests manually for all your forecasts and models, but if you were slick when designing your data model you wouldn't have to. Instead, you could build the data to be checked (plan versus actual or target versus actual) in such a way that the respective statistical algorithms can process hundreds or thousands of dataset pairs, in the end giving you a list of suspected abnormalities to check.

To get there, it is important *not* to structure your data by content but rather to set up *domains* (the areas covered by your data warehouse) within your data model on the basis of the algorithms that can be applied to data within these domains. This means you would want to do the following:

- Separate data on a nominal-type scale from data on a ratio-type scale
- Separate data that are (presumably) normally distributed from data that are uniformly distributed
- Separate data from stationary stochastic processes from data generated by nonstationary stochastic processes

[«]

Level of Measurement

The *level of measurement* (sometimes called *scale of measure*) of a parameter defines (among other things) which mathematical operations can be executed on these parameters.

The random variable "gender of visitor in shopping mall" can assume the two characteristic values "male" or "female"; for these values, you can neither calculate an arithmetic mean nor a median: an arithmetic mean cannot be calculated because the two values cannot be added, and a median cannot be determined because the observations cannot be sorted from lower to higher values. The random variable delivers values on a nominal-type scale.

If, on the other hand, a random variable can assume values such as "good," "medium," and "bad," then you still can't come up with an arithmetic mean ("good" and "medium" cannot be added), but you are at least able to determine the median (because the values can be brought into a rank order: "bad," "medium," "good"). In this case, we are dealing with an ordinal-type scale.

The income of people visiting the shopping mall would be an example for data on a ratio scale. For such data, you can execute many more mathematical operations; you can also use them as input for most statistical algorithms.

Interval versus ratio scale type

Apart from nominal, ordinal, and ratio scales, there is also the so-called *interval-type scale*; unlike a ratio scale, an interval scale doesn't have a natural zero point, and because its zero point is defined arbitrarily distances between values can be calculated but ratios cannot.

For example, 100 degrees Celsius is 50 degrees warmer than 50 degrees Celsius but not *twice as warm*, and 0 degrees Celsius is not a natural zero point but an arbitrary one. However, an income of $10,000 *is* twice as high as one of $5,000 (as there is a natural zero point: $0 or no income at all).

[»]

Stationarity

Simply speaking, a random variable is stationary if its underlying (stochastic) process is independent of time and space. In terms of statistical calculations, "time" is often the key factor in terms of stationarity.

If the gender structure of people visiting your shopping center in the morning were fundamentally different from those coming there in the afternoon, a sample that is a mixture of observations in the morning and in the afternoon would be comparing apples and oranges. We would therefore merge two (stochastic) processes that obviously underlie different principles. Each one of them individually might be stationary, but the combination of both isn't (which, in return, means that calculating an arithmetic mean for each of them makes more sense than calculating one for both of them together).

Whether time plays a role for whatever you are trying to find out is among other things a question of *granularity* (the resolution with which you are looking at things). If you only need to make a statement about the day as a whole, then you can blend data from the morning and the afternoon. But if you need separate statements for both parts of the day (because you want to run different commercials at 10 a.m. and at 3 p.m.), then combining data from different times of day would be a serious mistake (a mistake that would probably also have financial effects).

Column-oriented storage for processing-oriented data models

A column-oriented database system is ideal for mapping data models that were built around operations and algorithms that are permitted from a mathematical perspective rather than based around content- or business-related (and therefore often transient) grouping principles. Imagine a company processing sensor data from production. Technical progress will lead to more types of sensors (more columns) being added over time, but no matter what a new sensor is going to measure, its data will still reside on a nominal-, ordinal-, interval-, or ratio-type scale. Once properly categorized, these data can be processed in exactly the same way as all other data so far.

Analyzing Root Causes within Your Models

Looking for faults in your models

The approaches discussed in the "Detecting Outliers" section of Section 4.4.3 will tell you whether a model as a whole is working or not. Ultimately, however, you probably want more: in an ideal world you want to know which parts of your model might be faulty or aren't a proper representation of reality.

Horizontal and vertical atomization

To accomplish that, you need to thoroughly divide your (data) model both vertically (by processing steps) and horizontally (by input parameters or mathematical properties of input parameters—such as their level of measurement). What might in the past have been a black box of highly complex Excel worksheets will now become a matrix consisting of very small, atomic data packages. Between these data packages, you will find clearly defined, manageable, step-by-step transformations; a structure like that will set the stage for mechanisms that can automatically and precisely localize problems within your models. In addition, flexible, modular, process-oriented data models and SAP HANA's layered structure (we will take a closer look at some technical peculiarities in the next chapter) complemented by declarative languages such as RDL are a perfect couple.

Vertical Categorization (by Processing Steps)

Splitting (vertically) by processing steps

What exactly do we mean by saying you should "divide your (data) model vertically by processing steps"? The clearer the route between input and output layers is structured (and all too often there are many layers), and the more granular your processing steps are, the easier it becomes to spot the source of a problem. Knowing that one out of 5,000 lines of source code is causing an issue doesn't help you a lot when sorting things out. Your developers might spend weeks or months in Transaction SE38, incrementally debugging (often undocumented) source code.

There's wrong, and then there's wrong understanding

Finding out what's wrong is also aggravated by the fact that "wrong" might not mean that a program is crashing or an error message is popping up. In a business context, "wrong" more often than not means that output data don't match the user's expectations. Such a mismatch, however, might not indicate a bug but could result from people not understanding a solution's functionality from end to end. If you are working in an environment in which the responsibility for business processes (and therefore also for applications supporting these processes) stretches across a number of departments located on various continents, getting an overall view might take ages (or even prove impossible).

Thanks to outsourcing and offshoring, most users employed by major organizations have to contend with exactly that kind of scenario, and the giant, monolithic solutions they are working with are now in the same awkward position the dinosaurs found themselves in some 65 million years ago: living fossils in an environment that they were never made for.

Fortunately, SAP has opened a door to evolutionary adaption, but quite a few companies will still take the route the millers in our introductory story have chosen: they will pray for water instead of just innovating and evolving. Make sure your company isn't one of them.

Horizontal Categorization (by Input Parameters or Domains)

Splitting (vertically) by (groups of) input parameters

As well as cutting your (data) models into vertical layers by processing steps, you should also slice them horizontally by input parameters and/or domains. There are two reasons for this:

- In the "Monitoring Forecasts and Models Separately" section, we pointed out that (input) parameters should be grouped by mathematical properties (such as their scale levels). In doing so, you will create domains on which the same statistical algorithms can be applied. Should you find a problem with any of these algorithms, modifying it will fix the issue for all data processed by it.
- To check your model for errors, you first need to generate target values by using actual instead of forecasted numbers as your model's input parameters. Once your model has produced target values on that basis, you need to "compare" these target values to what really happened. (We put the word "compare" in quotation marks because we would expect you to do a bit more than just view these numbers side by side; as mentioned before, you would also apply statistical algorithms to them.)

  If (or when) you find significant differences, you need to isolate their source, which is why you also want to further break down your input parameters into, for example, quantities and exchange rates. Separating sales quantities from exchange rates makes it a lot easier to a) calculate target revenues using actual sales volumes with forecasted exchange rates and to b) calculate target revenues using forecasted sales volumes with actual exchange rates.

Such an approach will help you separate the effects of errors in your sales forecasts from the effect of errors with forecasting exchange rates. In the course of such an analysis, you may also learn that problems related to the forecasting performance of one of your algorithms are only affecting some of your input parameters.

Dependencies make the analysis more difficult

Have we already managed to confuse you? Not quite? Then you are going to love the revelation that in practice identifying issues within your model is even more complicated. The reason is that there can be (and there usually are) dependencies among different input parameters (such as exchange rates rising whenever the GDP goes up) and dependencies among the effects of input parameters on output parameters.

Some are more (or less) equal than others

For example, a higher GDP might lead to higher sales volumes, but if income inequality exceeds a certain threshold, then a higher GDP could suddenly cause sales volumes to decline. Why? Because if a lot of people are earning a good wage, then they can all afford to eat and buy things they enjoy; however, if only (say) 5% of the population is taking home 95% of the total earnings no matter how much they spend, they will tend to put a higher percentage of their income into savings. They are unlikely to spend *in total* as much as the masses would spend with a more equal spread of income.

Input parameters can influence each other, but they also influence the output parameters of your model in funny ways. Two plus two does not always equal four when random variables are involved.

Imagine that you decide to use your revenue model to calculate the following revenue-related deviations among the revenues forecasted by your model, using different input parameters for sales volume:

Sales volume–related revenue deviation (a): actual revenues – (target revenues forecasted by your model using actual sales volumes and forecasted exchange rates as an input)

Exchange rate–related revenue deviation (b): actual revenues – (target revenues forecasted by your model using forecasted sales volumes and actual exchange rates)

Total revenue deviation (c): actual revenues – forecasted revenues

Intuitively, many people would assume that $c = a + b$; unfortunately, this doesn't always have to be the case. All of this gets a bit complicated, so if you would like a more detailed explanation, like us on our book's Facebook page (*https://www.facebook.com/saphana.makingthecase*) and ask us a question; we will get back to you!

Principal component analysis

In the meantime, don't worry: there are statistical tools to deal with all of this. For example, you can look at your input parameter's covariance matrix calculated using SAP PAL's `MULTIVARSTAT` or R's `cov`, `cor`, or `cov2cor` functions or by a procedure called principal component analysis (PCA). Both covariance matrixes and PCA are still a long way away from the limits of what can be done with languages like R, which means that from the perspective of some of your veteran ABAP programmers you may now be entering uncharted waters, making those guys feel as comfortable as Frodo on his way to Mordor when entering the dark tunnels of Cirith Ungol (the "spider pass"). By now, you may understand why we said that in addition to the right kind of tools (such as SAP PAL or R) you might also need some well-trained artisans (that is, mathematicians and data scientists).

Evaluating Risks/Analyzing Impacts

Compared to the contorted maneuvers described in the previous section, estimating the effects of forecasting errors is a piece of cake, possibly even a walk in the park (you can choose). All you need for this is unlimited computing power.

Monte Carlo simulation and value at risk

If you knew the potential tolerance of all your input parameters (to be a bit more exact, if you knew their probability distributions), you would just need to derive the resulting distribution of your output parameters. One way of doing that is a so-called Monte Carlo simulation; just use your model to calculate revenues on the basis of a large number of potential values for sales quantities and exchange rates (all of them randomly generated and all of them adhering to the proper probability distributions). The resulting output values of your model can then be used in a number of ways. Two of them are as follows:

- You can calculate certain risk measures (like variance or value at risk [*VaR*]) based upon your model's output parameters.
- You can try to identify the probability distributions of your model's output parameters.

Considerations like this are common practice in corporate finance or financial services and can be extended to all kinds of plans. In the past, the circle of those who used them remained relatively exclusive; one needed an amount of computing power that only well-heeled and hardware-mighty banks could afford. SAP HANA, however, makes it a lot more affordable.

Note that you cannot estimate the effects of modeling errors. To do this, you would have to know what the correct model looks like. But if you knew that you would probably use it anyway, reducing the effect of modeling errors to zero.

Trip wire: assumption of normal distribution

Under no circumstances, however, should you give in to the temptation to overlook one trip wire that still remains a hazard in the brave new world of big data. Always ask yourself whether your assumptions regarding the volatility and tolerance (that is, the probability distribution) of your input parameters are correct. In this context, most organizations still rely on the normal distribution first defined by the German mathematician Gauss in 1809, although this model dramatically underestimates the likelihood of extreme events (that is, events on the distribution curve's tails). Indeed, thinking that a set of input parameters are independent of each other without first checking that they truly are is naïve, and can lead you to false conclusions.

Cauchy Lorentz and Student's t

The financial crisis (interestingly, occurring more or less exactly at the 200th anniversary of Gauss' publication) and the impairment of securitized mortgage loans have provided us with a lot of illustrative material to support this; nevertheless, the normal distribution is still haunting practically all systems and algorithms used to analyze risk, even though there are much more suitable alternatives like the Cauchy Lorentz or Student's t distribution (of which a detailed discussion is beyond the scope of this book).

Being Constantly Vigilant/Acting Faster

Architecture and computing power

When proposing constant vigilance and real-time alerts, we are addressing the speed as well as the availability of big data solutions. You want your algorithms to check the validity of your forecasts and models 500 times per hour and in the twinkling of an eye, and you want to know they are doing so day and night, 24 hours a day, 365/366 days a year.

With distributed systems, to achieve this you need to accomplish the following:

- Properly size your solution
- Calculate the required redundancy of critical components (based upon their probability of failure)
- Think about architectural models/products that are designed for real-time processing

To do this, you may have to understand the principles of the lambda architecture and consider implementing SAP Event Stream Processor (see Chapter 2, Section 2.1.3).

[+]

Big Data and Cloud Computing

With big data, appetite comes with eating, which is a recipe for trouble when it comes to sizing your hardware. Furthermore, how are you going to know how much computing power those algorithms that have not yet been developed are going to swallow up? We are, therefore, tempted to bet on the fact that cloud-based, adaptable solutions will become more popular in the long run.

When it comes to cloud computing, there are, however, security concerns. In the light of certain revelations regarding what secret services all over the world are up to (and considering the fact that the US government can obtain warrants to force US hosting providers to hand over even data that are stored abroad), such concerns should not be easily dismissed.

The problem can, however, be mitigated a bit by anonymizing data, as you would do with test data. Anonymization algorithms are also indispensable in crowdsourcing, when you need to be even more careful; unfortunately, de-anonymization and reidentification are classic application areas for big data data-mining algorithms.

4.4.4 Potential Benefits and Value Drivers

At this stage, you may want to briefly review the benefit–value driver matrix in Chapter 1, Section 1.5. For the first time in the book we will apply the more general ideas within that matrix to a specific case study, the scenario about flexible planning/monitoring plans and forecasts.

Potential Benefits

Existing and new business processes

Our case study deals with both existing and new business processes:

- **Potential benefits in existing business processes**
  The planning process as such is not at all new for RFT. Therefore, some of the improvements RunFlat is hoping for with SAP HANA are meant to pop up with existing and well-established activities. The key innovations in RFT's planning process lie in separating forecasts and models, dividing data and processing steps into very small, digestible portions, and using mathematical and statistical criteria instead of semantic, content-related categories for structuring the data model. All of these measures affect the how of an existing business process but don't create a new one, which is why all related value drivers would be found on the left-hand side of the benefit-value driver matrix. In regards to our approach to data modeling in general, also see Section 4.5.2.

- **Potential benefits in new business processes**
  What many organizations (apart from investment banks and the like) don't yet have are fully automated processes that employ statistical algorithms that monitor forecasts and plans in real time, detecting outliers, evaluating impacts, and alerting responsible employees on a case-by-case basis.

  The continuous search for outliers and the more or less permanent execution of impact analyses leads to new processes. These processes might have interfaces that connect them to existing ones, but they are independent of these existing processes. They also make very different demands on users and are implemented on different kinds of platforms, which is why related value drivers would be found on the right-hand side of the benefit–value driver matrix.

How can RFT create value?

In terms of the benefit–value driver matrix's vertical (how) dimension, the approaches described in Section 4.4.3 touch all four rows of the matrix (see also Figure 4.14):

- **New insights**
  The detection of outliers leads to new insights about forecasts and models. Previously, RFT couldn't distinguish between random and harmless deviations and those indicating serious errors in forecasting, modeling, and planning. Now, they can.

  Another thing new to RFT is the concept of receiving tips about critical areas within their models if significant deviations arise. These critical areas can now be localized vertically (by processing step) and horizontally (by domain).

  Going further, it would even be possible to build a role-based workflow solution that can automatically inform the affected users/user groups about detected problems and monitor whether these problems are solved or not. Offshoring and outsourcing would become a lot easier to manage with such workflows in place.

- **Better decisions**
  In Section 4.2, it became apparent that RFT's production plan serves as the basis for many other important decisions; errors in this plan will result in expensive mistakes in production and other areas.

  The more precisely (and the earlier) planning errors are discovered, the higher the probability that better follow-up decisions will be made. In an ideal case, these follow-up decisions are not only better than the ones that would have been made without the new solution but also better than those RFT's competitors are making.

- **Sophisticated tools**
  In Section 4.4.3, we suggested quite a few mathematical and statistical algorithms that RFT could use for, among other things, detecting outliers. This list is a far cry from being complete, and the only factors limiting the complexity of the algorithms you may want to use are your team's creativity and knowledge and the computing power of your systems. The further RFT is able to go with commercially sensible effort, the further they will stay ahead of the pack; SAP HANA helps them, because it excels at reducing the costs of building competitive advantages via sophisticated algorithms.

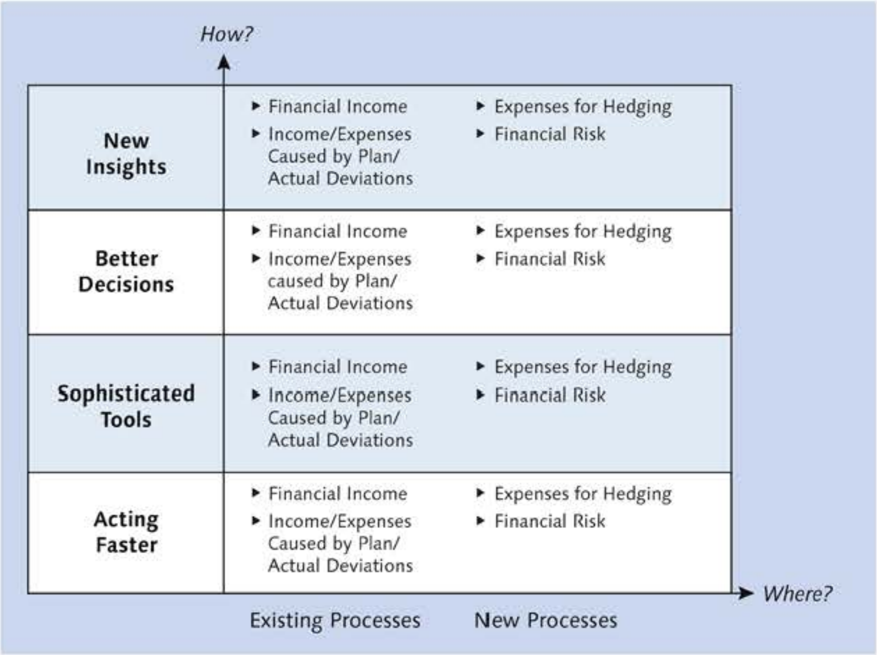
- **Acting faster**

  If RFT finds out earlier that reality does not abide by its beautiful forecasts and plans and that such deviations are not random but systematic, then it can take countermeasures more quickly. It can also adapt forecasts and models (plus the plans and decisions based upon them) faster than its competition.

  RFT gains time via real-time alerts (that is, the new business process), identifying weaknesses in its planning models is quicker and simpler, and much less time elapses between a change in its environment and the subsequent change to its models that reflect the environment.

Value Drivers

Figure 4.14 shows the benefit–value driver matrix for our planning scenario.



	Existing Processes	New Processes
New Insights	▸ Financial Income ▸ Income/Expenses Caused by Plan/ Actual Deviations	▸ Expenses for Hedging ▸ Financial Risk
Better Decisions	▸ Financial Income ▸ Income/Expenses caused by Plan/ Actual Deviations	▸ Expenses for Hedging ▸ Financial Risk
Sophisticated Tools	▸ Financial Income ▸ Income/Expenses Caused by Plan/ Actual Deviations	▸ Expenses for Hedging ▸ Financial Risk
Acting Faster	▸ Financial Income ▸ Income/Expenses Caused by Plan/ Actual Deviations	▸ Expenses for Hedging ▸ Financial Risk

Figure 4.14 Flexible Planning Benefit–Value Driver Matrix

Value drivers with RFT

We inserted some sample value drivers for RFT into the cells of the matrix. In principle, the following four value drivers might create value for both existing and new processes, but not only is RunFlat's new real-time alerting process likely to reduce the costs for ad hoc hedging and the total overall risk exposure of the company, but also improvements to their existing planning process (such as the option to adapt models more quickly) will make it a lot easier to deal with plan/actual deviations.

- **Exchange rates gains/losses**
  With the help of an early warning system, RFT can minimize (or even avoid) imminent exchange rate losses or take advantage of opportunities for gains. In the short term, this might mean that RunFlat—noticing a higher than expected volatility of, for example, the Polish Zloty—might want to hedge a higher share of its exposure via options or futures. Midterm or longer term, RFT may want to move production capacities from one country to another, thus getting to the crux of the problem instead of just curing symptoms.
- **Gains/losses due to plan/actual deviations**
  In general, better forecasts will help RFT make better decisions, and, when seeing things through the eyes of a conservative accountant, better decisions will at least help avoid mistakes and resulting costs. This is, however, only one side of the coin; better decisions are not just about avoiding losses but also about maximizing opportunities. Higher exchange rates can not only result in higher production costs but also lead to higher revenues in RFT's company or group currencies (CHF or USD). Fewer surprises in forecasting and planning can be beneficial in two ways, leading to lower expenditures and higher returns.

  Keep in mind that you need to differentiate between plan-to-actual and target-to-actual deviations. With forecasts, you have no models and can therefore only calculate plan-to-actual deviations. With output values of your models (planned values) you have both. Furthermore, if there is a difference between planned and target values with the output of a model, you should not blame the model but the forecast that fed the model with data. From the model's perspective, input data (forecasted or actual values) are set in stone and not to be brought into question. The model's job is to provide output data that make sense based on the data it received.

- **Hedging costs**
  Hedging risks by, for example, buying derivatives is only the second-best solution for RunFlat. Reducing the risk exposure by having better foresight in the first place would be a much better solution. Hedging costs money and becomes more expensive once your counterparty smells a rat (and your bank—often the counterparty for hedging transactions—is usually pretty good at rat detection). If you are flexible in terms of moving production quantities around among various plants and if you are getting better in terms of forecasting and planning, then you don't have to eliminate the risks resulting from each and every exchange rate fluctuation by buying call or put options for your cash flows in foreign currency. Instead, you only need to attend to the residual risk (emerging during the time it takes to change the manufacturing plant for the Super X7700); this will help you reduce your hedging costs by several orders of magnitude.
- **Overall risk**
  In Chapter 1, Section 1.4.3, we claimed that shareholders, returns being equal, will usually prefer lower-risk investment options. Hence, if RunFlat is able to plan more precisely, then the gap between the company's announcements and their published results is going to diminish. All other things being equal, shareholders should honor this improvement with higher share prices (and thus higher shareholder value).

4.5 Implementation Scenario and Architecture with SAP HANA

More than one scenario

Some say that all roads lead to Rome. Looking at RFT's requirements, not all roads, but maybe more than one, might take you to your destination. Some of them are long and winding; others may not be ideal for flip-flops but might not require climbing boots.

4.5.1 Implementation Scenario and Framework Architecture

Possible implementation scenarios

From our perspective, satisfying the demands set in Section 4.4.2 by using some of the algorithms mentioned in Section 4.4.3 could happen via three different implementation scenarios (see Chapter 2, Section 2.2):

- **App scenario**
  Considering what RFT is planning to do, their most obvious choice would be the app scenario. The reasons are as follows:
  - For the requirements, RFT only needs read access to relevant data; therefore, the range of applicable scenarios is narrowed down to three replication scenarios: the data mart, the app, and the content scenarios. (The SAP Business One analysis scenario, although also a read-only one, isn't a candidate, because RFT is not using SAP Business One.)
  - There is no standard content for the requirements, which eliminates the content scenario.
  - Standard products for data exploitation (be they Microsoft Excel or SAP BusinessObjects BI) will probably not satisfy RunFlat's very specific demands in terms of statistical algorithms. It will need tailored apps that are able to use SAP PAL and R. This rules out the data mart scenario.

  As defined in Section 4.2, RFT's (forecasted and planned) data are already held in an SAP HANA database sitting under its SAP BPC and SAP BW applications. Furthermore, actual data can be replicated in real time (to ensure timely alerting with outliers) from SAP products or third-party applications via the interfaces discussed in Chapter 2, Section 2.1.3.

  Note that content from SAP solutions (and a few others) doesn't necessarily have to be replicated; virtual tables might be used instead. SAP defines *virtual tables* as tables pointing to remote tables in other databases; they enable real-time access to data regardless of their location and will not affect the SAP HANA database.

- **Cloud on SAP HANA scenario (with limitations)**
  Theoretically, it would also be possible to set up all required apps within a cloud, but the purpose of the algorithms described in Section 4.4.3 is to quickly process large amounts of data (from SAP BPC or the underlying SAP HANA database). With the cloud on SAP HANA scenario, however, the SAP HANA database itself might not live in the cloud; this would raise some questions about data-transfer rates between the database and the cloud.

If all of the required analyses are not already taking place on the database layer (for example, in the form of SQLScript procedures with embedded R), then the cloud on SAP HANA scenario only makes limited sense. Furthermore, if the rubber meets the road at the database level, one may ask what else apps in the cloud are supposed to do. Things would only look different if other data from the Internet (such as Twitter streams) and in-house data needs to be analyzed. Such data could then be processed and prepared by a cloud-based (standard) service (such as Klout).

- **SAP BW on SAP HANA scenario (with limitations)**
  RFT is already working with this scenario, but SAP BW alone isn't the ideal application for exploiting data via algorithms like the ones mentioned in Section 4.4.3.

Possible products with app scenario

Having chosen the app scenario, we can now come up with proposals for which software products RunFlat might want to use (see Figure 4.15).

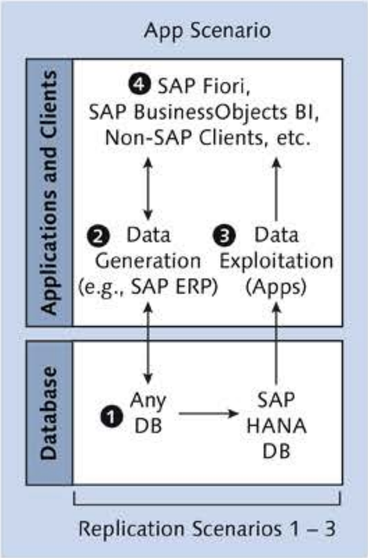

App Scenario
Applications and Clients
❹ SAP Fiori, SAP BusinessObjects BI, Non-SAP Clients, etc.
❷ Data Generation (e.g., SAP ERP)
❸ Data Exploitation (Apps)
Database
❶ Any DB
SAP HANA DB
Replication Scenarios 1 – 3

Figure 4.15 Flexible Planning Implementation Scenario

Databases ❶

Processing event streams

In our scenario, plan data (in SAP BPC/BW) plus some actual data (in SAP BW or SAP ERP) may already be in SAP HANA (in Section 4.2, we mentioned that RunFlat implemented SAP HANA, but we didn't explicitly state whether or not it already migrated SAP BW and/or SAP ERP onto an SAP HANA database). When talking about databases, we therefore primarily mean other data outside of SAP (such as real-time exchange rates or macroeconomic data underlying RFT's sales forecast), so RunFlat only has to replicate these data.

In some cases, such data may sit in classic relational databases, but more often than not, we are probably talking about event streams. Therefore, instead of classic solutions for data logistics, RFT will presumably use the input adapters of SAP Event Stream Processor (ESP) in connection with SAP ESP's output adapter for SAP HANA.

It could be that the delivering application doesn't store these data itself, in which case we may want to cater for a corporate memory layer within SAP HANA. Such a corporate memory layer will make reports/analyses reproducible and repeatable and provide you with a proper audit trail.

[»]

Corporate Memory Layer

In data warehousing, a *corporate memory* (CM; also called *institutional* or *organizational memory* [OM]) *layer* is an archive of all data brought into the data warehouse from delivering OLTP systems.

A corporate memory serves three purposes:

- It enables you to rebuild downstream layers without putting pressure on your delivering OLTP systems.
- It conserves longer-term snapshots of data that are subject to continuous change in your OLTP systems.
- It allows data in the downstream layers of your data warehouse to be tracked back to their source.

Products Generating Data ❷

The data that are relevant for analyses with RFT are predominantly generated by SAP BPC (forecasts and plans) and SAP Business Suite (actuals).

Products Exploiting Data ❸

SAP and non-SAP products

For the app itself (that is, the heart of the app scenario), there are quite a few options:

- In-house development using SQLScript, PAL, and R
- In-house development using RDL, PAL, and R
- In-house development using ABAP
- Standard products from other vendors (such as IBM SPSS Analytic Server, a competitive product to SAP PAL)
- A combination of all of the above with different apps/solutions providing individual (web) services

To serve its analytical applications, RFT may want to create a couple of application-specific data marts; ideally these data marts should be set up in SAP BW (which RFT is using anyway) and should adhere to the specifications of LSA++.

Clients ❹

The presentation layer

Given that RFT is already using SAP BusinessObjects BI (see Section 4.2), it will also tend to use these products for producing reports and graphical representations of its data. SAP BusinessObjects BI can deliver a wide variety of display formats, including items such as boxplots. Nevertheless, we have mentioned before that SAP BusinessObjects BI does have its limits (see Chapter 2, Section 2.1.2). SAP BusinessObjects BI can offer a lot in terms of diagrams but has never been great visualization software for statistical data. Hence, it may make some sense to use a) the graphical functionalities within R and b) other software products, such as SAS, SPSS, or freely available solutions such as ViSta or the package XLispStat for the programming language XLISP.

Instead of installing a piece of software, you might also consider using the services of specialized providers on the Internet. This makes even more sense if diagrams are to be analyzed in the course of crowdsourcing projects. To connect non-SAP clients to your SAP HANA database, you would want to use the interfaces mentioned in Chapter 2, in the "(Meta) Data Integration with Non-SAP Products" section.

4.5.2 Data Architecture

SAP's architecture model for data warehousing

When developing data warehouses, data marts, or applications that are based upon SAP HANA, it might be a good idea to adhere to the principles defined by SAP for SAP HANA-based data warehouses. These principles are summed up in SAP's LSA++ architecture model. We will take a (brief) look at these guidelines before discussing a few more ambitious aspects of data modeling. Many of these aspects will also be explored further within other case studies.

LSA and LSA++

Transforming and aggregating data

Layered Scalable Architecture (LSA) is an architectural reference model originally developed for SAP BW. In this model, data flow from the bottom (data sources) to the top (data marts and reports) via a couple of horizontal layers stacked atop one another. The data sources feed very detailed, granular data to a data-acquisition layer (see Figure 4.16). From there, data are then passed on upwards (that is, toward reporting); in the course of this vertical data flow, they are also restructured and transformed, reflecting functional/business requirements. At the same time, data are aggregated. The reason for aggregating the data on their way upwards is that strategic reports often only need summarized data. If data are already summarized, then queries supplying reports can be run with much higher performance.

Domains and data marts

On the lowermost layers of a data warehouse, its domains are determined by the structure of data sources supplying data, whereas at the upper levels data are categorized by business (reporting) requirements. The data flows within a data warehouse therefore, in a sense, convert data from a source-oriented domain structure into a business-oriented one (a data mart).

Partitioning

To improve a data warehouse's load performance, data flows are often split (*partitioned*) on their way from the data acquisition to the data mart layers via functional criteria. A typical example would be dividing customer data by postal codes, processing those for postal codes between 00000 and 10000 separately from those within postal code areas between 10001 and 20000. The purpose of *logical partitioning* (as this splitting of data is also often called) lies in part in enabling parallel processing of data.

In addition to dividing your data by logical (content-related) criteria, many data warehouses and databases (including SAP BW and SAP HANA) also support *technical partitioning*. This means physically spreading data within one logical table across a number of physical tables. Similar to logical partitioning, technical or physical partitioning is meant to enable parallel processing and therefore improve query performance.

Data stored redundantly

In LSA, one accepts a certain level of redundancy. Redundancy (of granular and summarized data) can help improve performance but also makes data within reports reproducible and traceable. This also gets one beyond the issue that data in delivering (OLTP) systems can change over time (for example, the contents and status of a sales order) and so snapshots of those data need to be taken and stored at certain times or process-point intervals.

LSA++

Today, an SAP BW data warehouse might be sitting on top of a classic database and on SAP HANA. This is one reason why SAP has replaced its LSA reference architecture with its successor, LSA++. Unlike with LSA, data are only stored persistently in an LSA++ architecture if there is a business need to do so (such as traceability). Storing data persistently for performance reasons (for example, in the form of persistent key figures) is no longer necessary (although this might change again in the future because data volumes are growing exponentially).

Furthermore, logical partitioning is no longer necessary for performance reasons, because SAP HANA is a lot faster than classic databases and because it is a distributed system that can split work processes itself. In fact, we don't even *want* logical partitioning, because it leads to extra maintenance work.

[«]

Persistent Key Figures

In a data warehouse, key figures are normally calculated by queries or within reports, but to improve reporting performance such calculations could also be shifted into the data flow. The results of these calculations (also known as calculated key figures) are then stored persistently within the database and called *persistent key figures*.

Agile data marts

In LSA++, a centrally managed enterprise data warehouse (EDW) is supplemented by agile data marts and extensions using operational (detailed, nonaggregated) data. Such agile data marts are set up centrally or decentrally as and when required. Reports are then based on data in the EDW core, agile data marts, operational extensions, or a combination of all three sources.

[»]

Data marts and agile data marts

(Agile/Virtual) Data Mart

Classically, data marts are a copy of one or more subsets of data from one or more data warehouses or other databases. The purpose of data marts is to combine data from different sources to satisfy specific business requirements. To improve reporting performance, data within a data mart are often copied from the respective sources (instead of, for example, simply creating views) and then stored persistently and redundantly.

An agile data mart is a data mart created ad hoc, following the principles of agile BI (see Chapter 1, Section 1.1.2).

In SAP's new LSA++ reference model or more generally in SAP HANA-based environments, data marts can be either persistent or virtual. Because agile data marts tend to be used for temporary requirements, they are often virtual.

[»]

Composite/Virtual/Transient Provider, Analytic Index

Virtual data marts can be implemented using different object types:

- With SAP BW, you may want to use CompositeProviders (created via Transaction RSLIMOBW) plus VirtualProviders or TransientProviders based upon SAP HANA models (created via Transaction RSDD_HM_PUBLISH).
- From an SAP HANA perspective, data marts could be represented by calculation views; such calculation views may, in return, use objects from SAP BW.

A CompositeProvider consolidates data from InfoProviders and/or analytic indexes, applying a *union*, *inner join*, or (*left*) *outer join* operation. InfoProviders are (often persistent) data pools within SAP BW; analytic indexes contain data that are the result of calculations performed by an analysis process within SAP BW's Analysis Process Designer, a tool for data analysis and data mining.

For example, given two databases, A and B, these database operations (union, inner join, and left outer join) work as follows:

- A `UNION` (in SQL) returns the union set of A and B.
- An `INNER JOIN` (in SQL) returns the intersecting set of A and B.
- A `LEFT OUTER JOIN` (in SQL) returns all records from A plus data from matching records (records with the same key) in B. A `RIGHT OUTER JOIN` would give you all records in B plus matching data from A, and a `FULL OUTER JOIN` would return the combined result of left and right outer join. The difference between a full outer join and a union lies in the fact that a full outer join will return all columns of both tables, delivering `NULL` in fields that are not provided by the database delivering the record, whereas a union only delivers columns that are contained in both databases.

In principle, CompositeProviders serve the same purpose as InfoSets in SAP BW, but, unlike InfoSets, CompositeProviders are optimized for use on an SAP HANA database.

VirtualProviders are used to access data within analytic views and calculation views in SAP HANA from SAP BW without having to replicate these data. TransientProviders serve a similar purpose. The main difference between both objects is that VirtualProviders are often used if structures are needed for a certain (limited) time, whereas TransientProviders (as their name suggests) serve fairly short-term, ad hoc requirements.

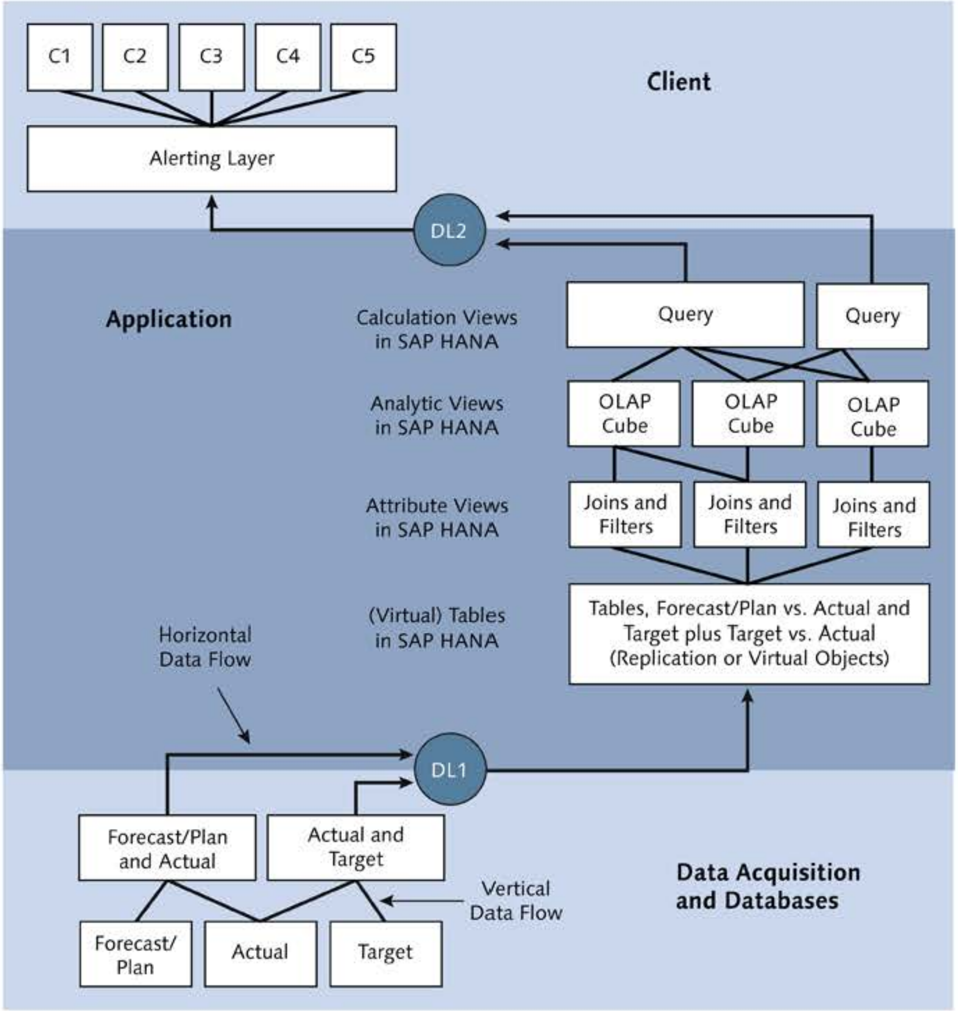


Figure 4.16 Flexible Planning Data Architecture

Analytic views and calculation views are (probably) the two most important concepts within SAP HANA. We mention them in Figure 4.16, and will get back to both in later chapters. Nevertheless, we believe it's a good idea to supply you with a basic definition right now.

[»]

Analytic and Calculation View

Analytic and calculation views can be defined as follows:

- An analytic view is an object that consists of dimensions and fact tables, modeled according to the star (data mart) schema (very much like an OLAP cube).
- Most calculation views are a combination of analytic views. The data of all analytic views involved are merged (using, for example, join operations), filtered, and/or processed—for example, via SQLScript or R.

More or fewer layers in LSA++?

As a rule of thumb, LSA++ leads to a reduction in the number of layers between data sources and data marts; due to SAP HANA's extraordinary performance, many reports that once needed preaggregated data can now use raw, granular datasets instead. This reduction in the number of layers only applies, however, if your LSA++ data model is serving exactly the same purpose as your previous LSA-based design. As you will see, some of the proposals in our case studies will breed *more* rather than *fewer* layers.

Functional views via semantic layers

In LSA++ (and in our data models), functional/business views on data are provided by respective (virtual) semantic layers. In our data model for this case study, a functional view on data is only needed at the top level (represented by "C1," "C2," and so on—"C" stands for client—in Figure 4.16). We have allowed for more than one client because each client might not only serve its own group of recipients but also address different hierarchical levels within the company or different channels of communication (SMS, tweet, and so on).

However, we needn't worry about business semantics until we get to the first or second layer underneath the clients; in our data model, this layer is called the *alerting layer* (because it is there to collect messages that will trigger alerts).

Further Considerations about Data Architecture

Characteristics of SAP HANA data architectures

Our thoughts about data modeling go a bit further than what is laid down in LSA++ (which primarily focuses on SAP BW). We recommend following five additional design principles:

- Process-oriented instead of content-oriented
- Configuration instead of programming
- Declarative/functional instead of imperative
- Virtual instead of persistent
- Business process–oriented instead of performance oriented

When discussing RFT's data architecture, we do not strive to provide you with a detailed data model down to the level of an individual database object. Instead, we would rather present a couple of general suggestions. Figure 4.16 features a high-level data architecture built on two key specifications from Section 4.4.3:

- Monitoring forecasts and models separately
- Analyzing root causes within your models

The lower area (DATA ACQUISITION AND DATABASES, two layers) constitutes RFT's data sources (SAP ERP, non-SAP ERP systems, and SAP or non-SAP data warehouses on any database systems, including SAP HANA). The middle section (APPLICATION, four layers) stands for the application (app) RFT wants to build in SAP HANA, delivering the required analytical functionalities. Finally, the upper segment (CLIENT, two layers) represents clients and/or user interfaces.

The naming of the three areas within Figure 4.16 (DATABASES, APPLICATION, and CLIENT) corresponds to the naming of layers within our implementation scenarios (see, for example, Figure 4.15).

The two circles labelled DL1 and DL2 at the upper and lower boundary of the middle area in Figure 4.16 stand for data logistics solutions (such as SAP Data Services). If data are already stored within SAP HANA, then no products for data management (data logistics or metadata) are required, at least as long as there isn't the need for extensive data cleansing or transformation.

Process-Oriented Instead of Content-Oriented

Problem: content-oriented data models

In many organizations, data inventories are structured by content (that is, by business areas or business functions). Accounting documents, cost-allocation records, material master records, routings, and customer orders all reside in their own separate domains and are kept apart from each other in terms of data management. Also, data that are written in the course of linked business transactions (for example, customer inquiries, customer quotations, and customer orders) are also linked in the database (for example, via SAP's so-called document flow), whereas data related to the same customer but to different transactions (for example, dunning) are again unconnected. Categorizations like this make perfect sense from a business perspective; nevertheless, they are still arbitrary.

Breaking down data in this way gives rise to three problems when shifting application logic to the database layer:

- Your departments keep reinventing the wheel for their respective operational areas. When it comes to big data, one and the same clustering algorithm can be valuable for both financial and logistics data, but if both data pools are living in different worlds, isolated and sealed off from each other (owned and administered by separate departments, projects, or teams) the chances are that—after a couple of years—you will encounter some kind of mishmash that demonstrates in a very creative way that there are a hundred different ways of solving exactly the same problem. Core algorithms are often the same; they are just packed into impermeable layers of design and user interfaces, making it impossible to see how alike they are in principle.
- Processing steps are tailored to the structure or the naming of certain contents, but such proprietary criteria are rarely reusable and very fragile when structural changes are made to their underlying data.
- Detecting new dependencies becomes almost impossible. If data from customer orders and purchase requisitions are located in their own domains (each of them following different design principles), then only a few exceptionally gifted genius developers will be able to come up with data-mining algorithms that can automatically unveil dependencies between customer orders and purchase requisitions sent to vendors (in the whole of Europe with its 750 million people, we know of only three to five people of this caliber).

Analogy: dimensions of an OLAP cube

Our demand for process orientation is inspired by SAP's guidelines for SAP BW data modeling. When assigning characteristics to an InfoCube's dimensions, technical instead of semantic considerations should be paramount. A well-modeled InfoCube (once filled with data) has dimension tables that are a) of similar size and b) relatively small (5–10%) compared to the size of its fact table (size in both cases is measured by number of records). As with OLAP modeling, we strongly advise you to not get distracted by content but instead to remain focused on the mathematical and statistical properties of your data and on the ways in which you intend to process and analyze these data.

Level of measurement, granularity, and further processing

Data with similar mathematical and statistical properties (such as level of measurement) or similar *granularity* (fineness, level of detail) and data that are meant to be processed via the same algorithms should be grouped within the same domain. Data that are alike in terms of their business meaning or the transactions they were generated by should not constitute domains. Both principles are asking for a certain level of abstraction but will—when properly implemented—lead to very lean and at the same time very flexible data models.

We are going to explain this concept via Figure 4.16:

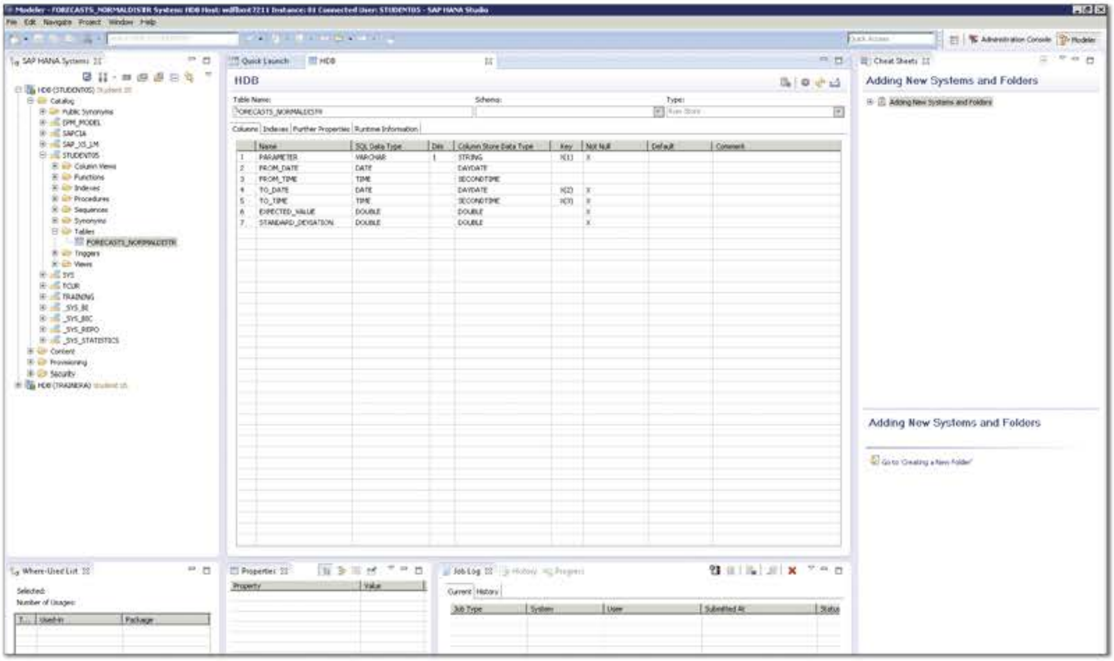
- With the source data mentioned at the bottom of Figure 4.16, forecasts are separated from actual and target values. One of many reasons for this is that quite often the granularity of forecasted data is a lot lower than that of actual data; hence it does not make much sense to store both via the same record structure, mixing them in your data model.
- Why is their granularity different? For forecasted exchange rates, we may only have the distribution parameters (μ and σ) mentioned in Section 4.4.3 (that is, two numbers per parameter); with actual exchange rates, we may instead have one number for every second of every day within each fiscal year.
- In the same way, we have separated raw data (boxes FORECASTPLAN, ACTUAL, and TARGET in Figure 4.16) from consolidated and adjusted ones (boxes FORECAST/PLAN AND ACTUAL and ACTUAL AND TARGET in Figure 4.16). In Figure 4.16, the layer called DATA ACQUISITION AND DATABASE only has two layers; in reality, you'll often have more than two.

- Separating raw from partially processed data boosts data model flexibility. If changes to the structure of raw data (such as using more complex distributions that need more than two parameters) become necessary, such changes don't have an impact on higher layers.

Whether the layers within DATA ACQUISITION AND DATABASE are virtual or persistent does not play a major role in an SAP HANA environment.

Describing normally distributed forecasting data

RFT assumes that sales quantities and exchange rates in a certain month and for a certain country are normally distributed; its external consulting partners provide the parameters needed to completely and definitely identify these distributions (μ and σ, one μ and one σ per combination of month and country). On this basis, RFT could describe forecasted values using a table like the one shown in Figure 4.17; such a table could be created in SAP HANA's administration workbench SAP HANA Studio, and it could not only contain distribution parameters for exchange rate forecasts but for all normally distributed parameters.



HDB

Table Name: ORECASTS_NORMALDISTR
Schema:
Type: Row Store

	Name	SQL Data Type	Dim	Column Store Data Type	Key	Not Null	Default	Comment
1	PARAMETER	VARCHAR	1	STRING	X(1)	X		
2	FROM_DATE	DATE		DAYDATE				
3	FROM_TIME	TIME		SECONDTIME				
4	TO_DATE	DATE		DAYDATE	X(2)	X		
5	TO_TIME	TIME		SECONDTIME	X(3)	X		
6	EXPECTED_VALUE	DOUBLE		DOUBLE		X		
7	STANDARD_DEVIATION	DOUBLE		DOUBLE		X		

Figure 4.17 Table for Sales and Exchange Rate Forecasts

Implementation as an SAP HANA table

All RFT would have to do to create such a table is define the names of required fields plus their SQL data types in SAP HANA Studio. Additional information (such as the column store data type needed for columnar data storage) is automatically created when generating the table. Whether the table will be stored in a column- or a row-oriented format is defined by the field TYPE in the upper-right-hand corner of Figure 4.17; in our case, the table is row oriented (ROW STORE).

Row-based table

The table will have a simple structure, will not contain a lot of records (only one entry for each forecasted parameter), and the potential for compression is limited. Do keep in mind, however, that when talking about exchange rates you would still have one entry per country per month (as expected values and standard deviations could be different for each combination of these two characteristics). Nevertheless, we opted for ROW STORE; another reason for this was the fact that aggregating distribution-related parameters across random variables doesn't seem to make much sense to us.

Other modeling options

Alternatively, data stored within a table, like the one shown in Figure 4.17, could also be held in a DataStore object (DSO) created within a classic or SAP HANA-based SAP BW system. This DSO could be embedded into a data flow that is based upon a push web service of an external partner delivering exchange rates. Likewise, instead of being based upon a web service, the data flow could process data from a table within any external, non-SAP HANA database-management system or from a CSV file periodically provided by RFT's external partners. In each of these cases, the model shown in Figure 4.16 doesn't change (substantially).

Standardized data vectors

Wherever source data are coming from, the only thing RunFlat's app (that is, the algorithms sitting between the analytical and calculation views in Figure 4.16) should have to be aware of is that the layers underneath are going to deliver between *2 * n* and *3 * n* data vectors for *n* different parameters:

1. For each of *n* parameters that are either forecasted or planned by RFT, there is one data vector containing distribution-related information; if all of these parameters are normally distributed, then each of these vectors looks like a data record from the table shown in Figure 4.17.
2. For each of RFT's *n* parameters, we will get actual values (once again, one vector per parameter); the number of objects within each of these vectors will depend on how many actual values there are.

3. For planned parameters, there will also be corresponding vectors containing target values (that is, the values produced by the respective models if they had known the actuals for their input parameters). For forecasted parameters, there are no models and so no target values, which is why we said we are going to get between *2 * n* and *3 * n* data vectors for n different parameters.

Number of parameters (n) irrelevant from app's perspective

Based upon these *2 * n* to *3 * n* data vectors, RFT's app could, for example, execute between n and *2 * n* chi-squared tests for goodness of fit (that is, *n* tests comparing forecasted/planned and actual values plus—if applicable—*n* tests comparing target and actual values). If your data model is as abstract as the one shown in Figure 4.16, then you don't have to know about how many parameters you are working with (that is, whether *n = 10* or *n = 10,000*). You also don't even have to know what these parameters stand for from a business or functional perspective or whether the parameters you are monitoring this year are the same ones you kept an eye on last year.

Abstraction = flexibility

Okay, a lot of that sounds a bit complex and abstract, but abstraction is exactly what makes this model so flexible. Even if you need to add or remove data records from the table shown in Figure 4.17, you don't have to change anything within the analytic and calculation views based upon it. Your app executes whatever tests are suitable for all random variables mapped in that table, and if you limit further processing to a subset of the data contained in the DATA ACQUISITION AND DATABASE LAYER, all you need to do is, for example, add a `Yes/No` field (or any other field controlling the behavior of algorithms) to the table in Figure 4.17.

The algorithms themselves (that is, your analytic and calculation views) will no longer depend on (frequently changing) contents or business logic but only on the mathematical and statistical properties of your data—properties that are usually quite stable in the long run. RFT (or you) will end up with a data model that is extremely adaptable and needs very little maintenance.

Configuration Instead of Programming

To explain the difference between configuration and programming (something SAP was sensitive to right from its early days), we first need to define horizontal versus vertical data flows. We have already talked about horizontal and vertical categorization, both terms (horizontal and vertical data flows) are already used in Figure 4.16, and both will be picked up again in other case studies.

[«]

Horizontal Data Flows

Horizontal data flows consist of the following:

- Data flows between different applications (for example, data flows between an ERP system and a data warehouse)
- Data flows between clearly separated data pools within one single application (such as looking up master data records or enriching data)

Horizontal data flows are thus primarily used to move data from one area to another or to make data (temporarily) available elsewhere but not to (fundamentally) modify them.

Data logistics for horizontal data flows

Quite often, horizontal data flows take place within products for data management (data logistics, metadata); horizontal data flows can, however, even exist if no data are moved at all:

- If both its source and its destination are located within an SAP HANA database, a horizontal data flow can be modeled by creating a view that refers to the source's tables as a destination.
- Even when crossing system borders, horizontal data logistics does not necessarily need products for data management. As long as no transformations are required, the clear separation between source and target can be accomplished using virtual tables that access external sources from SAP HANA.

[«]

Vertical Data Flows

By contrast, we call data flows *vertical* if they are not primarily moving data around but are predominantly processing, analyzing, and exploiting them.

In terms of our definitions, feeding sales quantities and sales prices from an external solution into SAP BW would be a horizontal data flow. Multiplying both values within an SAP BW transformation (to calculate revenue) would be a vertical data flow.

In Figure 4.16, we have tried to illustrate the distinction between horizontal and vertical data flows by using arrows for horizontal and plain lines for vertical data flows. Data logistics tools (which are used in horizontal data flows) are represented by the two circles labelled DL1 and DL2.

Purpose of horizontal data flows

To recap, horizontal data flows serve two purposes:

- They make data from one application/database available in another application/database (regardless of whether this means physically transferring/replicating the data or not).
- When making data available, horizontal data flows either leave them unchanged or transform them. Transformations, however, do not process data using business or statistical algorithms. They only cleanse, aggregate, consolidate, or enrich data, adapting what the source can deliver to the needs of the destination.

In Figure 4.16, horizontal data flows feed data into DL1 and DL2 and pass on data from DL1 and DL2.

In contrast, vertical data flows process data. In a big data/SAP HANA environment, *processing* often means *analyzing*. If the results of these analyses are then needed in other applications or databases, then you will once again use horizontal data flows to make them available. In Figure 4.16, there are vertical data flows in all three areas of the diagram.

Avoid expensive shortcuts

Quite often, things get mixed up. When using (often powerful) transformation tools in data logistics solutions, organizations (and especially experienced programmers) are tempted to execute business logic in horizontal data flows. This makes tracing back outcomes extremely difficult; such an approach covers the tracks between original source data and insights gained from them and dramatically increases the effort required to maintain the system. Indeed, it's our strong belief that fixing the problems caused by such design errors has become the key source of revenue for offshore IT service providers.

Use on-board means

For both horizontal and vertical data flows, we'd recommend configuration instead of coding. In other words, move data around with on-board means—that is, use tools and configure them according to your needs instead of developing processing logic from scratch.

Because horizontal and vertical data flows are based upon different kinds of solutions, this implies the following:

- **Horizontal data logistics**

  Data can be acquired using any of the tools introduced under "Products for Data Management (Data Logistics and Metadata)" in Chapter 2, Section 2.1.3. This objective can be accomplished in three ways:

  - By accessing data via virtual structures (such as views or virtual tables)
  - By replicating data using appropriate tools (such as SAP SLT)
  - By replicating data using ETL tools (such as SAP Data Services) and (if necessary) transforming them on the way

  In all three cases, one should whenever possible avoid classic procedural programming and should instead simply configure (which means that certain settings should be made with, for example, transformations in SAP Data Services). Our experience has shown that configuration will suffice in the vast majority of cases. Unfortunately, organizations frequently choose programming because team members responsible for designing data flows often have extensive programming experience and prefer to do what they are most familiar with.

  We are aware of the fact that even configuring transformations will in the end lead to some kind of executable code in the background. Admittedly, automatically generated code is often less elegant than code that human programmers could deliver, but it does have a priceless advantage: having been generated automatically, it can be regenerated automatically if required and will therefore never cause you a headache with new releases.

- **Vertical data logistics**

  With vertical data logistics, requirements are slightly different. Here you rarely get away without some kind of coding, but in most cases you are at least able to use declarative instead of imperative languages, so you are a bit closer to configuration than you are with classic programming (also see the next section).

Our goal is to design vertical data flows in such a way that an experienced expert (one not, however, familiar with our specific environment; maybe a data scientist, developer, or consultant) is able to find her or his way around them without extensive documentation. (Have you ever seen a project that delivered 100% of the desired documentation?)

Declarative/Functional Instead of Imperative

Tools for vertical data flows

To model vertical data flows in SAP HANA, first of all you will need classic database operations, as defined in Section 4.5.2 (`Union`, `Join`, filters, and so on). All of that can be found in SQLScript.

If SQLScript isn't enough, then you'll probably want to use any of the following, ordered by power (see the recommend resources in the book's online appendix for links to additional reading on these topics):

1. Other declarative logic in SQLScript
2. Calculation engines (CEs) in SQLScript
3. SAP PAL and, in the future, SAP InfiniteInsight (formerly KXEN)
4. RDL as a declarative language generating SQLScript
5. R (as a functional language)
6. Imperative logic in SQLScript
7. Other declarative languages
8. If absolutely necessary, other imperative languages (such as ABAP)

Declarative and functional languages

As you move down that list, you will have more work to do and at the same time will be moving further away from configuring/declaring what you want and closer to programming how you want things done.

One of the key differences between configuration/declaration on the one hand and programming on the other is that with configuration or declarative and functional languages one can often see at first sight *what* is meant to be done. With imperative language, you can see the *how*, but the *what* is to be figured out by using (often nonexistent) documentation or—in the worst case—by debugging the code.

Under "Application Logic in the Database Layer" in Chapter 2, Section 2.1.3, we carved out the advantages of declarative programming languages. Because we are including functional languages here, we will go one step further and claim that one could—in analytical apps—live without imperative languages altogether.

[«]

Functional Programming Languages

Functional programming languages treat computing like the execution of mathematical functions, mapping input data onto output data. Functional languages get along without loops and value assignments and only use expressions instead. Iteration in functional languages is often accomplished via recursion.

Registering/ collecting metadata

Another advantage of configuration—compared to coding—or of declarative/functional languages—compared to imperative ones—is that your data model still remains transparent and searchable from the perspective of bots and crawlers pecking away on behalf of metadata repositories. Furthermore, your procedures are a bit closer to your business requirements than they are when using imperative languages.

The information that certain data are analyzed using this or that (statistical) algorithm usually means more to functional experts than the fact that two fields are processed by an `IF...THEN` statement.

[«]

Bots and Crawlers

A *bot* is a computer program continuously and more or less autonomously executing a certain task. A *crawler* is a bot that independently searches and analyzes data. Metadata repositories might use crawlers to detect new objects in a company's data dictionaries or models.

Virtual Instead of Persistent

Virtuality = flexibility

When explaining SAP's LSA++ architecture model (see Section 4.5.1), we mentioned that the high performance of in-memory and distributed computing has boosted the tendency to create more virtual, instead of persistent, objects and layers, but although high speed is a prerequisite for this trend one might still ask *why* virtual objects may be superior to persistent ones.

Let us use an analogy to explain this. In a world in which many of us have to change jobs, homes, and even partners every couple of years, a respectable house with a nice garden might be an attractive idea but not a very practical one. It takes some ten years for bushes to settle in and grow and closer to twenty before many trees reach a respectable size.

In this timespan, many people have worked and lived in a couple of countries, maybe even on more than one continent. The wagons of the early American settlers or the luxury RV better reflects today's lifestyle than a house made of bricks and mortar and passed on from generation to generation.

Data structures are not for eternity

Whatever kind of life you may prefer, in a business environment structures are often even less stable than in your personal surroundings. A couple of years ago, Facebook was *the* novelty among teenagers; now quite a few youngsters see it as a platform for an older generation. Furthermore, the factors determining how long visitors are lingering on your website and whether they want to book a flight with you or with a low-cost carrier can change within a day. As a result, it doesn't make much sense to build durable data structures that are carved in stone—even less so when it comes to planning models that might be subject to permanent change.

Performance without persistence

A couple of years ago, we were telling our customers to persistently store calculated key figures in data warehouses, thus improving reporting performance. There used to be a considerable tradeoff between speed and flexibility. With SAP HANA, this no longer plays a major role, at least for now; increasing data volumes might sooner or later absorb an in-memory database's extra performance, and back around the loop we might have to go.

As long as this is not the case, however, persistent structures are only needed if intermediate or final results have to be conserved permanently for auditing, traceability, or historical reasons or for later analysis (see our remark about using historical patterns as the basis for learning in Section 10.3.3).

The two principles "declarative/functional instead of imperative" and "virtual instead of persistent" are the reason that we are using three SAP HANA–specific object types (attribute view, analytic view, and calculation view) in the model illustrated by Figure 4.16. In the course of the next case study (see Chapter 5), we will take a closer look at these object types.

Business Process–Oriented Instead of Performance-Oriented

Finally, our data model is built around the processing logic to be applied to the data; it is therefore structured on the basis of mathematical and statistical criteria. Questions of performance (such as logically partitioning data) play a secondary role and have been ignored.

On the face of it, structuring a data model according to processing logic might seem to contradict the primacy of business processes. This is, however, only a contradiction at first glance:

- When saying that your data models should be divided by the mathematical properties of your data and by the way in which these data can be processed, we are talking about your data models' horizontal structure.
- At the same time, the model's vertical structure (such as the number of layers or analytical apps piled up on each other) is determined by your business requirements, their complexity, and the number of processing steps and tools required to get from input to output.

With these remarks about data architecture, we have now drawn a full circle from a general definition of planning via a fictitious sample scenario featuring particular planning- and forecasting-related challenges (exchange rates), benefit potentials, and value drivers to the resulting functional requirements and their impacts on architecture.

Next, case studies

The structure of the following case studies will be similar to this chapter. The scenario in this chapter is also meant to lay a foundation for those that follow.

In the next chapters, we are not going to repeat all that has been said so far; instead, we will focus on the peculiarities of the respective scenarios and the resulting conclusions. Each case study is meant to emphasize one specific business requirement, highlighting this requirement's effect on solution and data architecture.

We are drowning in information but starved for knowledge.

John Naisbitt, Megatrends

5 Reducing Travel Costs and Travel Times

Having finally arrived, Derek leaned back in the lounge chair by the window of his English uncle's holiday home. He could see a flock of Shetland sheep peacefully grazing in a nearby field, and behind Badentarbat bay the slopes of An-Teallach rose up to over 3,000 feet. The upper third of the mountain range was clad in snow, making a perfect canvas for the rich red sunset. Thanks maybe to the relaxing effect of the view, or perhaps to the glass of Talisker in his hand, the arduous journey slowly faded away, to a hazy image of airports and train stations.

Derek—a perfectly organized IT consultant by nature—had been meticulously planning his way here for weeks. Via a search portal for low-cost flights, he had found a connection from Niederrhein airport in Germany to Stansted that cost just €17.99, but getting to the airport by train on Monday morning had taken five hours, and his flight had departed six hours later than scheduled because of a technical problem. Under European passenger rights, he would have been entitled to a compensation of €250, but the airline was well known for quietly forgetting such impertinent claims from their passengers.

He had finally arrived at Stansted in the middle of the night. At this time of the day, train connections to Gatwick had been pretty sparse, and his non-rebookable connecting flight to Inverness had departed long ago. Left with no choice but to grimly accept the situation, he had decided to spend the night in an uncomfortable yet expensive airport hotel. On Tuesday morning, he had taken the train into to London, with nothing to do but kill time on a typical English rainy day. After some 11 hours of boredom, he boarded the Inverness-bound Caledonian Sleeper at Euston, heading in the right direction at last.

Figure 5.1 Sunset over the An-Teallach Range, Ross and Cromarty, Scotland

Arriving at the car rental counter just before sunrise on Wednesday morning, he had found a handwritten note at the counter informing him that they wouldn't be opening for another three hours.

Including the drive through the Highlands by car, he had spent two and a half days and well over €1,500 to cover a distance of about 700 miles (as the crow flies). Why hadn't he just taken the Eurostar train from Bruxelles Midi to London St. Pancras? In fact, the early connection departing Brussels at 8 a.m. would have taken him to Inverness for €100 in just 12 hours.

Derek sipped at the single malt. Downstairs in the basement, his uncle, a retired banker from the City, had stored some boxes of the 25-year-old vintage (worth about €400 a bottle); but from Derek's point of view, the Talisker Storm (just a tenth of the price) was much more suitable for this kind of weather and for the smoked wild salmon waiting for him in the fridge. There were hundreds of types and brands of Scotch and the price for an exclusive single malt could be as high as €60,000 a bottle. Picking the right one, however, was not just a matter of money. The choice depended on one's mood, the season, wind, weather, and, so importantly, the culinary delights on offer.

The weakness in his travel plans was that he had let the search engines seduce him and focused on the ticket price, completely ignoring other dimensions, such as transfer times or the reliability of the various means of transport and carriers.

When it comes to traveling, the Internet provides us with an unmanageable flood of data that doesn't make decisions easier. Things may look different, however, from the perspective of a company employing 10,000 top-notch consultants who are working on-site with customers most of the time. Small but sustainable improvements of one or two percent in terms of travel costs and travel times can easily add up to seven-digit numbers. This is why, in this chapter, we will take a closer look at resulting potential benefits of big data solutions in this area.

Flood of data

First of all, we will extend the definition of travel costs a bit, going beyond the mere dimension of cash, after which the following fictitious scenario will be presented. A consulting firm, operating globally, is wondering how travel planning can be improved to reduce overall travel costs. Initially, we are going to examine their as-is costs, risks, and opportunities (that is, before implementing a big data/SAP HANA solution); we will then—as before—develop an approach that is independent of SAP HANA, looking at business requirements and checking resulting benefit potentials and value drivers.

Wider definition of cost

Our key objective in this chapter is to familiarize you with the inductive approach to data, one that is especially important for big data. We won't talk again about monitoring models (as we did in the previous chapter) but instead will discuss how models should be developed. As in all chapters, we will then show the generic solution within the SAP HANA space.

5.1 Time is Money

Theoretically, business trips should be a relic of the past in the era of online video conferencing, but despite all the talk about recessions and crises, both airlines and airports all over the world are extending their capacities. Telephone and screen sharing are good enough for simple support services, but issues with complex negotiations are often best sorted out face-to-face, all the better to observe the body language of those you are bargaining with.

Business trips remain indispensable

Travel costs have always been a favorite target of cost reduction efforts. At first glance, they frequently offer substantial potentials for savings that are a lot easier to realize than higher revenues from higher sales

Travel costs as a target for cost reduction

quantities or sales prices. Customers can walk away and buy with your competition; when it comes to travel costs, the company is in a customer role, both when dealing with suppliers of travel services and its own employees.

Classic cost-reduction measures

When travel costs are the focus for cost savings, four things are likely to happen:

- Travel policies are changed. Flights in business class or train trips in first class are limited to certain hierarchy levels or trip durations.
- Travel budgets are cut by a certain percentage across all areas of the organization, and everybody is encouraged by the boss to somehow implement these reductions.
- Travel expense claims are checked more carefully. Existing or new rules are enforced more strictly.
- Travel departments or external partners are urged to thoroughly research and compare prices and to use special search engines and portals on the Internet to do so before booking.

SAP solutions for travel management

All of that is easy to justify, and SAP supports the implementation of your company's travel policy with a variety of solutions. SAP Travel Management in connection with SAP ERP Financials (FI) and SAP ERP Human Capital Management (HCM) can help you ensure that guidelines are respected. Prices can be compared using SAP's integration with booking systems like Sabre or just on the Internet.

Transport providers beating back with big data

Meanwhile, however, providers of travel-related services have used the time to optimize their revenues and upgrade their booking systems with ever-more-powerful algorithms (many of which are big data solutions themselves). On routes like Zurich to Brussels (very little competition, because other means of transport, such as cars or trains, are a lot slower), flight tickets are sold at premium prices; at the same time, tickets on more competitive flights that are more or less the same in terms of flight distance (like Geneva to London) are sold for a fraction of that price.

Ticket prices by time of booking

Quite a few airlines are pretty good at not only structuring their pricing by destination but also by time. We briefly touched on the topic of temporal segmentation in Chapter 1, Section 1.5, and will return to it in Chapter 7.

On a Sunday, a day on which many customers are planning and booking their family holidays, a flight to a popular holiday destination might be offered at a higher price than at 9:30 a.m. on Monday morning (we are assuming you are an honest person and have never used the Internet at your office for private purposes!). Airlines or rental car companies can also make life difficult for price-comparison portals by technical means—for example, by artificially extending response times.

This arms race of price segmentation on the one hand and price comparison engines on the other is going to continue for quite some time. Reducing travel costs is, however, not only a matter of comparing fares, focusing on one variable within the equation (transfixed like a rabbit in headlights). If you do so, you may end up like the rabbit, missing much better bargains and becoming road kill.

Three types of travel costs

From our perspective, travel costs (for example, for a consulting firm) consist of at least three components:

- **Costs for travel-related services/allowances**
  This category embraces classic travel costs, those that most of us think of first in this context—things like costs for flight or train tickets, rental cars, hotel accommodation, daily allowances, and other travel-related services.
- **Travel-related opportunity costs**
  By this, we mean losses in terms of labor time and output caused by traveling. With consultants that are paid on an hourly basis, lost revenue could serve as a measure. This is why we went for an example with a consulting firm. In other industries, such costs may be a lot more difficult to determine; however, they must not be underestimated.

  In principle, mobile communications enable us to work all the time, anywhere. In reality, however, developing a sales presentation on a cheap flight with minimum seat distance and crying toddlers around you or—even worse—on the London underground might be tricky. Catching a nasty cold isn't tricky, however, which could further reduce the company's income.
- **Soft travel costs**
  This last group of travel costs comprises everything that is difficult or impossible to measure but can nevertheless have a significant impact.

If a company changes the rules for business travel, asking its employees to no longer get there one day in advance for early morning meetings and thus saving accommodation expenses, an overeager sales manager might have an accident on his way to the airport at 4 a.m. and then be off sick for three months while at the same time suing the company, or he might make an expensive mistake when negotiating prices. In both cases, spending €100 less on a hotel stay turned out to be shortsighted and not worth it.

Measuring the unmeasurable

Some of the components mentioned previously are admittedly very hard to get to grips with, but ignoring them because of that is just sticking your head in the sand. When dealing with this chapter's case study and potential solutions, we'll have to think about how we could manage all three cost categories.

In the previous chapter, we looked at how to monitor forecasts and models; now we are going to *develop* these models. Our key challenges are as follows:

- Theoretically, an unlimited number of input parameters could be relevant for our model.
- In practice, only a small fraction of them are significant, but picking which are (and aren't) is about as tricky as picking the winner of the Grand National.
- The input parameters we are dealing with are very different in terms of their mathematical properties (like level of measurement, granularity, and so on).

Process applies to many business issues

The process presented in this chapter for developing models can be transferred to many other business issues; it doesn't matter one jot whether you are dealing with modeling/predicting travel costs or sales figures.

5.1.1 Costs for Travel-Related Services/Allowances

When and what to measure

Capturing expenses for travel-related services as well as for allowances isn't too much of a headache. In all cases, there is a flow of cash into the pockets of employees or third parties; costs are therefore clearly visible. The only two questions we'll have to answer here is when and what to measure.

- **When**
  - As long as a trip has not been booked (or the booking can still be changed) we are dealing with *theoretical* costs that can change every minute. We therefore need to keep an eye on quite a wide range of sources of data in real time.
  - Once the trip has been confirmed and booked, we finally have firm data. Unfortunately, it's now too late to correct any bad decisions; the only thing we can do now is learn from our mistakes.
- **What**
  - Derek's odyssey clearly shows that, in most cases, we don't only have to decide between different providers of the same service (like flights) but instead have to choose between different means of transport (like flights and trains) or even combinations of them. Even worse, using Stansted instead of Heathrow (both airports reasonably close to London) also gives us different options for continuing the journey.
  - Such comparisons are difficult or impossible for most travel portals. With Flightcentre (a well-known UK domestic and international travel company), for example, you can search for flights to (or from) London (as one single entity) and get quotes for flights to one or more of the London airports (Heathrow, Gatwick, and so on) so that you can compare the prices for flights to all these airports. The portal does not, however, take into account that, in getting to or from any of these airports, you are going to face very different levels of inconvenience and costs. It also doesn't know how far the airports are from each main line metro station, let alone from each other or from your final destination.
  - Therefore, Flightcentre has only helped Derek optimize *one single aspect* (the fare) of *one single leg* (the flight) of his trip, considering only *one means of transportation* (a scheduled flight). Try to think beyond travel costs, and generalize the example a bit. How good do you think you'll be in terms of optimizing business decisions if you are wearing such limiting blinkers?
  - If your point of departure is Brussels or Paris, you should definitely consider taking the Eurostar train right into the center of London

instead of flying to an airport miles away from the city. Using the Flightcentre website, however, you are not able to compare train journeys; that's not their area of business interest. The German railway's portal—which is pretty good at coming up with train connections for the whole of Europe (not for minor Swiss villages though)—doesn't help you with flights. What about coaches? Traveling by coach might be a bit slower but then again a lot cheaper.

5.1.2 Travel-Related Opportunity Costs

(Travel) time is money

Generally speaking, measuring travel times isn't a major issue. If we had a travel portal that could compare all modes of transport and all conceivable combinations of them, this portal could not only come up with a price but also with a duration for each option. Nevertheless, we would still face two issues:

- Are we talking about the *real* (or at least the *likely*) duration of the trip (in statistical terms: the expected value), or is this just a planned one, and do we have to assume—learning from past experience and punctuality statistics—that getting there is going to take us twice as long?
- Time is money is a truism, but from an arithmetic perspective a parameter measured in currency units and one measured in hours or minutes cannot be added together. We are not only dealing with different units of measurement, but also with different *dimensions* (money and time).

Actual Trip Duration

Timetables and actual travel times

Timetables are often not only wishful thinking, but also a marketing instrument for providers of transport services.

Thanks to the Internet, it is, however, relatively easy to find out how often (and by how much) planned and actual flight times vary. There are quite a few websites that feature historical delay statistics for airlines and airports. Up-to-date departure and arrival times can be found on airline or airport sites, and future ones can be derived from flight weather forecasts.

Things are getting a bit more complex when it comes to predicting how late you will arrive at your final destination if there happens to be (say) a delay of 45 minutes on a certain route at a specific date and time of day; to do so you would also have to include, for example, the effect of missed connections.

Aggregating Time and Money

How much is an hour worth?

It's relatively easy to calculate the costs of one hour of work for a certain employee. Well, we said *relatively* easy. You'll see that it is a piece of cake compared to the question we *really* need to answer here, which is this: What's the *value* of one hour of saved travel time?

With a consultant for whom your customers are paying by the hour (assuming that your customers don't consider work done while traveling as billable), the second question can still be answered. But what about somebody from headquarters? What would this person have done had that hour been spent in the office? Surf the Internet booking the summer holiday or have an idea worth millions to the company?

Although travel time might be lost time for some, others tend to be more creative on a plane than in their dowdy, somber office environments. Furthermore, there is a relationship between the means of transport and the kind of job that can be undertaken. An hour on a train or in the first-class lounge at an airport could comfortably be used to work on a laptop; an hour on a downtown train or driving a car is different, but if a phone call has to be made instead, then the car—not least due to confidentiality—might be a much better location (well, at least if traveling alone, which adds another variable to the equation).

5.1.3 Soft Travel Costs

The beast that is most difficult to tame is soft travel costs.

Costs of employee dissatisfaction

Percy Peckham, known to some of his less generous colleagues as "old PP," is a purchasing manager in the textile industry. In the past, he could use business class on a direct British Airways flight from London to Asia, but due to a change in travel policies, he is now supposed to fly Aeroflot (cattle class) via Moscow. This change saves the company 70% on his ticket but has a couple of side effects. In the good old days, Percy could board the plane in the evening, enjoy dinner at 40,000 feet, go for a nightcap or two, and then comfortably slumber in his fully horizontal sleeper seat for some eight hours. Now, his flight doesn't depart in the evening but late in the afternoon, taking him to Moscow in the middle of the night. Once at Sheremetyevo, the business class lounge and a free massage for long-haul passengers are nothing but a distant memory. Having arrived at Beijing, Percy is not only jet lagged but brain-dead too. His Chinese supplier's sales manager—a gifted psychologist who has also absorbed all the teachings from *The Art of War* (Sun Tzu)—senses a wonderful opportunity to get Percy to commit to three times the annual purchasing volume at a price 20% higher than last year.

Percy cuts and runs

Two years after the change to the travel policy, Percy thinks it might be a good idea to resign, taking his contacts to Asian suppliers and his knowledge about purchasing prices to a competitor who is a bit more generous and willing to let him fly first class instead of just business class in the future. This turns out to be a good deal for them because they are able to take over Percy's former employer at a bargain price one year later.

All these considerations involve factors that are hardly measurable at all. The sequence of events was (maybe) unpredictable, and evaluating all of that in currency units seems impossible even with hindsight. Nevertheless, we are going to face this challenge in Section 5.4. We don't claim to be the fount of all knowledge, but we will still provide you with a couple of helpful ideas.

If you want a definitive solution for the problem, just turn to Walk-on-Water, the consulting firm that we are going to introduce in the following section. It might not have one either, but it won't hesitate for a moment to try to sell it to you.

5.2 Scenario: Travel Costs with an International Consulting Firm

The (fictitious) consulting firm Walk-on-Water Associates (also known as WoW) is one of the world's leading strategy consultants. Thanks to excellent contacts with decision makers in business and politics (mainly in Asia), Walk-on-Water has experienced a meteoric growth rate in the last couple of decades. At the moment, the company employs some 15,500 people at 55 locations all over the world.

Administrative processes outsourced

Almost all administrative processes (accounting, personnel administration, travel management, and so on) have been outsourced to offshore service providers. WoW's own administration is extremely lean. Its utilization rate is excellent; theoretically, all consultants could be working with customers 100% of the time, which is why Walk-on-Water has tried to free its people of all administrative, nonbillable tasks. When it comes to traveling, consultants send their requirements to a travel center that handles all of their travel planning and travel booking and also helps them claim back expenses.

All consultants have company credit cards that they use to pay the lion's share of whatever cannot be paid in advance by the travel center. The only thing consultants have to do at the end of a business trip is scan related documents using an application on their smartphones and electronically sign the expense sheets and credit card statements that have been prepared for them.

New service partner

To improve integration and transparency (and after going cross-eyed when looking at their new, cross-function analyses), WoW's top management has been thinking about bundling all processes that can be supported by an ERP solution with one single service provider. At the moment, the favorite candidate is a company called SAP Rickshaw (as their name indicates, SAP Rickshaw is using SAP solutions). If SAP Rickshaw wins the contract, WoW's travel management would then be based upon SAP ERP Human Capital Management (HCM). Everything SAP Rickshaw has to offer is available as a hosted, cloud-based service.

Bottleneck: capacity

Although Walk-on-Water has invested a lot of time and creativity into developing exemplary admin processes for all nonbillable tasks, the company will soon reach its limits in terms of growth:

- Capacities in consulting are not open-ended. Highly qualified experts are thin on the ground and much sought after. Watering down their level of quality and expertise is not an option for WoW, and experienced people leaving the company are very difficult to replace.
- As Walk-on-Water is offering highly priced services, most customers insist on maximum on-site presence of consultants. On the other

hand, it is impossible for WoW to have experts on everything at every office. Furthermore, there is an imbalance between the supply and demand for consultants. For example, Walk-on-Water's best experts in terms of data science are former Soviet nuclear physicists based in the European part of the Russian Federation; they are, however, mainly in demand in Asia.

The combination of all these facts leads to two consequences:

- Minimizing travel times and reducing travel-related stress are far more important for WoW than just simply driving down hard travel costs.
- On the other hand, WoW doesn't want these hard travel costs to get out of hand.

5.2.1 Collecting Ideas

Brainstorming about travel costs

About a year ago, WoW's board of directors set up a new internal project called NUTS: Never Underestimate Travel Stress. Considering the fact that WoW was putting admin support out to tender, the main goal of that project was to develop ideas for a holistic view on travel cost, outlining approaches for travel cost optimization.

Ideas for improvement

In the course of a presentation to top management, the leader of that team has come up with the following proposals:

- Collect data concerning the punctuality and service quality of airlines and data about the factors determining these qualities.
- Set up an early warning system for major delays affecting a certain means of transport (such as difficult weather conditions affecting flights).
- Create a geocoded database that contains the consultants' experiences from previous trips.
- Develop a system that can compare travel costs and travel times across all means of transport and for complete trips (instead of only looking at hard travel costs and individual segments of a trip).
- Analyze feedback from employees about the comfort (or otherwise) of individual trips, evaluating this largely unstructured feedback via sentiment detection.

Geocoding [«]

Geocoding is the process of converting text-based location information (for example, street addresses) into geographic coordinates (latitude and longitude). Geocoding can be performed by locally implemented solutions and by web services and often goes hand in hand with address verification or cleansing.

For the SAP community, geocoding is available in SAP Data Services (`Geocoder`) as of release 4.2 SPS 6 (Beta) or SPS 7 (productive); SAP HANA also comes with specific functions for storing and processing geospatial data.

Flight delays and flight weather online

In the course of the presentations to the board, some very convincing travel apps (such as FlightTrack by Mobiata LLC) were demonstrated. FlightTrack is an app that can show flight information across airlines and airports. The project leader also mentioned that data related to factors determining punctuality were (to a certain extent) available free and in easily processable formats on the Internet. As evidence of this, he showed the so-called Meteorological Aerodrome Reports (METARs) and Terminal Aerodrome Forecasts (TAFs), weather reports that are standardized internationally.

5.2.2 Strategic Decisions

Basic decisions

Based upon the project team's presentation, WoW's management have made the following long-term decisions:

- In the future, WoW would like to take a holistic view of travel costs.
- Therefore, WoW wants to develop systems that are able to optimize business trips, taking into account all costs (including opportunity costs and soft costs). On top of that, these systems should be able to forecast and to take into account expected deviations from flight schedules or timetables.
- WoW would like to set up an early warning system. Once an alert is triggered—for example, if fog is forecasted at London Heathrow, probably leading to delays of one to two hours—external service providers (travel admin or travel agents) will be informed and asked to cancel or rearrange trips.

Management affected as well

Most members of the top management travel a lot, both in their role as strategy consultants serving external clients and to attend sales meetings and the like, so it doesn't come as a surprise that the majority of participants in that meeting were happy with these principles; most of them considered solutions like those presented both useful and fascinating. WoW's head of controlling, however—the only member of that exclusive circle hardly traveling at all—was skeptical. His view was that WoW was facing the danger of investing a lot of time, energy, creativity, and money into a solution that would a) never work and b) make life cozy for the consultants but have no tangible benefits beyond that.

Other ideas in controlling

The project leader has learned from the controller's assistant that, instead of setting up an early warning system, controlling would like to suggest some changes to WoW's travel policy. When visiting one of the company's major customers at London, they should no longer use Lufthansa flights from Frankfurt into Heathrow but should instead fly Ryanair from Frankfurt-Hahn to Stansted.

Proof of concept (PoC)

After a short discussion, the board therefore decides to charge the NUTS team with analyzing potential benefits. To start with, they have to find out whether expected delays could be forecasted on the basis of, for example, airport weather reports and how long before the event such warnings could be triggered (an early warning system would not be much use if the alert occurs just as the consultant is parking his car at the airport).

5.3 One-Dimensional Optimization: Costs, Risks, and Opportunities

If our project leader has a good nose for psychological undercurrents and political views in organizations, then he might not (or at least not immediately) start by looking at the planned early warning system. Instead, he might think about how influential and well-connected the skeptical controller is, what the reasons for his negative attitude might be, and what consequences could result from his proposal for using Ryanair.

5.3.1 Problem: Politics and Organizational Psychology

Handling political resistance

Quite a few consultants with WoW are frequent travelers and therefore hold the exclusive HON-Circle status with Star Alliance airlines.

These consultants collect so many bonus miles every year that they can take holiday trips to the Caribbean or Asia with their partners for free! Meanwhile, the controller—living at his house at Opfikon near Zurich airport—suffers from more and more noise caused by aircraft landing and has never managed to get beyond the lowest status with these airlines.

Preparing for battle

It is, however, obvious that this guy is not going to let envy be seen as the driving force behind his skepticism. Instead, we can assume that he will brace himself with a lot of supporting data. The project manager should therefore follow a dual strategy (or even a "duel" strategy):

- First, he should also put together some suitable numbers that will help him survive the expected discussions about the financial benefits of the new solution.
- Second, he needs to be able to substantiate that the data in scope (such as flight weather) will be suitable for forecasting travel times.

5.3.2 Numerical Example

Many years ago, WoW recognized the importance of business intelligence and invested in a very powerful SAP BW system that served as the basis for a variety of SAP BusinessObjects BI clients. Recently, WoW's SAP BW was migrated from an Oracle database to SAP HANA.

Although WoW does not manage its operational systems itself, its partners provide it (via DataSources for web services) with a lot of OLTP data for that data warehouse.

[«]

DataSource and DataSource for Web Services

Within SAP BW, a *DataSource* serves as a kind of interface to the outside world for incoming data. Technically, DataSources are nothing but sets of fields for which external sources are providing the content.

When trying to keep afloat with heterogeneous system landscapes or service-oriented architectures, *DataSources for web services* are often favored. Such data sources are not used to periodically load data from SAP BW, but instead to actively push external data regularly or as and when required from another system into SAP BW.

Supporting arguments from business intelligence

This data warehouse now comes in handy, helping the project manager do some quick analyses on using Ryanair to fly Hahn-Stansted:

- During the last two months, consultants have flown from Frankfurt to Heathrow 271 times.
- If these consultants had flown from Hahn to Stansted using Ryanair, then WoW could have saved an average €500 per flight, amounting to a total of €135,500 for 271 flights. This assumes that all the consultants had booked early enough and been happy to forego some of their perks, like having a meal on the plane and not having to pay extra for luggage so that they could take enough clothes for their trip.
- Furthermore, the average duration of a consultant's trip to the airport (Hahn instead of Frankfurt) would have increased by about 1.25 hours and the trip from their flight destination (Stansted) to their customers would have increased by another hour, in total leading to an extra 610 hours of travel time (for outgoing trips only!).
- Using an average hourly rate of €300, this would have equated to €182,925.

Of greater concern is the fact that in the course of their exit interviews five of the 13 consultants who gave notice last year cited "travel-related stress" as the most important reason for leaving the company. Although head-hunters have helped fill these vacancies in the meantime, recruiting costs averaged €120,000 per consultant, or €600,000 in total for those tired of hanging around in airports.

Major differences in terms of punctuality

As well as looking at costs, the project leader is also checking the benefits of travel time forecasts. Some research into delay statistics (source: *www.flightstats.com*, December 2013 to January 2014) showed the following:

- Almost all of the trips undertaken by consultants took place on Wednesdays between 9 a.m. and noon.
- According to FlightStats, 31% of all flights during this slot were delayed. Their average delay was 56.22 minutes.
- Between Frankfurt and London City, 38% of all flights on Wednesdays between 9 a.m. and noon were late. The average delay there, however, was only 23.5 minutes.

- Between 6 a.m. and 9 a.m., only 19% of the flights to Heathrow were late; the average delay was 40.12 minutes.
- When flying to London City on Wednesdays between 6 a.m. and 9 a.m., on average 14% of the flights were delayed; the average delay was 5.2 minutes.
- If the consultants had flown into London City instead of London Heathrow and if they had traveled between 6 a.m. and 9 a.m. instead of flying between 9 a.m. and 12 p.m., then time lost due to delays would not have been *271 flights * 31% * 56.22 minutes/flight = 4,723 minutes = 79 hours* but *271 flights * 14% * 5.2 minutes/flight = 197 minutes = 3.3 hours*. Note that these extra travel times due to delays would not take into account any follow-on interruptions caused by missed trains and the like. (To keep things simple, we assume that delays don't change dramatically across the three-hour slots we are talking about.)

Seeing the complete picture

Finally, it takes about 50 minutes to travel from Heathrow to Canary Wharf (the location of WoW's most important London-based customer), but getting there from the much more conveniently situated London City airport is a matter of 20 minutes by public transport or just 10 minutes by taxi.

5.3.3 Conclusion: Juggling Numbers versus Reality

Numbers are often ambiguous

The former British Prime Minister Winston Churchill is credited with saying "I only believe in statistics that I doctored myself"; indeed, with the figures presented previously, one can argue that these are at best theoretical and riddled with many ifs and buts:

- How on earth are we going to know that it makes consultants happier to travel three hours earlier? Could this actually increase the number of resignations?
- Do consultants prefer traveling 50 minutes on Heathrow Express's first-class service rather than being packed for 20 minutes into the underground-like Docklands Light Railway taking them from London City to Canary Wharf?

- Can saved travel time really be converted into revenue in a 1:1 ratio?
- Can appointments with customers be scheduled around optimized trips? Aren't travel plans driven by customer appointments instead?
- Finally, assuming we can sort out all of that, will WoW be able to gain any insights that have even the glimmer of a chance of having any kind of predictive power from that jumble of weather data and arrival times?

[+]

The Naked Truth

To be honest, we haven't got the faintest idea, which is, amazingly, one of the secrets of success when designing state-of-the-art big data solutions.

Start with being as ignorant as you can and let data lead you, working inductively instead of deductively. In Section 5.4 we will explain what we mean by this.

Within the scope of this case study, we are not able to answer all relevant questions and come up with the best possible solution for WoW. Therefore, we will focus on the last question in the preceding list: How can WoW find out whether flight delays can be predicted on the basis of weather data? As mentioned in Section 5.1, this chapter is not really about weather, flight delays, or trip durations. We only use these examples to illustrate how to detect and quantify dependencies.

Participation creates acceptance

Another remark in terms of your ultimate goals: you may or may not be able to build a system that can find the best (whatever "best" may mean) amongst zillions of theoretical travel options, but if you instead build one that presents some preselected alternatives to your consultants in a structured graphical format (for example, via a dashboard) that allows them to choose for themselves, then user acceptance and satisfaction might be a lot higher anyway.

5.4 Solution: Induction Instead of Deduction

Induction and deduction are two terms from epistemology that describe different approaches to achieving a deeper understanding. With an *inductive* approach, one uses individual observations to derive a general rule (via a process called abstraction). In our example, this would mean looking at weather and punctuality reports without prejudice, trying to gain insights by looking at the data. The biggest risk of induction is that data may mislead you. You may draw false conclusions or consider random phenomena the manifestation of a universal rule. When working *deductively*, you are using a general rule or a set of prerequisites to make conclusions regarding individual cases/events. One assumes that certain kinds of significant weather (storms, low visibility, and so on) are going to result in delays within the next three hours. Based upon that assumption, you are going to develop a system to detect the respective keys in TAFs, automatically sending a rebooking request to your travel agent. One of the problems with deduction is that you have a preconceived opinion about dependencies, which may not be accurate. Another major problem is that although the way you are drawing conclusions is perfectly reasonable you might still end up with poor results due to false assumptions.

Induction is (as a rule) more open

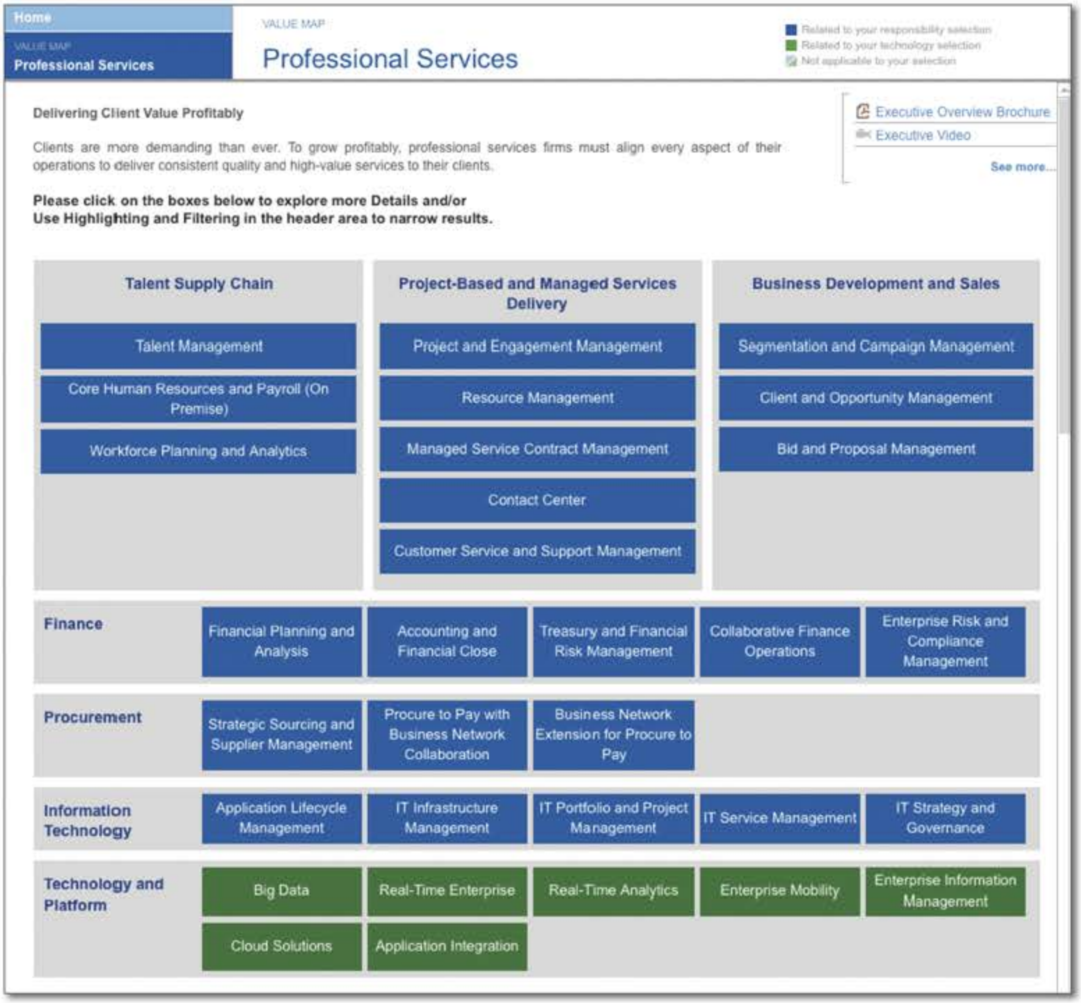
When analyzing data inductively, we are running at a lower risk of being misled by our own prejudices. Most inductive algorithms (such as cluster analysis) are based on certain mathematical or statistical assumptions, and it is extremely important that you or your data scientists understand these restrictions before letting such algorithms loose. Nevertheless, the scope of potential insights is a lot wider with induction. In the end, when working inductively you are not trying to prove preconceived hypotheses but are instead deriving new ones from your data (and then verifying these new hypotheses using different samples).

Configuration settings in ERP systems (such as SAP Business Suite) are usually based upon rules derived via deduction. One of the most interesting aspects of big data/SAP HANA is the fact that SAP HANA-based solutions enable an inductive approach instead. This is just the message we are trying to impart with this case study.

5.4.1 Related Value Maps in SAP Solution Explorer

Cross-industry and industry-specific value maps

Travel costs play a role in many industries. Related solutions from SAP can therefore be found in a cross-industry value map (FINANCE • COLLABORATIVE FINANCE OPERATIONS), but because we picked a scenario from the consulting industry, Figure 5.2 shows the industry-specific access path to travel management (via PROFESSIONAL SERVICES).



Home

VALUE MAP

Professional Services

VALUE MAP

Professional Services

■ Related to your responsibility selection
■ Related to your technology selection
▨ Not applicable to your selection

Delivering Client Value Profitably

Clients are more demanding than ever. To grow profitably, professional services firms must align every aspect of their operations to deliver consistent quality and high-value services to their clients.

Please click on the boxes below to explore more Details and/or
Use Highlighting and Filtering in the header area to narrow results.

Executive Overview Brochure
Executive Video
See more...

Talent Supply Chain	Project-Based and Managed Services Delivery	Business Development and Sales
Talent Management	Project and Engagement Management	Segmentation and Campaign Management
Core Human Resources and Payroll (On Premise)	Resource Management	Client and Opportunity Management
Workforce Planning and Analytics	Managed Service Contract Management	Bid and Proposal Management
	Contact Center	
	Customer Service and Support Management	

Finance	Financial Planning and Analysis	Accounting and Financial Close	Treasury and Financial Risk Management	Collaborative Finance Operations	Enterprise Risk and Compliance Management
Procurement	Strategic Sourcing and Supplier Management	Procure to Pay with Business Network Collaboration	Business Network Extension for Procure to Pay		
Information Technology	Application Lifecycle Management	IT Infrastructure Management	IT Portfolio and Project Management	IT Service Management	IT Strategy and Governance
Technology and Platform	Big Data	Real-Time Enterprise	Real-Time Analytics	Enterprise Mobility	Enterprise Information Management
	Cloud Solutions	Application Integration			

Figure 5.2 Professional Services Value Map

Solutions for Professional Services

Travel management is part of the Expense Management end-to-end solution that resides within the Project and Engagement Management end-to-end solution. SAP offers six individual solutions related to Expense Management:

- PreTrip Approval (Cloud)
- Online Booking (Cloud)
- Expense Management (Cloud)
- PreTrip Approval (On Premise)
- Online Booking (On Premise)
- Expense Management (On Premise)

5.4.2 Functional Requirements

Detecting dependencies

As mentioned before, WoW's first challenge is to find out if flight delays can be forecast using weather data when working inductively and without any preconceived views. Or—from a more abstract viewpoint—the project team is dealing with three tasks:

- **Detecting dependencies**
  They would like to find out whether there is a relationship between the weather and flight delays. Naturally, a simple yes or no won't do; we would like to know more precisely:
  - Which weather phenomena are likely to cause flight delays?
  - How tight is that relationship (for each weather phenomenon)—that is, what percentage of flights will be late and by how much (on average, expected value)?
  - How quickly does a certain weather phenomenon (like visibility below 2,000 meters) lead to delays? Immediately after occurring in a METAR? Five hours later?
- **Modeling dependencies**
  Let's assume we had found out that fog (for which the key "FG" appears in a METAR) causes 35% of all arrivals within the next 30 minutes to be late. When talking about delays of less than 15 minutes, it probably doesn't make sense to rebook or cancel flights; doing so on short notice would be pretty expensive. Changes to travel plans might only be sensible if we learned about the delay from a TAF 30 hours in advance.

This means that we need to model and (if possible) quantify detected dependencies. One of the challenges here is the fact that sometimes we are talking about dependencies between input and output parameters on a ratio scale (such as visibility or average delay), but on other occasions we are dealing with links between nominal input values ("fog" or "no fog") and ratio scale output values (average delay).

- **Verifying and monitoring dependencies**

  The fact that a dependency has been detected does not imply that this dependency is still effective or will still be effective in the future. Maybe the installation of a new instrument landing system on a runway is suddenly making an airport less prone to weather-related delays; maybe global climate change in general is leading to more severe weather conditions in winter. A lot of England thought this was pretty certain in November 2013 when three months of heavy and persistent snow was forecast.

  There are three ways to ensure that your forecasts and models are always up-to-date:

  - You could run the algorithms used to detect dependencies quickly and often, day and night, continuously adapting your model and comparing the predictive power of old and new rules. Based upon such comparisons, you may (alternatively or additionally) trigger alerts.
  - Or you go for the approaches extensively discussed in Chapter 4. After all, our predictions of expected delays are nothing but the output of models that use weather forecasts as their input. Hence, the basic structure of the problem is the same as that of sales quantities in Chapter 4.
  - There is no reason that you shouldn't combine both concepts, particularly if your SAP HANA implementation can give you the performance needed for this.

5.4.3 Building Blocks of the Solution

Suitable (statistical) tools

In Section 5.5, we will look at implementation scenarios or data models that WoW may deploy to detect dependencies between weather forecasts and flight delays. Before doing so, we will take a brief look at the statistical tools that are available to satisfy such requirements. These reflections are once again application neutral and database-agnostic—that is, they do not depend on whether you use SAP HANA or something else.

Detecting Dependencies

Covariance and correlation

The classic statistical measure for quantifying the strength of the dependency between two random variables on a ratio scale is covariance, or—a bit more precisely—the corrected sample covariance.

[«]

Covariance

Simply stated, *covariance* measures how closely linked two random variables seem to be by measuring their joint deviation from each variable's expected value (unlike variance, which only looks at how much one single random variable deviates from its individual expected value). A high (absolute) covariance is a fairly good indicator for linear dependencies between parameters that are on a ratio level of measurement.

A *positive* covariance indicates that both parameters tend to jointly move in the same direction; a *negative* covariance suggests that one parameter going up will make the other one go down. A covariance of 0 suggests that both random variables are statistically independent.

For mathematical reasons we don't want to explain here, you should keep the following in mind: when estimating covariance on the basis of a sample (and not using the whole population), you should always use the formula for what is called the corrected sample covariance. If your sample contains n values, you simply need to multiply the sample's covariance by $n / (n - 1)$.

Processing data that are not on a ratio scale

Like real life, however, our WoW example is not limited to linear relationships and things that can be measured on a ratio scale. Weather observations are often nominal (precipitation—rain, snow, hail, and so on—yes or no) or ordinal (light/moderate/heavy). Luckily, there are quite a few other measures of association; two examples of additional, more robust measures of association are the quadrant count ratio and the point biserial correlation coefficient. The term *robust* comes from *robust statistics*, a collective term for formulas and algorithms that are used for data drawn from a wide range of probability distributions. Robust formulas and algorithms are not unduly affected by outliers.

The covariance of two random variables depends on their dimension. If WoW wants normalized, comparable measures of dependency, then the correlation coefficient can be calculated from the corrected sample covariance. Unlike covariance, the correlation coefficient will always be a value between –1 (strong inverse relationship or *anticorrelation*) and +1 (perfect linear relationship).

Time-shifted dependencies and fractals

In a big data environment, the question of whether there is a time-shifted dependency between two random variables can be answered with justifiable effort. You only need to execute the preceding analyses with a time lag. Instead of measuring the correlation between "fog now" and "delay of more than 10 minutes now," you would calculate the one between "fog 30 minutes ago" and "delay of more than 10 minutes now" and "fog 60 minutes ago" and "delay of more than 10 minutes now."

But why stop there? You could use a 30-minute time lag and a 60-minute one, but why not go for a 35-minute time lag or one of 33 minutes and 17 seconds? Fortunately, METARs and TAFs are only published at fixed time intervals, which saves you from frying your brain by thinking too much about fractals, although thinking about questions like that will also provide you with some deeper insight into why we tend to be skeptical about the concept of reality. However, let's get into that another day, or in another book, and discuss the topic over a glass of single malt.

Modeling Dependencies

Gordian models

Once you have detected some kind of dependency, you probably want to conceptualize it—that is, write it down in the form of a rule, a formula, or a mathematical model. When doing that, the sky's the limit for your creativity, which also means that the risk of slipping back from induction into deduction is highest here. Furthermore, the dependency detected in your sample might be random.

There are three countermeasures you should take:

- Once you have designed a model, you should verify its validity on the basis of at least one additional sample, making sure that it is *not* the one you used to detect the dependency.

- Avoid slipping back into deduction by using statistical algorithms that are based upon as few premises as possible and by only applying algorithms appropriate to the underlying data.
- Lastly, confine yourself to algorithms whose potential and limitations you are reasonably familiar with.

Algorithms that can build models automatically

Typical modeling algorithms include the following:

- Multiple linear regression (as mentioned in Chapter 1, Section 1.4.2), sometimes also called multivariate linear regression
- Polynomial regression (as mentioned in Chapter 4, Section 4.2.1 and also available in SAP PAL) or segmented regression
- Decision-tree training algorithms, such as C4.5 or CHAID (both are also available in SAP PAL)

[«]

C4.5/CHAID

C4.5 and CHAID are two of the many algorithms that can be used to (automatically) generate decision trees from data samples. Both can be used to classify data. The acronym CHAID stands for *Chi-squared automatic interaction detectors*; CHAID uses the chi-squared test on independence to step-by-step split data. Unlike C4.5, CHAID can also process nonratio data.

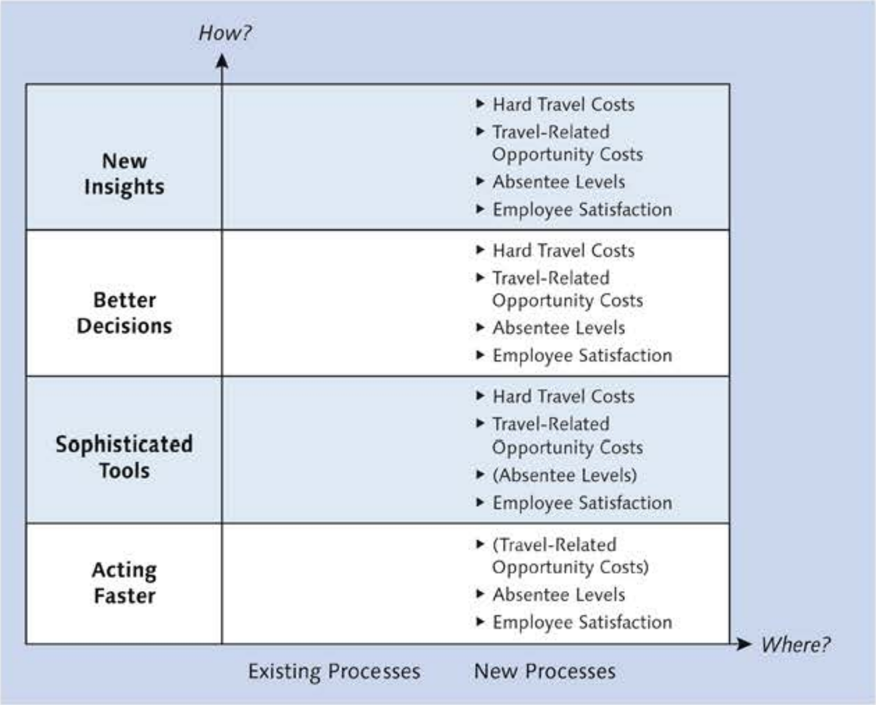
Verifying and Monitoring Dependencies

In the course of this chapter, we are not going to spend time discussing how to verify and monitor dependencies; have a look instead at our thoughts in Chapter 4, Section 4.4.

5.4.4 Potential Benefits and Value Drivers

Potential benefits derived from cost categories

Potential benefits and value drivers for WoW can be derived from the details in Section 5.1. Figure 5.3 provides you with an overview. The benefit–value driver matrix in Figure 5.3 is based upon the assumption that WoW will go beyond just analyzing potential benefits, as mentioned in Section 5.2.2. Instead, we are assuming that all the solutions and systems mentioned there will be implemented.



	Existing Processes	New Processes
New Insights		▸ Hard Travel Costs ▸ Travel-Related Opportunity Costs ▸ Absentee Levels ▸ Employee Satisfaction
Better Decisions		▸ Hard Travel Costs ▸ Travel-Related Opportunity Costs ▸ Absentee Levels ▸ Employee Satisfaction
Sophisticated Tools		▸ Hard Travel Costs ▸ Travel-Related Opportunity Costs ▸ (Absentee Levels) ▸ Employee Satisfaction
Acting Faster		▸ (Travel-Related Opportunity Costs) ▸ Absentee Levels ▸ Employee Satisfaction

Figure 5.3 Travel Costs/Travel Times Benefit–Value Driver Matrix

Soft travel costs = new processes

Traditional travel-management processes (travel planning, booking, and settlement) and IT solutions supporting these have been in place for quite some time at WoW, and they are not the subject of this case study. This is why all potential benefits are listed on the right-hand side of the matrix (under New Processes).

- **Hard travel costs**

  Although we are not focusing on hard travel costs (that is, costs for travel-related services and allowances) in this chapter, we assume that the solutions WoW is thinking about can still create value in those areas:

  - A holistic view on business trips going beyond individual segments or modes of transport will lead to more reasonable optimization results. A ticket from Hahn might be cheaper than one from Frankfurt, but making such as change means that instead of just refunding a ticket on a suburban train WoW may then have to compensate its employees for hundreds of Euros of taxi fares, mileage allowances, and parking charges that they incurred getting there.

- Delays in one segment of a trip often not only lead to follow-on delays but also to follow-on costs. When arriving late at the destination airport, nonchangeable tickets might be forfeited, there could be additional expenses for overnight accommodation, or a taxi may have to be used instead of public transport to make it to an appointment on time.

The aspects discussed so far have—apart from a few exceptions—gone into every How? row of the matrix. One of these exceptions is the row called ACTING FASTER. The early warning system WoW is considering would enable the company to act faster and may also help reduce hard travel costs, but its primary point of attack would be soft costs. This is why the value drivers of ACTING FASTER are different.

- **Travel-related opportunity costs**
  From our perspective, this category affects two value drivers; the only difference between them is to what extent traveling affects the ability to work productively:
  - Some travel time (for example, when using buses) can't really be used productively. With WoW, this has significant impact in the form of lower revenues. In other industries, evaluating such losses might be more difficult.
  - Although some travel time can be used productively, there are still limitations. In consulting, not all such time can be charged to customers; in other industries, unstable or unsafe Internet connections or confidentiality issues (making phone calls on a train) might restrict the kind of work that can be done.

The question of whether travel or waiting times are dead times or can be used productively will mainly depend on industry, job type, and means of transport, so we have put opportunity costs in brackets under ACTING FASTER. Also, a consulting firm like WoW will not always be able to convert all saved travel times into billable hours at short notice; customer availability, expectations, and order volume will have a greater influence here.

- **Absentee levels**

  It is well-established (see *https://www.iamat.org/editorials.cfm*) that travel-related stress can cause mental and physical illness. Frequent changes of environmental temperature and humidity and the contact with a multitude of people on public transport during flu season will increase the risk of infections. Uncertainty and frustration caused by delays and missed connections also play a role in the development of stress-related conditions. In the short run, they will result in lost revenues for a consulting firm like WoW; longer-term, they are one factor contributing to more serious health risks, such as burnout or coronary heart disease.

  As well as intercontinental flights into different climates and time zones, travel-related stress is often caused not by the journey itself but by the need to adapt quickly to changed, unforeseen conditions. Due to late arrival of a flight, a trip once carefully planned on the computer in the quiet environment of your own office might now have to be reorganized on the fly using a smartphone and a patchy Internet connection. Better and earlier forecasts plus an early warning system can help reduce the probability and impact of such problems.

- **Employee satisfaction**

  Employee satisfaction is a determining factor for commitment and work quality. In the long run, it also has an impact on staff turnover, and in Section 5.3.2 we showed that resignations are expensive for WoW.

  All four how dimensions have the potential to influence employee satisfaction, but in addition to these technical solutions, two other aspects play a major role in determining how well they really help improve people's happiness:

  - Improvements delivered by new solutions need to be visible/tangible, directly perceivable, and understandable to people who need to be informed about the functionality of the new systems and should be able, and encouraged, to feed back their own preferences—for example, in terms of what makes a business trip a positive experience.

- Employees should have the chance to actively become involved in using new insights. As we said before (Section 5.3.3), you should not simply confront employees with results, regardless of whether such results were generated by systems or by external agents, but should rather provide them with a number of options to choose from. On the other hand, keep in mind that too many options tend to confuse and paralyze people.

Value drivers hard to measure

We are well aware of the fact that these value drivers as well as their impact on shareholder value are hard to measure. Getting a grip on each of them could become a project in its own right. But although we said in our catalogue of requirements in Chapter 1, Section 1.4.3 that value drivers should always be observable or measurable, we never claimed that this would be a stroll in the park.

5.5 Implementation Scenario and Architecture with SAP HANA

Open and integrated processes

In the case of WoW, solution and data architecture are influenced predominantly by company policies and the company's current position (lean administration, open systems/standards, close cooperation with external partners, some decisions about outsourcing not yet made, etc.).

5.5.1 Implementation Scenario and Framework Architecture

Cloud scenario and SOAP standards

WoW has already outsourced quite a few processes to external partners and is now thinking about exchanging trip-related data with these partners. It currently favors SAP Rickshaw's cloud-based services, so putting its money on services hosted in a cloud and *SOAP*-related standards is logical for WoW. SOAP (Simple Object Access Protocol) is a networking protocol very common in service-oriented environments.

To reduce the amount of data that need to be transferred, migrating WoW's locally implemented SAP BW to SAP Rickshaw should also be taken into account. Consequently, following the cloud on SAP HANA scenario as the best possible implementation variant is an almost inevitable conclusion for WoW (see Figure 5.4). Also, running SAP BW (plus the SAP HANA database itself) and not just the analytical apps in the cloud would be a small difference compared to the standard cloud on SAP HANA scenario (as defined in Chapter 2, Figure 2.16).

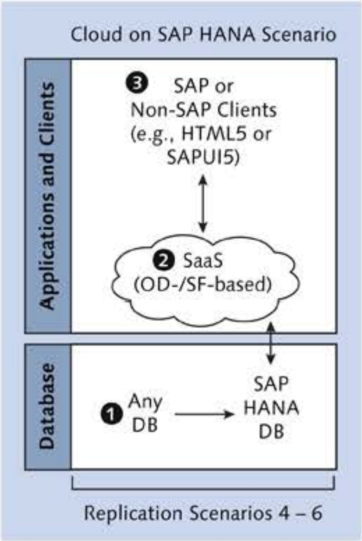

Cloud on SAP HANA Scenario
Applications and Clients
❸ SAP or Non-SAP Clients (e.g., HTML5 or SAPUI5)
❷ SaaS (OD-/SF-based)
Database
❶ Any DB
SAP HANA DB
Replication Scenarios 4 – 6

Figure 5.4 Traveling Implementation Scenario

Databases ❶

Internal as well as external and unstructured data

Some of the data WoW may want to analyze can be found within its SAP ERP systems. This data might be replicated from SAP Travel Management (TM)—for example, for hard travel costs—or corresponding cloud-based solutions or—like timesheets—from SAP Project System (PS) and other products that are part of SAP ERP or SAP Business Suite.

External sources of data

However, when detecting and modeling dependencies we are not only talking about classic databases used by WoW or its service providers. Flight weather reports and forecasts are not company-specific OLTP data; instead, such information would have to be acquired from websites via programming interfaces (application programming interfaces [APIs]). Furthermore, WoW might want to use certain unstructured data sources for its purposes. Breaking news to be found on news portals about imminent industrial action (or inaction, as it usually turns out to be) on the railways or in air traffic control could also be an asset.

[«]

Application Programming Interface

An *application programming interface* is code within a piece of computer software, the purpose of which is to make data available to other (external) applications. Many websites on the Internet offer access to their data via APIs as a paid or free service.

Products for Generating/Exploiting Data ❷

Core component of the solution

Data generation and data exploitation are two core functions that form the heart of WoW's envisaged solution. In this area, a multitude of products can be used; for the very first phase (analyzing potential benefits), we will need at least the following components:

- SAP Cloud for Travel and Expense
- SAP Data Services for geocoding using transaction `Geocoder` (if necessary)
- SAP PAL (functions: `CREATEDT` for C4.5 and `CREATEDTWITHCHAID` for CHAID and `LRREGRESSION` for multiple linear regression or `POLYNOMIALREGRESSION` for polynomic regression)
- SAP HANA, with an emphasis on the following functionalities:
  - Spatial processing (for geodata processing)
  - Text analysis (to use data published on news portals)
  - R scripts for further dependency-related algorithms that are not available within SAP PAL (examples: calculating covariances via `cov` and `cov2cor`, calculating correlations via `cor`, and testing the dependency of parameters using `summary` or the significance of correlations via `cor.test`)
- Other algorithms capable of drawing conclusions about a population from sample data—for example, algorithms from the realm of Bayesian statistics
- The Natural Language Toolkit (NLTK), a popular collection of programs and libraries with far-reaching functionalities for text analysis for the programming language Python

SAP PAL/SAP InfiniteInsight have got their limits

In Section 5.4.3, we mentioned that we are not necessarily dealing with linear relationships between parameters on a ratio level of measurement. The statistical algorithms required to analyze data on the other side of this boundary are usually not available in SAP PAL or SAP InfiniteInsight. From our perspective, you should therefore seriously consider using R and employing experts familiar with that language. Furthermore, you need to carefully think about which algorithms are mathematically suitable for processing your data.

Clients ❸

Enable user interaction

We have already pointed out that getting users involved and enabling them to express their personal preferences when making decisions about travel options is one of the keys to user acceptance and therefore to unlocking the benefit potential of new solutions. Hence, both the early warning system and solutions that holistically optimize business trips considered for later phases should not make autonomous decisions without user interaction.

Instead, acceptable alternatives—maybe the top ones for dimensions such as costs, time, and comfort—should be presented to the consultant in a clear and easy-to-understand format. Weighing the tradeoffs between, for example, a faster but less comfortable trip option and a nicer but more expensive one can be left to consultants; their choices can also be stored and used in the future to reflect, to a certain extent, their personal preferences when proposing alternatives to them.

We could also imagine a system that keeps track of the cheapest and the fastest alternatives for each business trip, distributing points to consultants whenever they are willing to accept comfort-related losses to realize cost advantages; at the end of the year, such points could be converted into bonus payments that share the company's cost savings with these thrifty consultants.

HTML5 versus Flash

Because all consultants with WoW have a company iPad and because Apple favors HTML5 over Flash for strategic considerations, WoWs's clients should be SAPUI5-based.

5.5.2 Data Architecture

Expanding the planning scenario

There are similarities between the data architecture needed for WoW and the one presented in Chapter 4; both architectures follow the same principles that are laid down in Chapter 4, Section 4.5.2, in "Further Considerations about Data Architecture," which is why we only focus on differences here:

- Because we are talking about a cloud-based, heterogeneous solution that crosses company boundaries, data logistics will primarily be based upon SOAP/XML standards. SAP BW, SAP Data Services, and SAP Event Stream Processor can still be used because all of them come with appropriate adapters/interfaces for incoming and outgoing data. They will, however, be used in slightly different ways.
- The structure of the data we are dealing with is more diverse than with RunFlat; we may therefore have to go a bit further in terms of atomization and semantic neutrality.
- Unlike the planning example, we are dealing with databases that are different in terms of their mathematical properties, such as their level of measurement (nominal, ordinal, or ratio). As statistical algorithms very often assume certain levels of measurement with the data they are processing, we need to keep track of such data properties. Furthermore, we need to bear in mind that some data that are continuous by nature might still be delivered as discrete data by the solutions or services we use. One example of this is the visibility in METARs that always comes in steps of 100 meters, with no difference being made between visibilities above 10 kilometers; the latter are all reported as "10 km or more."

Object types in SAP HANA

In Chapter 4, we focused on the wider picture—that is, the links between various layers within our implementation scenario. In this case study, we are going to take a closer look at the layer's architecture in an SAP HANA environment; we will also use this opportunity to explain a couple of technical terms that were mentioned in Chapter 4, Section 4.5.2: attribute view, analytic view, and calculation view (see also Chapter 4, Figure 4.16).

In principle, each and every data architecture that is implemented with SAP HANA will use four different object types to model layers, along with procedures that sit between these layers and (mostly) represent vertical data flows:

- **Persistent database tables**
  Within SAP HANA, tables serve exactly the same purposes as in classic database management systems. They are used to persistently store data for later retrieval, keeping them ready for further processing. Compared to classic databases, there are three differences with regards to tables:
  - In an in-memory database, persistency means that data are stored permanently but still in memory only (apart from the backups that may exist on SSDs or hard drives). The term *persistent* is therefore to be interpreted differently than it is for traditional databases.
  - Within SAP HANA, tables can be classic, row-based ones or stored in a columnar format, the latter being more common with in-memory databases and more suitable for reporting purposes.
  - As well as tables, SAP HANA also supports a number of views. These views are similar to those in a classic database. Views are often used to combine *hot* (up-to-date or recent) and *cold* (historical) data. Views within SAP HANA can not only be based on joins or SQL statements (as with classic databases) but also represent the output of complex—for example, statistical—routines.
- **Attribute views as joins**
  Attribute views come into existence by linking a couple of persistent and virtual tables via joins, by applying filters to these data—for example, selecting the data for just one country—and by processing these data via very simple formulas and operations—for example, isolating the year from a data record.

  Attribute views often represent master data; tables combined within an attribute view don't usually contain transaction data (called *facts* in a data warehousing environment).
- **Analytic views as OLAP cubes**
  Analytic views are created by combining attribute views that contain master data with tables that contain transaction data. Their role is sim-

ilar to that of OLAP cubes in classic data warehouses, with attribute views representing dimensions and tables containing transaction data acting as fact tables. When building analytic views, as with attribute views, you can select data and (to a certain) extent also process these data.

- **Calculation views as MultiProviders/queries**
  Compared to a classic SAP BW system, calculation views are some kind of cross between MultiProviders, InfoSets, and queries. Very much like MultiProviders or InfoSets, they make data available from different analytic views (OLAP cubes); furthermore, calculation views let you add a semantic layer called a *projection*. Projections can be used to translate technical structures into structures that are more familiar to the business.

  In terms of operations and algorithms, calculation views are superior to MultiProviders and InfoSets as well as to queries. In SAP BW, there are no transformations between InfoCubes and MultiProviders; if you have to dramatically modify data before reporting them from a MultiProvider, you will need at least one additional persistent layer for that.

  Calculation views, on the other hand, can be set up not only using analytic views but also based upon procedures. You are therefore able to implement highly complicated algorithms without having to create inflexible, persistent layers that would not be needed otherwise. Procedures are one of the mainstays of SAP HANA. They eliminate the need for layers that are merely around for technical reasons, and due to the quantity and variety of functions that can be used within them they can do a lot more than transformations, helping you migrate computations from the application to the database layer.

Procedures [«]

In SAP HANA, *procedures* are reusable building blocks for data processing. Calculation views are often based upon procedures.

In principle, procedures serve the same purpose as transformations in SAP BW or in SAP Data Services. Procedures, however, have a much wider scope, especially when it comes to statistical functionalities. Also keep the following in mind to avoid confusions:

- Procedures within SAP HANA are (mostly) used in vertical data flows.
- Transformations in SAP BW can be used with data acquisition (horizontal data flows) and with data processing (vertical data flows).
- Transformations in SAP Data Services are (usually) applied within horizontal data flows.

Procedures can analyze as well as modify data. From a developer's perspective, procedures can be devised in SQLScript, L, and R; they can call functions from SAP PAL (such as `CREATEDTWITHCHAID` to create/train a decision tree). From the introduction of RDL in 2013 onwards, procedures not only can be written manually but, like other objects in SAP HANA, also generated automatically.

SAP HANA part of the data architecture

In our example—that is, to answer the question about whether flight delays can be forecast on the basis of weather data—the object types described previously could be used as indicated in Figure 5.5.

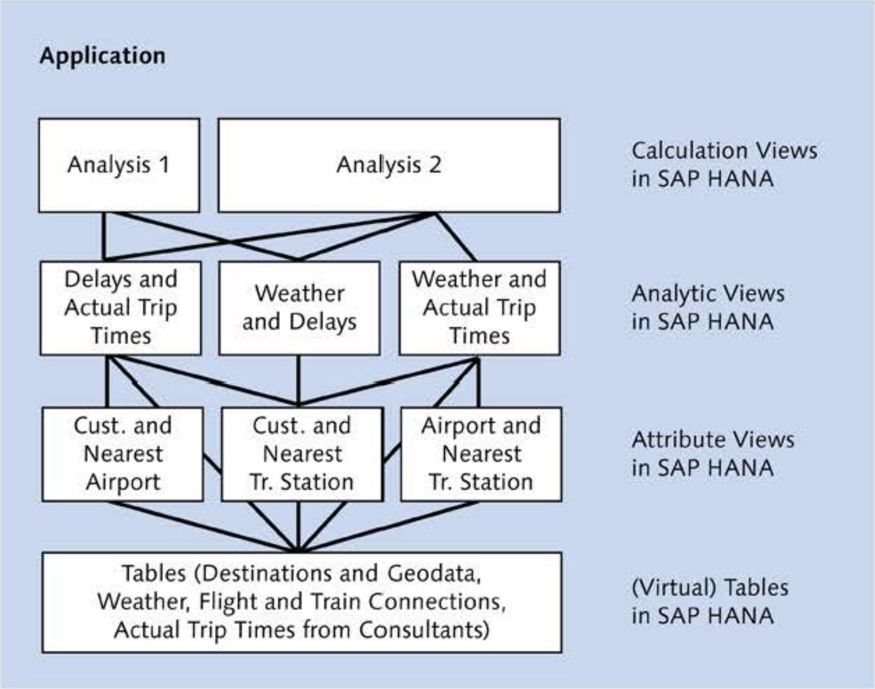

Application
Analysis 1
Analysis 2
Calculation Views in SAP HANA
Delays and Actual Trip Times
Weather and Delays
Weather and Actual Trip Times
Analytic Views in SAP HANA
Cust. and Nearest Airport
Cust. and Nearest Tr. Station
Airport and Nearest Tr. Station
Attribute Views in SAP HANA
Tables (Destinations and Geodata, Weather, Flight and Train Connections, Actual Trip Times from Consultants)
(Virtual) Tables in SAP HANA

Figure 5.5 Traveling Data Architecture (Detail)

Let's take a closer look at this schematic diagram:

- **Application areas for tables**
  You could create three tables that contain trip destinations (airports, railway stations, and customers) and their geodata, three tables to record weather reports and/or forecasts (one containing weather phenomena, such as hail or fog, another one with visibilities in general and on specific runways, and a third one for wind speed), and tables for flight and train connections and their scheduled and actual departure and arrival times.

  In addition, there could be other tables that contain data from the consultants' time reports in SAP PS that tell you more about actual travel times; ideally, these data should be split by segments of the trip. Depending on where the data come from, we may be talking about physical or virtual tables; if the data are already in SAP HANA due to SAP Business Suite running on the same database, then views are another option to separate source data from your app's data without holding data redundantly.

- **Examples for attribute views and analytic views**
  A number of attribute views could be used to assign nearby airports or railway stations (hubs) to customers. That list could be kept up-to-date automatically, using the most current customer information.

  Nearby hubs can be determined using the distance as the crow flies, the so-called *orthodromic* (or *great-circle*) *distance*; unlike more simple approaches, the great-circle distance takes into account that the earth is not flat. To calculate the orthodromic distance, you will need trigonometric functions, such as sines, cosines, and so on; the Formula Editor for analytic views does not deliver these.

  There is, however, an even better way to solve the problem. Instead of thinking about how to determine orthodromes, you could simply use SAP's extension for processing spatial data (method `ST_Distance`). One could consider calculating the distances or determining the nearest hub via a separate data flow and then merging it with an attribute view for customers.

Regardless of how the data contained in attribute views are meant to be determined, there could be three analytic views based upon these attribute views:

- In Figure 5.5, the first analytic view is used to integrate delays and actual trip durations.
- The second one contains weather and delay data.
- The third one comprises weather data and actual trip durations.

All three analytic views will also need to contain destinations (airports, railway stations, and customers) plus their geodata to subsequently, for example, enable geodata-based destination grouping for analytic purposes.

- **Examples for procedures and calculation views**
  The calculation view called ANALYSIS 1 could, as an example, contain the correlation between visibility and delays calculated in a procedure that uses R's `chisq.test`. As different correlations could be calculated using different time lags (see the "Detecting Dependencies" section in Section 5.4.3), there will be more than one result for this.

  Figure 5.6 shows a sample procedure in SAP HANA Studio's SQL Console. This sample procedure contains the following elements:

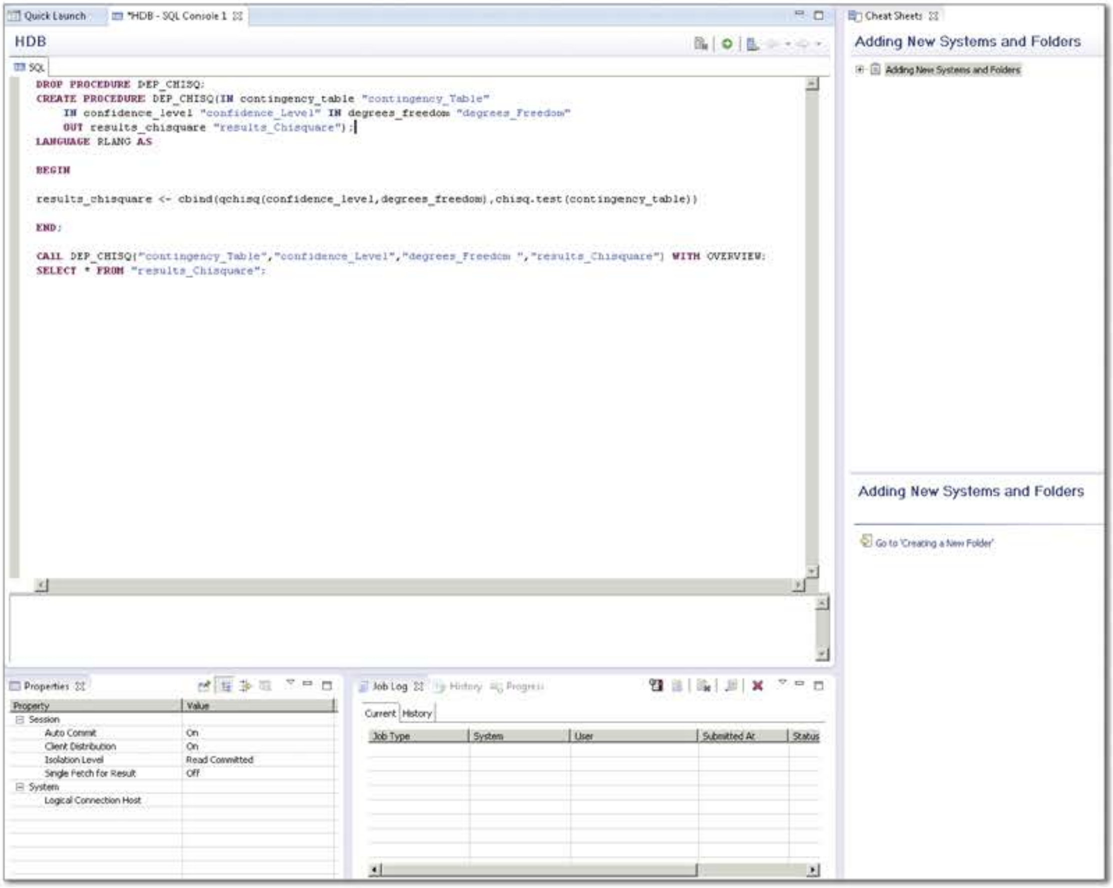
  - **`DROP PROCEDURE`** (SQLScript)
    Deletes procedure `DEP_CHISQ` (in case it already exists).
  - **`CREATE PROCEDURE`** (SQLScript)
    Creates a procedure called `DEP_CHISQ`, also defining this procedure's input and output parameters; as input parameters, we have allowed for a contingency table, a confidence level, and the degrees of freedom. If a confidence table contains *r* rows and *c* columns, then the degrees of freedom can be calculated as *(r – 1) * (c – 1)*; the output of the procedure will be a table containing its results. When defining input and output parameters, we are linking these to corresponding objects. The contents of these objects are later handed over to R.
  - **`LANGUAGE`** (SQLScript)
    Informs the system that the procedure is going to be written in R (`RLANG`). Note that no R server had been configured for the system

that we used to take the screenshot in Figure 5.6. Therefore, it would not have been able to execute this procedure.

- **BEGIN…END** (SQLScript)
  Defines start and end of the code written in R.
- **qchisq** (R)
  Uses the confidence level and the degrees of freedom provided to calculate a threshold for χ^2. If the χ^2 of the data contained in the contingency table is above this threshold, then one should refuse the null hypothesis, saying that the data in the rows and column of the contingency table are independent; instead, one should assume that there is some kind of (statistical, not causal!) dependency between them.
- **chisq.test** (R)
  Executes a chi-squared test for independence for the data contained in the contingency table, returning the results in an R-specific format.
- **cbind** (R)
  Converts the preceding R-specific format into a matrix. The first column of this matrix will contain the threshold calculated by qchisq within each row, and the second column will deliver the results of the test. The purpose of this function is to convert the data into a form suitable for further processing in SAP HANA.
- **CALL** (SQLScript)
  Calls the procedure defined previously, handing over the respective input parameters; the option WITH OVERVIEW ensures that the results will be physically stored afterwards.
- **SELECT** (SQLScript)
  Displays the contents of the table results_Chisquare.

For more information about using R within SAP HANA procedures, refer to the additional resources listed in the book's online appendix.

The calculation view called Analysis 2 in Figure 5.5 could use SAP PAL's function LRREGRESSION to develop a model to forecast what kind of average delay would result from a certain visibility and wind speed. Similar to the procedure underneath Analysis 1, you could call the function as often as you like, letting it compute results for all kinds of time lags between weather observations and delays.



```
DROP PROCEDURE DEP_CHISQ;
CREATE PROCEDURE DEP_CHISQ(IN contingency_table "contingency_Table"
    IN confidence_level "confidence_Level" IN degrees_freedom "degrees_Freedom"
    OUT results_chisquare "results_Chisquare");
LANGUAGE RLANG AS

BEGIN

results_chisquare <- cbind(qchisq(confidence_level,degrees_freedom),chisq.test(contingency_table))

END;

CALL DEP_CHISQ("contingency_Table","confidence_Level","degrees_Freedom ","results_Chisquare") WITH OVERVIEW;
SELECT * FROM "results_Chisquare";
```

Property	Value
Session	
Auto Commit	On
Client Distribution	On
Isolation Level	Read Committed
Single Fetch for Result	Off
System	
Logical Connection Host	

Figure 5.6 Embedding R Code within SQLScript in SAP HANA Studio

[»]

How sure do you want to be?

Confidence Level

In simple terms, the *confidence level*, sometimes also called *confidence coefficient*, specifies how likely it is that a statement made on the basis of a sample will not only true for the sample but also for the complete population it has been drawn from. In mathematical terms, this definition is not 100% correct, but to keep things clear we won't bother with the fine print here!

[«]

Confidence Level

When executing a chi-squared test for independence, as shown in Figure 5.6, the confidence level would be applied as follows:

- We use function `qchisq` to calculate a threshold for χ^2 determined by a confidence level of, let's say, 95% and the degrees of freedom for our contingency table.
- Subsequently, we use `chisq.test` to calculate χ^2 for the data in the contingency table.
- If the χ^2 calculated via the contingency table via `chisq.test` is higher than the one calculated via `qchisq`, then we can refuse the hypothesis that the random variable represented by the rows of the contingency table is independent from the random variable represented by its columns; the probability that you are right is 95%.

Two additional remarks:

- The *chi-squared test for independence* should not be confused with the chi-squared test for goodness of fit first mentioned in Chapter 4, Section 4.4.3. The chi-squared test for independence checks whether two random variables seem to be statistically independent and can also be used with non-ratio random variables.
- `chisq.test` can also do the whole job; you don't really need `qchisq`. We chose a slightly different approach to help you understand what is going on and why.

The confidence level's counterpart is the *level of significance*. The level of significance, again simplifying a bit, stands for the probability of making a mistake when refusing the null hypothesis. In our example, making a mistake would mean rejecting the idea that the data are independent even though they are.

With a given level of significance α, the confidence level can be calculated as $1 - \alpha$. By definition, level of significance and confidence level therefore always add up to 100%.

With statistical testing, you often work with so-called two-sided tests. If the level of significance α was 5% in the coding example shown in Figure 5.6, then this would mean that function `qchisq` would have to be fed with a confidence level (input parameter `confidence_level`) of $1 - 5\% / 2 = 97.5\% = 0.975$ instead of $1 - 5\% = 95\% = 0.95$. We don't want to delve into this topic any further and only mentioned it to appease the mathematical geeks among you.

[»]

Contingency Table

A *contingency table* (also called *cross-tabulation* or *cross-tab*) is a table that describes the frequency of occurrence for different combinations of values that two random variables could adopt.

Let's say you had a sample of 1,000 women for which you had collected the hair color of mothers and daughters. The relevant categories are blond, brown, red, and black. In the form of a contingency table, the data from your sample would look similar to Table 5.1.

Mother/Daughter	Blond	Brown	Red	Black	*Total*
Blonde	42	57	50	102	*251*
Brown	60	29	85	101	*275*
Red	21	18	7	24	*70*
Black	90	114	22	178	*404*
Total	*213*	*218*	*164*	*405*	*1,000*

Table 5.1 Example of a Contingency Table

In a nutshell, this table shows how many blond mothers have blond daughters, how many brown-haired mothers have blond daughters, and so on. To be processable via R's `chisq.test` function, a contingency table also needs to contain totals in its rows and columns, which is why we have added these in Table 5.1.

Limitations of our sample data model

Despite the relatively high level of detail, our description of WoW's data architecture does not (by a long shot) represent a technically mature or complete data model; this also applies to all other case study chapters. What we are trying to do is give you some hints to help you design some possible approaches. Please note that the procedure shown in Figure 5.6 is also not executable as is. It lacks the steps needed to create its input and output tables or a loop or, better, a recursion that enables multiple executions with different time lags.

A more abstract approach

Although you may find some of the considerations in this chapter fairly abstract, we would still strive for an even higher level of abstraction on a real project. This would, for example, mean modeling random variables, such as wind speed or type/intensity of precipitation, as something like TIME SERIES 1 and TIME SERIES 2 and then handing them over to R as vectors or matrixes without saying anything about their business content. R could then count the values within these time series, using function `freq`, for use as input to a contingency table.

Abstraction is flexibility

Users obviously don't want to and don't have to deal with such extremely high levels of abstraction. There would be a number of semantic layers between the procedures we are talking about here and the ones that the clients' end users would be working with, so they wouldn't even know what's going on under the hood.

But why on earth are we so keen on abstraction, then? Over the years, WoW may note that in the age of computerized airplanes and landings visibility has become insignificant for forecasting delays. Instead, the runway length of an airport's longest runway has become an important factor. Applying a high level of abstraction means that WoW's forecasting tools could just replace, add, or remove time series without opening a can of worms and triggering hundreds of follow-on changes to downstream objects when doing so. Furthermore, our procedures can be designed to handle whatever time series they find without having to care what their name is or which data these time series contain. This also takes us a tiny bit closer to an inductive approach. It does, however, also make great demands on architects. We will get back to abstraction a few more times, showing you even more ways of taking the flexibility of your data models to a new level.

Insights from this case study

In Walk-on-Water's case study, we tried to communicate the following insights:

- How to detect dependencies using SAP HANA
- How to use SAP HANA to develop models inductively instead of deductively
- How a high level of abstraction can contribute to more flexible, more inductive solutions
- What certain technical terms in SAP HANA mean

Next case study

Going back to the options discussed earlier in the chapter about time lags, we indicated that intervals of measurement play an important role in statistical analysis. In the next case study, we will look at time from a different perspective. By treating time as just another parameter that can be used to classify customers and segment markets, we hope to open your eyes to whole new horizons for analysis.

Order is for idiots; genius can handle chaos.

Attributed to Albert Einstein

6 Decision-Oriented Data Models

The weather had calmed down a bit. A winter gale had been raging for four days and nights, driving alternately heavy rain then hailstones across the land over and over again. This morning, however, the sky had suddenly turned bright blue. Derek had decided to make good use of the sunny spell and take a walk around the headland West of Dornie, and after about two hours he had arrived at a gravel beach. As usual for this time of the year, the sun had been low but still warmed him a wee bit. Looking for a good photo, he had strolled around between flotsam and jetsam and the remains of fishing boats, and finally fixed on a shack that was cuddled up against an embankment.

Figure 6.1 Shack on a Gravel Beach near Dornie, Wester Ross, Scotland

Even the word "shack" was a bit generous. The rotting hut was so small that he couldn't stretch out horizontally on its rough floor. Nevertheless, its architect had provided a rickety pedestrian bridge to make it accessible (well, just) *and included a stove. A lopsided chimney still protruded through its gable, but sadly the domicile seemed to have been abandoned for quite a while. Its walls and roof didn't look watertight anymore, as far as Derek could tell, assuming they had ever been. The function of the building was a bit of a mystery to him, but whatever purpose it may once have served, it seemed to have been built in a couple of stages. Timber blockings, metal, chipboard, and fishing nets added up to a jumble of construction materials. The pedestrian bridge might have been added a bit later on, as well as the small flight of stairs leading to a shelter open on one side behind the hut. Altogether, the ensemble looked a bit makeshift, duct-taped together a while ago—but it certainly had been built there for a reason.*

It was very much like data architecture with Sell-Your-Soul (SYS), one of Derek's major customers. Two years ago, it had been planning to go public; Derek had been working on a project that was supposed to help it fulfill reporting requirements and crosschecking differences between a couple of reporting solutions. While doing so, Derek had discovered that these solutions were based upon totally different data structures in their respective source systems.

SYS was trying to reconcile all these anomalies by using pretty elaborate and complex consolidation and harmonization routines in SAP BW. This only worked, however, as long as the structures within the systems delivering all the underlying transactional data remained reasonably stable; sadly, that never happened for more than a couple of days at a time. In reality, the source systems were continuously coming up with new combinations and permutations of data that the programmers hadn't imagined at the outset. As a result of this kludge, the routines within the data warehouse had grown into an impermeable thicket of many individual layers. Each of these layers was bending the data it received from below; regrettably, this often happened without the programmers having the faintest idea about what the preceding processes were doing and why.

In the end, Derek had given up, recommending to SYS that it should redesign and rebuild its data and reporting architectures from scratch. Just as for the shack in front of him, the day on which a storm would bring about the downfall of a skeleton that consisted of nothing but patchwork and temporary fixes didn't seem to be far away for SYS.

Challenges for data governance

When engaging in big data, we are dealing with ever-increasing volumes of data, the structure of which changes faster and faster. This multiplies the problems in terms of data governance and data architecture, things that most companies are struggling with already.

In this chapter, we will first define what we mean by data governance and why this topic is becoming more critical against the background of big data. Using another fictitious scenario, we are going to explore the kind of damage that data model inconsistencies and data models that have been allowed to grow organically over the years can cause. Initially, we will look at different types of inconsistencies; for further analysis, we will then primarily focus not only on differences between various data models but also on discrepancies between these models and the reality they are meant to reflect.

When trying to avoid inconsistencies, getting things right from the start is certainly better than trying to fix them later. Which is where SAP HANA and SAP PAL enter the scene: Data flows can also be seen as a sequence of minor decisions finally leading to a statement regarding key performance indicators or value drivers. Both SAP PAL and R come with powerful algorithms to build decision trees. The performance of SAP HANA makes it possible to calculate many alternative options for the same data flow, in the end picking the one that works best for you.

As in our previous chapters, we will also establish the relationship between our scenario and SAP Solution Explorer, discussing functional aspects and potential benefits and tools. The key aspect reviewed in this chapter is how data flows can be developed at least semiautomatically.

6.1 Data Governance: Rhetoric and Reality

Data governance is in vogue

In addition to buzzwords like *business intelligence* and *big data*, terms like *data governance* or *data stewardship* have been in vogue in IT during the last couple of years. A few major companies are now not only treating themselves to CEOs (Chief Executive Officers) and CIOs (Chief Information Officers) but also CDOs (Chief Data Officers). Meanwhile, each and every manufacturer of products for data generation, management, data storage, or exploitation is swearing to the high heavens that its solutions can not only turn water into wine but also solve all data governance–related problems once and for all.

It's not by chance that the importance of data governance has grown in parallel to data warehouses becoming bigger and bigger. In February 2014, SAP secured itself an entry into the *Guinness Book of World Records* by fabricating a data monster comprising 12.1 petabytes of data in Santa Clara, California. Just to put these numbers into perspective, a petabyte equals a quadrillion (1,000,000,000,000,000) bytes or the amount of text in 250 million bibles which would—if put side by side—fill a shelf that is more than 6,000 miles wide (the distance between Frankfurt in Germany and Ho Chi Minh City in Vietnam). Unlike the bibles example, SAP's giant data warehouse (hopefully) doesn't just contain endless repeated instances of the same, and therefore redundant, data. We are dealing with the equivalent of some 368 billion *different* printed pages.

Management liable for consistency

Facing these magnitudes, one may rightfully ask exactly who—except maybe Hasso Plattner (cofounder of SAP) and the Good Lord—would be able to tell *which* data are stored *where* within that data warehouse? Furthermore, getting back to our analogy about books on a shelf from Germany to Vietnam, which CFO would like to bet his last million dollars (or his freedom) that the information in the section near Tashkent (Uzbekistan) is identical to that in a jungle some 2,500 miles farther southwest? How do you know that local primates haven't shuffled the pages?

6.1.1 What Is Data Governance?

Guidelines for data management

Questions like these are robbing CEOs and CFOs of their sleep, which is why a lot of consultants are getting paid lots of money to wrack their brains over data governance.

[»]

Data Governance

The term data governance has not yet been defined unambiguously. Often, data governance incorporates all measures, guidelines, and policies related to administering all data that exist in an organization. Among other things, data governance is meant to tackle data quality, data model consistency, data usage, data lineage, data transparency, data safety and security, and data-related legal requirements.

Regulations versus technical solutions

Data governance problems can often be addressed by technical or organizational measures, but there are also tools to help. We will focus on the tools, but let us first take a more detailed look at the challenges.

- **Data quality**

  In many cases, only people are able to judge the quality of data. A data-governance solution might dial a customer's number via a computer-telephony integration (CTI) system and detect whether somebody is picking up the phone on the other end of the line—but it can't tell whether the person who answered is the one recorded in your master data.

  Products for data management can compare different sources of data, such as your own database and public telephone directories. They can also help you periodically and systematically draw samples for manual quality checks. The results of such manual quality assurance controls can be aggregated, stored, and tracked automatically.

- **Data model consistency**

  In many organizations, armies of internal and external staff do practically nothing but reconcile numbers in different reports, explaining deviations to managers. Understandably, the board gets a bit nervous if a profit center is showing a profit of $100 million in one system and the same profit center is in the red on a management dashboard.

  In theory, this problem can be solved via a data warehouse with clearly defined and clean single points of truth (SPOTs), sometimes also called single spots of truth (SSOTs). A profit center's result should only ever be calculated once at the maximum granularity needed (that is, for example, per profit center or per profit center group) and then stored centrally precisely at this SPOT. From there, the number will then be distributed to various recipients and/or aggregated on the way. If there are inconsistencies between aggregated values, then all you need to do is track them back to the SPOT, checking the processing in all affected data flows from there on.

  If you had such structures (and if your data flows were simple and transparent), then explaining differences would be routine. The staff members that spend precious time analyzing discrepancies could engage in ferreting out data flows or processing steps that are redundant with regards to their contents. Detecting such duplicates automatically would be even more elegant.

- **Data usage**
  Practically all products for data generation and exploitation come with some kind of statistics about report usage, but such statistics don't tell you how often source data are being used. In addition to the question of whether certain reports are benchwarmers, it may also be exciting to learn whether two-thirds of your hardware and operating costs are flowing into keeping data that no one will ever use or whether your highly paid people in IT are maintaining data flows that empty into unloved and unwanted reports.
- **Data lineage**
  Products for data management, such as SAP Information Steward, often include functions for checking data provenance (also called data lineage or data pedigree). Ideally, by working across system boundaries, such functions are used to capture and display connections among databases, thereby telling you which sources are feeding data into your reports no matter how labyrinthine the data flows between source and target may be. Being able to answer such questions could also be of interest in the context of data security issues.
- **Data transparency**
  Once you have collected metadata about an end-to-end data flow, you can usually ask the system to graphically show that data flow in all its beauty, starting with whatever processing step you like. This makes it a lot easier for us humans to understand what is going on inside the guts of a system.
- **Data safety and security**
  There are loads of tools to help you protect your data from getting lost (mirroring them at locations that are geographically apart from each other, backing them up, and so on). When defending against data loss, data theft, and unauthorized access, you may want to go beyond static rules and authorization profiles, instead implementing some of the solutions and tools described in Chapter 10. In general, there is no difference between fighting energy theft and data abuse. In addition,

you may want to deploy products from SAP's portfolio for governance, risk, and compliance (GRC).

- **Data-related legal requirements**
  Legal rules and regulations that are related to reporting or the traceability of data within reports are often more long-lived than internal reporting requirements. The bad news is that you are not free to design related data flows according to your personal liking; on the other hand, the related requirements won't change as often as the ones your managers come up with. Data flows or structures that only exist to fulfill legal requirements are therefore ignored in this chapter.

Data owners for metadata

Like all kinds of data, metadata repositories also need a bit of tender loving care from time to time. They need to be configured, managed, and maintained. Furthermore, somebody will have to look at all the great graphical displays that explain where data come from, where they go to, and how they are processed in between. Then, if your metadata repository detects problems related to data governance, it will get confused and be in need of somebody to talk to (or to alert), which is what CDOs, data owners, and data stewards get paid for.

[«]

Data Stewardship

The idea of data stewardship is based upon the concept that there is an owner for each and every bit of metadata. The data owner makes sure that metadata comply with the organization's principles for data governance; he also ensures they don't represent a theoretical ideal but instead correspond to what is actually implemented here and now.

In some contexts—for example, in this book—there is a difference between data owners and data stewards. A *data owner* is somebody who has got the power of decision about data or metadata; a *data steward* is the person doing the preliminary work for the data owner, supplying her or him with information. The boundaries between both roles are blurred.

6.1.2 Challenge: Data Volume, Speed, Agility

The problem of rapidly expanding data volumes is just one of three reasons that questions of data model consistency have gained importance recently and have done so in step with big data.

Data Volume/Heterogeneity

Heterogeneous and unstructured data

The more data there are, the more heterogeneous they become—and the more versatile their sources are, the more difficult safeguarding data model consistency across all layers of a data flow will become. Even worse, a growing percentage of these data is unstructured (*unstructured data* are data that do not have a clearly defined schema or data model; a typical example for this is free text). Furthermore, data are used in a growing number of ways, many of which are neither known nor foreseeable when the data are collected.

Speed

Processing gigantic, heterogeneous amounts of data in real time is a colossal challenge on its own. More awesome from a modeling perspective is the fact that the composition of these data are anything but stable in the long run. All these data provide opportunities organizations couldn't even have dreamed of a couple of years ago (there are nightmarish dangers as well, but these are not a subject of this book). Using the data's full potential does, however, mean that you have to adapt both algorithms and data flows in ever-shorter intervals.

Quantity tipping over to quality

The iPhone 6, for example, can record videos in full HD (1,920 by 1,080 pixels) at a rate of 60 frames per second; without question, smartphones will sooner or later come with the ability to record UHDV (7,680 by 4,320 pixels) at 120 frames per second and in 3-D. This would generate 128 times more data than full HD. But if you also need to process UHDV, a hardware or software platform that is 128 times as fast as before and can therefore handle the higher data throughput, won't do. You'll also need algorithms that can realize the additional benefits of higher resolution or, for example, 3-D recording. Unlike 2-D, 3-D could provide you with information about the distances between objects, so this will not only be a quantitative but also a qualitative jump.

Agility

Changing business requirements

The fact that we are dealing with ever more heterogeneous data steaming down the track at increasing speed and frequency is probably already driving the odd data owner into despair. Also, in parallel, the number of analyses and reports derived from these data is growing as well. Once your users have experienced an in-memory database, they will ask for more. A world in which reports are centrally designed in a cozy little corner of the universe, where they remain unchanged for years, is a thing of the past; this might not yet be the case with all businesses, but the days of those who haven't changed their mind sets might be numbered. Today, the number of requested reports is expanding at least as fast as the amount of data they are based upon.

Virtual structures don't know much about history

As a consequence, building blocks of data flows are now often set up as virtual objects (without their own persistent databases). Such virtual objects are then continuously and decentrally adapted to changing needs; in many cases this happens without any kind of formal change request or documentation. Data stewards are therefore in danger of not only losing the race in terms of documenting the data's origin, but also with regard to recording data usage and ensuring the traceability of reported numbers.

6.1.3 Gap between Data and Metadata

No budget for documentation

Most IT projects we have worked on in the course of our careers have run out of budget, time, people, or on occasion all of these while documenting their solutions. This is one reason that we prefer declarative, self-explanatory languages that don't need much verbiage to imperative and procedural tools. However, even with declarative programming, ever-shorter development cycles mean that you will sooner or later run into the same problem; while the documentation is being written the system may still be under review or even undergoing a new review and subject to further change.

Writing documentation for a system is like trying to freeze the details of a pond's moving surface by painting it; before you even dip the brush into the paint pot the picture has altered entirely. With data (the surface of the pond) and metadata (your painting), the problem is just as tricky, if not more so.

Automation needed

To reflect reality, metadata have to be intrinsically tied to the actual corresponding objects within your system; whenever these objects are changed, metadata have to be adapted without noticeable delay and automatically—just like in SAP ERP's classic ABAP Dictionary (Transaction SE11). However, as with SAP's ABAP Dictionary, the real challenge is linking processing or program logic with functional or business requirements. The fact that a certain function module exists and that it uses two specific tables says nothing about what exactly it is supposed to do and which other objects are affected by it. It does not define what the purpose of a function module in the context of certain business processes might be nor which function modules have to be adapted after a business process has been transformed. Furthermore, the ABAP Dictionary's reach is limited to certain SAP products, such as SAP Business Suite.

[+]

Metadata Repositories as a Prerequisite

We defined the term metadata repositories in Chapter 2, Section 2.1.3. Unfortunately, collecting metadata that are accurate and up-to-date is anything but trivial. Although SAP BW, for example, is able to export its metadata in a standardized format (XML), metadata from a data warehouse alone won't suffice. A data warehouse sits at the end of the food chain; you therefore also need metadata from the respective products for data generation (other SAP or non-SAP products).

The practical use of any metadata repository therefore depends on how easy it is for this product to seamlessly connect to the other solutions in your organization and how smoothly their metadata can be collected. If your environment is SAP-based, then SAP's enterprise information management (EIM) solutions will obviously have a home-field advantage.

Throughout the rest of this chapter, we will assume that you have already implemented appropriate products for data management and that the metadata you will need for what we are proposing are at your disposal. We will also ignore topics like data quality, data safety and security, and data-related legal requirements. Instead, we want to address how big data and SAP HANA can help you improve data model consistency. Other data-governance topics (data usage/lineage/transparency) addressed in Section 6.1.1 are of a supportive nature for data model consistency.

6.2 Scenario: Determining Trade Margins in Retail

SYS: a specialty retailer for consumer electronics

Sell-Your-Soul AG (SYS), the company mentioned in our introductory story, is one of the world's leading retail chains for consumer electronics. The company is based in the Swiss town of Zug, a well-known Swiss tax haven. SYS has more than 800 stores in 20 countries, selling some 65,000 different articles (things for sale). Its 73,000 employees generate annual revenues of more than 30 billion Swiss Francs (CHF).

Trade margin as value driver

One of the most important strategic objectives for SYS is to increase its earning power in individual countries or locations, currently measured via the respective trade margin of the previous calendar month. The *trade margin* is defined as the difference between net sales price (after bonuses and other costs or sales are deducted) and cost of acquisition (purchase price plus other add-on costs) expressed as a percentage of the advertised sales price. The trade margin serves as the basis for decisions related to marketing budgets, investments, and even expanding in or withdrawing from a country.

Inconsistencies with costs of acquisition

When calculating the trade margin, SYS faces the same problem again and again; costs of acquisition are also used for stock valuation and are therefore subject to legal rules and regulations in some countries. As a consequence, costs of acquisition and the trade margins based upon them are often not comparable. Furthermore, a couple of countries calculate so-called shadow costs of acquisition for their own reporting; this leads to an even greater variety of numbers.

Although SYS uses SAP ERP everywhere, it is not unusual to find that the ERP system in one subsidiary coughs up a trade margin of 11% while SAP BW in the very same location insists that the margin is only 9%. As a general rule, SYS expects a margin of about 10%; articles that underperform are usually removed from the assortment. Hence, from the management's perspective, 9% and 11% are worlds apart.

Reconciliation efforts are high

The data warehouse calculates margins via highly complex reconciliation routines that were developed by a team of Indian consultants some ten years ago. At the time, the consultants were flown in for the project's duration, but despite the fact that they came at a much lower price than domestic experts the project's budget was still exceeded, and so the project was stopped during the documentation phase. All consultants left the country long ago and are not available or even reachable any longer.

Some country subsidiaries therefore employ up to 10 programmers, trying every month anew to explain the substantial differences among key figures that should theoretically be the same in all systems. The only way to do this is to debug the code of the reconciliation routines—that is, to try to figure out whether the logic in the code is doing the right thing incorrectly, if it's doing the wrong thing correctly, or if it's just plain out-of-date and needs rewriting.

On top of that, *fixing projects*, which are supposed to redocument what was once implemented, have been set up in a number of locations. Unfortunately, the responsibilities for reconciliation and redocumentation lie with different suppliers.

It's not hard to see why cross-country comparisons can't be done. Article groups are defined differently in each country. Even what-if analyses in terms of regrouping articles are bordering on the impossible in practice. At the moment, there is no way for the group's headquarters in Switzerland to compare trade margins among different regions or to recalculate trade margins on the basis of alternative product groupings.

6.3 Inconsistent Data Models: Costs, Risks, and Opportunities

Costs in IT

The inconsistencies between the data models and the systems in individual countries have become a paralyzing handicap for SYS. IT costs related to analyzing and reconciling numbers have increased exponentially. According to the CIOs in some countries, such costs account for some 60% of total personnel costs in IT—and because local CIOs are aware of the fact that top management knows about the issue as well, it also gives them an easy way to explain away overspending.

Organization unmanageable

Another consequence is, however, a lot more serious. Due to all these different algorithms and article groupings in various systems and countries, the group has become more or less unmanageable. The head of controlling at Zug has lost all trust in the numbers delivered by the subsidiaries and has built up his own reconciliation team intended to recalculate the trade margins for each country based upon his own specifications; this has added even more costs to the payroll.

Another result of this emergency measure is that there now is a morass of Microsoft Excel workbooks and VBA routines in central controlling; the formulas within these workbooks are extremely complicated and undocumented, and the numbers generated by them add new trade margin versions and thus more confusion, leading to even higher reconciliation efforts in IT.

Stock market launch canceled

In May of the previous year, SYS contemplated floating the company on the US stock market. Then the CFO was given the job of making the necessary preparations, which included reviewing reporting structures. Under the Sarbanes-Oxley Act (SOX), the CEO and the CFO were required to confirm the accuracy of their statements by statutory declaration: swearing an oath in writing. Bearing in mind that he (the CFO) could have become subject to criminal prosecution in the United States if the numbers were wrong and knowing that even *he* didn't trust these numbers, it's not surprising that the CFO eventually suffered a nervous breakdown, ending up in a private psychiatric clinic in Switzerland. Prudently, SYS canceled its stock market launch indefinitely due to "unfavorable conditions on capital markets."

6.3.1 Problem: Different Formulas

Example: freight costs

Let's use an example to illustrate the problem of diverging formulas and algorithms. For both SYS in the United States and SYS in Japan, freight costs are added when determining costs of acquisition. In both countries, costs of acquisition for stock-valuation purposes are calculated within SAP ERP (with the Balance Sheet Valuation component [MM-IM-VP]), applying the respective legal rules.

Furthermore, both countries use SAP BW to calculate the management costs of acquisition used to determine their trade margins. In SAP BW, freight costs are handled in the following ways:

- **Freight costs in the United States**
  In the United States, freight costs are calculated by applying across-the-board percentage uplifts. Once a year, actual freight costs per article group are loaded into the data warehouse. These actual values are then used to produce an article group-specific percentage that is stored and later read and added via a formula in a transformation when loading data into the data marts. SYS's US subsidiary uses 50 article groups for this; their data marts contain data at the article-group level, not the article level.
- **Freight costs in Japan**
  Japan uses a much more exact approach when determining freight costs. In terms of SAP ERP, they have implemented the Material Ledger (CO-PC-ACT); data collected there are loaded into SAP BW via the standard DataSource `0CO_PC_ACT_1`. When filling the data marts, a routine within a transformation reads, corrects, cleanses, and aggregates these data per article group. At the moment, this routine comprises more than 27,000 lines of code. As in the United States, data marts only contain data at the article-group level, but Japan knows of 1,325 different article groups that cannot be assigned to the 50 used in the United States. Expressed in the language of data modeling, article groups in Japan and in the United States are not in an *n*:1 but in an *n*:*m* relationship.

6.3.2 Problem: No Single Point of Truth

Historically grown data flows

Both countries differ not only in terms of how and at what level of detail they calculate freight costs, but also in the architecture of their data warehouses. Ultimately, these differences are not only technical; they go back to hardware used, expectations in terms of processing speed, and different mentalities and structures that have evolved over time.

- **Data architecture in the United States: Processing data step-by-step**
  In the United States, the data warehouse more or less reflects the recommendations of SAP's LSA reference model. Data used to calculate the trade margin are processed step-by-step, passing through nine layers in total. Summarizing article data by article groups does not happen until the next-to-last layer; lump-sum freight costs are then applied

when passing on the data from this layer to the highest one. As a rule, the US system does not use routines.

The US data model has emerged from the fact that the data warehouse has been maintained for years by domestic and foreign partners. These partners are more competitive than SYS's own IT resources, although the level of experience with their people is a bit lower. Hence, SYS's US architecture team has attached great importance to creating simple and transparent structures.

- **Data architecture in Japan: complicated and complex routines**
  Japan also works with a layer-based model; the Japanese model, however, only has five layers. An even more potent difference from the US model is the fact that articles are grouped at a very early stage: this happens with data acquisition when extracting data from the source system. SYS Japan uses customer enhancements, such as customer exits or BAdIs, to make these extractions.

  There are two reasons that the Japanese have taken this route. On the one hand, they only use their own, highly qualified experts to maintain these systems. These experts sometimes tend to overstep the mark, developing almost artistic aspirations. On the other hand, downstream processing within the data warehouse is pretty complex. Implementing the follow-on routines on the article level instead of the article-group level led to major performance problems, so SYS Japan decided to classify articles in the course of the load process, from then onwards only working at the article-group level.

  The Japanese approach is further obfuscated by the fact that data from the material ledger (from DataSource `0CO_PC_ACT_1`) are still provided per article. Therefore, these data also need to be aggregated; this happens in a totally separate data flow that uses its own master data hierarchies to assign articles to article groups. The great danger with this is that the assignment of an article to an article group can be different in both data flows.

6.3.3 Numerical Example

Starting to analyze impact of inconsistencies

A new CFO joined SYS a few months ago. After a first analysis of the processes in controlling, he decided to dissolve the reconciliation team with immediate effect, instructing the head of group controlling to develop a business case for a centralized data architecture. In particular, the head of controlling is expected to estimate three numbers:

- Reconciliation-related costs in each country's IT department
- Damage caused by the failed stock market launch
- Costs of mistakes resulting from erroneous trade margins

When evaluating reconciliation-related efforts, the head of controlling decided to use Japan as an example. From his perspective, neither the stock market launch nor the costs of mistakes are country specific; instead, both aspects refer to decisions made in the corporate headquarters at Zug.

As an example for a typical mistake, he has picked the withdrawal of a complete article group from SYS's offering in the United States. All relevant numbers were converted into CHF at the same exchange rate; the effects of exchange rate fluctuations were therefore eliminated and ignored.

Reconciliation-Related Costs

Costs for reconciliation, consolidation, and repair work

SYSs IT department at Tokyo has 220 people in their own offices. Average annual per-capita costs for these employees are 100,000 CHF. Time studies in the course of last month's closing work have revealed that some 30 full-time equivalents (FTEs) are spending about five days per month clarifying differences. Another 55 are permanently assigned to projects that only serve the technical optimization of existing solutions. Although these projects are meant to address the issue of differences, they tend to make coding even more complex, and thus differing results become even more likely. In addition, five developers are allocated to the Japanese controller's office; their main job is to adjust the numbers delivered by Japan's own systems to match the headquarters' specifications.

A first estimate

A first rough estimate, based upon 200 net productive eight-hour working days per year and per person, leads to the conclusion that a better data architecture could save SYS Japan some 31% of its total IT costs, adding up to potential savings of seven million CHF per year:

- Total IT personnel costs: 220 * 100,000 CHF = 22,000,000 CHF
- Per-hour costs for people in IT: 100,000 CHF / (200 * 8) = 62.50 CHF

- (Annual) costs for 30 FTEs during closing: *30 * 5 * 8 * 62.50 CHF * 12 = 900,000 CHF*
- (Annual) costs for project team members: *55 * 100,000 CHF = 5.5 million CHF*
- (Annual) costs for IT staff at controller's office: *5 * 100,000 CHF = 0.5 million CHF*
- Total potential savings per year: 900,000 CHF + 5,500,000 CHF + 500,000 CHF = 6,900,000 CHF
- Total potential savings (percentage): *6,900,000 CHF / 22,000,000 CHF = 31 %*

Bearing in mind that Japan is just a medium-sized subsidiary, the potential savings in the group as a whole should amount to a triple-digit million CHF sum.

Damage Caused by Failed Stock Market Launch

Access to capital markets impeded

For a couple of years, SYS has suffered from a declining rate of return on capital employed (ROCE). Right now, this number is around 10%, relating to a total capital of 10 billion CHF. The stock market launch was meant to provide SYS with the capital it needed to improve online sales. SYS was planning to raise around five billion CHF and was planning to buy an e-tailer with a ROCE of (then) 25%. This would have enabled SYS to earn an additional 1.25 billion CHF per year.

Today, management is acting on the assumption that the stock market launch will have to be postponed by at least three years. Not discounting the respective amounts, this alone has therefore led to damages of about 3.75 billion CHF; in addition the opportunity to acquire the other company will most likely be gone in three years' time, and due to their poor presence in terms of online sales SYS's market position will have deteriorated further by then.

Costs of Mistakes

No sound basis for decisions

Due to the high pressure of competition, SYS has not been able to generate satisfactory trade margins with mobile phones in the United States for quite a few years. Mobile phones and tablets belonged to the same article group, mobile, in the United States; because mobile phones were far more influential than tablets, due to the sheer volume of their market, this has dragged down the trade margin for the whole article group.

Table 6.1 shows the trade margins and revenues for the mobile phones and tablets article subgroups and the corresponding figures for the mobile article group, which consists of these two subgroups five years ago.

Article Subgroup/ Group	Revenue	Trade Margin (%)	Trade Margin (Absolute)
Subgroup: Mobile Phones	103 M CHF	0.5%	0.515 M CHF
Subgroup: Tablets	70 M CHF	15.3%	10.710 M CHF
Group: Mobile	303 M CHF	3.7%	11.211 M CHF

Table 6.1 Trade Margins for the Mobile Article Group

Short-term margin losses worth the trouble

The table illustrates that tablets had a healthy margin of 15.3%, but the mobile article group as a whole was underperforming at 3.7%. However, because headquarters was focusing on the margins per article group it was decided to remove the whole lot from the mix in the United States a while ago. In Canada, where tablets belonged to the same article group as desktop computers, a similar mistake was made. Only SYS Germany saw tablets as a brand-new article group in its own right. At the moment, the Germans are achieving gross revenues of around one billion CHF and a margin of more than 100 million CHF per year, hawking about three million tablets every twelve months. Considering the much bigger US market, SYS is probably losing out big time there.

Young customers wandering off

Abandoning tablets also had an impact on SYS's image with younger customers in the United States. Because they were no longer selling tablets or oversized smartphones, the number of young customers in SYS's US markets has declined substantially. In the wake of this development, the numbers of expensive Internet-enabled TVs have declined as well, whereas sales of electric blankets and infrared lamps have increased a little bit. Unfortunately, these lamps and blankets are selling at a low price and are generating very little margin. In the end, SYS has also deprived themselves of the chance to bind customers to the brand in their younger years, harvesting from this once these young customers start to earn real money. From the perspective of the controlling department, quantifying the resulting losses in terms of revenues and results is impossible; on the other hand, all agree that these losses are going to be remarkable.

Decisions based upon data structures

The preceding example shows that SYS has made different decisions in different markets. These decisions were, however, not adequately in line with reality. They weren't based upon different environmental conditions but upon differences in SYS's data structures. Meanwhile, back at base, SYS is still struggling to define a common strategy for the tablet market. As well as missing opportunities on the sales side, this has also created cost disadvantages in procurement. Due to relatively low purchasing volumes with makers of tablets and a lot of variations in country-specific business strategies, SYS is seriously handicapped compared to more agile competitors.

6.3.4 Conclusion: Types of Data Model Inconsistencies

Makeshift design is driving outsourcing and offshoring

The CFO's intervention has clearly demonstrated that inconsistent structures and data models will cost you a lot of money and hurt your organization in a big way. Our experiences in the IT service business lead us to speculate that providers of onshore, nearshore, or offshore outsourcing services make most of their money from reconciliation and repair work that would never occur in a properly designed environment. Due to budgetary restrictions, affected systems are then often impaired, because the problem solvers are focusing on symptoms instead of actually thinking about the root cause of the problem.

Incorrect decisions are your key cost driver

However, no matter how high SYS's IT costs might be despite using offshore resources, mistakes are often more expensive. The lost margins with tablets we discussed here are only the tip of the iceberg. Erroneous decisions resulting from incorrect, inconsistent, or—as in our example—

inaptly aggregated numbers are devilishly expensive. Maybe the United States is wasting 500 million CHF of margin per year by not selling tablets, but these costs are dwarfed by almost four billion CHF missed due to the lack of capital. Our examples in terms of the failed stock market launch or the article groups are taken from real events with real customers. We have, however, changed the industry, the figures, and the scenario a bit to protect the (not so) innocent.

Types of inconsistencies

Based upon our thoughts in this section so far, you can distinguish between two different types of data model inconsistencies:

- Data model inconsistencies can result from the fact that identical functional requirements are not only modeled redundantly but also differently. The more implementations of the same thing there are, the more likely it is that inconsistencies will arise.
- As with all models (see the definition of models in Chapter 4, Section 4.1), there can be inconsistencies between data models and the reality they are meant to describe. As mentioned when defining deduction and induction (see Chapter 5, Section 5.4), your data models should not be driven by prejudice or limited to only what the designers can imagine—for example, in terms of which articles should belong to the same article group—but instead should be set up without any fixed expectations of any kind of result. Otherwise, it is highly likely that your decisions will not be based upon the reality out there; they would only be adequate if that reality happens to match your expectations. Think about your personal life, and figure out your hit rate in terms of properly modeling reality.

[+]

Metadata Repository Defining Scope

The more your rely on procedural, complex, and nontransparent code, the more often you will encounter unexplainable differences between data flows that should theoretically provide identical results. Explaining these differences—let alone identifying and fixing their causes—will become harder and harder; one reason for this is the lack of up-to-date and detailed metadata. Instead of trotting down the road to hell, spend a bit of time meditating about data flows that use declarative tools to process your data in a lot of easily understandable small steps.

In this scenario, we are assuming that you have suitable metadata at your command. We will therefore focus on how big data can help you avoid inconsistencies between a data model and reality on the basis of these metadata. (It should be mentioned in passing, however, that big data can also help you detect and eliminate inconsistencies between data flows, regardless of whether we are talking about data flows in the same or in different data models.)

[«]

Big data can compare data models

Taxonomy

A *taxonomy* is a (often hierarchical) schema used to classify beings, objects, or terms. For example, a trade margin is calculated using net sales price and cost of acquisition. To determine the cost of acquisition, you'll need (among other things) transport costs, customs, and taxes; transport costs in turn may comprise freight costs and postal charges.

Sometimes, the word *taxonomy* not only refers to the outcome of classification efforts but also to the science of classifying things—that is, to building taxonomies in the narrower sense.

6.4 Solution: Generating Layers and Domains Automatically

Basic structure of a data model

A simple, two-dimensional data model is nothing but a table or a matrix. In the context of data models, the rows of such a matrix are called *layers*, and its columns are called *domains*. Moving from the data model's bottom to its top, data are split, merged, and processed in different ways by (vertical) data flows (see Figure 6.2):

- D may contain all the data within A and B or within a part of A plus a part of B.
- E may contain all the data within B and C or within a part of B plus a part of C.
- F and G may contain all the data within C or a part of the data within C.
- The data within D, E, and F may just be a selection of the data in A, B, and C. They may, however, also have been processed on the way via any formulas or algorithms of your choice.

We are going to use the basic structure shown in Figure 6.2 in all following case study chapters to illustrate our ideas about the data architecture for each scenario.

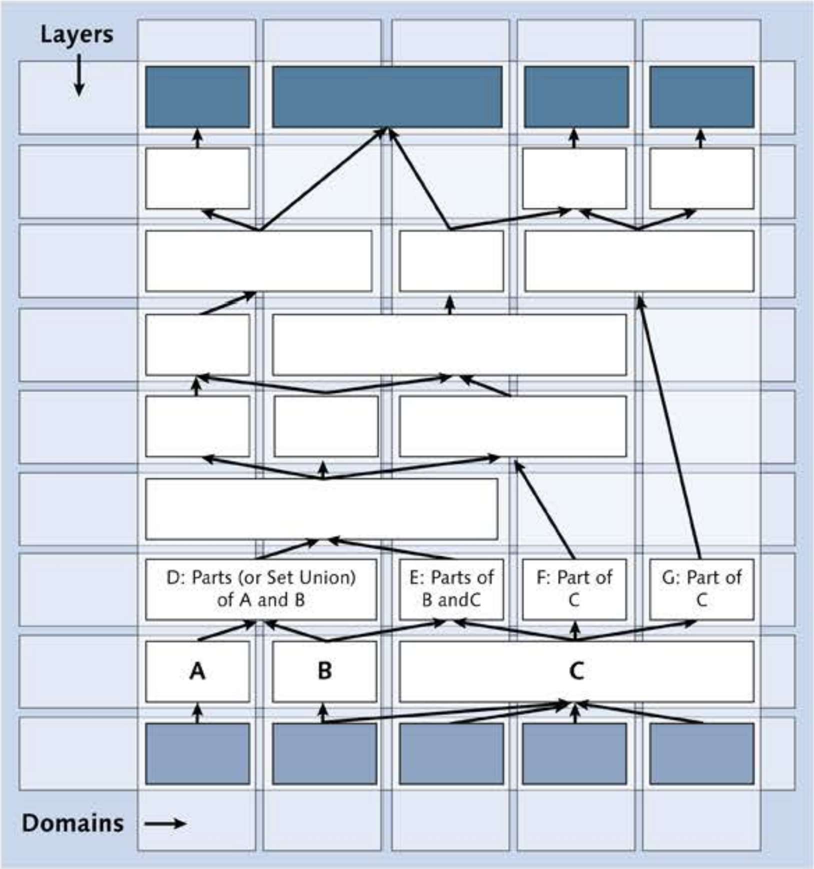


Figure 6.2 Data Model (Basic Structure)

Data source layer

The layer at the very bottom (red, if you are reading the electronic version of this book) represents data held in the source systems, whatever they may be. They could be SAP or non-SAP solutions; in the former case, the data could already be within SAP HANA—as with SAP Business Suite on SAP HANA—or outside of SAP HANA.

Data acquisition and replication will look different in each solution. For data that are already held within the SAP HANA database we are going to work with, we only need to implement views. For data outside that database, we may have to create virtual tables or set up data-acquisition processes in data logistics solutions, such as SAP Data Services.

Data within SAP HANA

All the boxes on other layers (the green ones in the electronic version of the book) represent data within SAP HANA. The arrows connecting these boxes stand for processing steps within vertical data flows that transform the data in the ways described under "Analyzing Root Causes within Your Models" in Chapter 4, Section 4.4.3. Frequently, such an arrow would correspond to a procedure in SQLScript.

Client layer

The top layer (blue in the electronic version of the book) in Figure 6.2 represents clients. By *clients*, we generally mean two different kinds of solutions:

- Clients can be reports implemented, for example, using products within the SAP BusinessObjects BI portfolio and displayed on whatever kind of device you choose.
- Clients can also be applications that send messages to people or other applications. We will further review this scenario in Chapter 9.

Layers and domains

Our data models are not only split vertically (into layers) but also horizontally into domains; in Figure 6.2, the second layer counting from the bottom is split into A, B, and C. The number of layers and domains within our data models is only an example; in your specific projects, layers and domains may be broken down further or may be combined instead.

Different domains on different layers

Whereas layers stand for processing steps, the structure of domains is determined by the nature of your data and the way in which you would like to process them; remember our thoughts regarding levels of measurement under "Monitoring Forecasts and Models Separately" in Chapter 4, Section 4.4.3. The number of domains as well as the grouping principles according to which they are formed can be different on each layer. We have tried to imply this in Figure 6.2. There are three domains (A, B, and C) on the lowest layer within SAP HANA (the second layer from the bottom), then there are four (D, E, F, and G) on the layer above, and there is only one (nameless) domain on the next layer. The arrows (our processing steps) sometimes connect domains within consecutive layers but sometimes also stretch across a number of layers.

Domains in lower and upper layers

For obvious reasons, the structure of the domains within lower layers—those that are closer to the source systems—depends on the structure of the data to be acquired. In SAP BW, the structure of the DataSources (i.e., which data are delivered by which source and in what sequence) determines the structure of domains; therefore a domain often reflects a combination of a DataSource and a source system. The layers at the top that are closer to clients are there to satisfy either business or—if your solution is feeding data to another application—target system requirements.

Data flows within our model

A data flow moves data within a domain on a lower level into a domain on an upper level, passing various other domains and layers; a data flow can therefore also be seen as a series of processing steps. Imagine that your data model was a map of a North American metropolis: in this analogy, a data flow would be a series of turns telling you how to get from Wall Street (the data source) to East Village (a data mart). Unlike when strolling through the Big Apple, with data modeling you can move, remove, or merge buildings, divert rivers, and redesign streets. Furthermore, the data you are moving around will change during transport.

Layers between data source and data mart

We have no objections to aligning the data-acquisition layer(s) with your source systems or to aligning the data marts or reporting layers with more or less static, inflexible business requirements; however, you damage the relationship between your data model and reality in two situations:

- **Functional requirements/data sources worming through**
  Business or functional requirements don't only have an impact on the top layers of your data model; they also snap through to the bottom, determining the domains on *most* layers. Conversely, data sources can have an unexpectedly long-range effect all the way up your data model. Whichever way it happens, the structure of the layers in the middle is no longer semantically neutral but determined by either the senders or the receivers of data.
- **Functional requirements limiting your insights**
  Functional requirements tend to express more or less arbitrary views on the world. They are often driven by some kind of preconception about what the world is like, in which case they also limit your epistemological horizon, working like a filter that only lets insights pass that had crossed your mind from the very start.

Why should SYS group articles according to their functionality and not by price? Why are their sales regions based upon districts or states and not by the nearest sales office? It's because we are dealing with structures somebody came up with ages ago that are now trying to structure a very complex reality using straightforward categories.

Decisions are key

In the end, both situations mentioned previously address the same issue. In the first situation, the central question of concern is how the layers in the middle of your data model should be constructed to be as flexible as possible; the second situation extends that question to the upper layers, asking not only for flexibility but also for more closeness to reality to generate decisions that produce the desired effect.

Old structures might not breed new ideas

Reports are fed from data marts, and data marts are the final destination of data flows. If we limit ourselves to concepts we already had in mind when designing these data flows, there can't be much room for new ideas. Sounds logical, right?

In our example, there are two options for each article group. We could keep selling the articles within that group or drop them altogether, removing the article group from our assortment. But why should you use the trade margin of the last calendar month or even the last year as a basis for that decision? Why the trade margin and not the sales revenue?

Expected sustainable trade margin

SYS could build an alternative approach around something like an *expected sustainable trade margin* (ESTM)—that is, some kind of long-term-oriented measure. ESTM could be something like the average trade margin to be expected—in the sense of an expected value—for a certain article group throughout the coming three years. In this case, the article group's trade margin during the last month might not be irrelevant, but it would become one input parameter among many others determining ESTM. Conceivably, the average age of customers buying articles from that article group or their average annual purchasing volume might be equally important.

Limiting assumptions in the data model

SYS's existing data model is not only built around a key figure that is backward looking but also has a couple of other weaknesses. It is also founded on a lot of unproven assumptions—for example:

- That the current article groups used by each country embody the optimum structure for that subsidiary (that is, the one leading to the best decisions in terms of shareholder value)
- That the taxonomy used for categorizing articles should be based upon functional criteria (such as "tablet" versus "laptop") and not, for example, on their respective attractiveness to younger customers

- That the grouping of articles (and based upon that, the data flows used to calculate trade margins per article group) can be treated as stable over a considerable timespan and that the underlying parameters determining customer behavior and the interdependencies among these parameters remain unchanged over months and possibly even years
- That it makes sense to make all decisions at article-group level instead of looking at each article individually

Data models in tune with today's reality

SAP HANA gives you the chance to question assumptions like these, developing dynamic data models that do not reflect the biased views of past generations of managers but that simply adapt to whatever is happening right now. Adjusting data models to an ever more quickly changing reality will then no longer take decades but just days, hours, or even minutes.

Running through alternatives

One of the key advantages of high-performance data crunchers like SAP HANA is that you can run through hundreds, thousands, or millions of alternatives. Instead of just calculating one key figure at an article-group level, you can calculate dozens of them, at the same time going through alternative groupings. Or opt for maximum granularity instead, performing your computations at article level or even at document level if necessary.

Then, once you have the results of these calculations, you can also use SAP HANA to automatically generate decision trees and therefore data flows. If these data flows are then implemented using virtual instead of persistent objects, as suggested by SAP's LSA++ reference model, changes resulting from new insights can be implemented easily and, to a certain extent, automatically.

[»]

Decision Trees

Decision trees are directed graphs that resemble trees or flowcharts; they are used for decision making. A decision tree contains a series of questions that can be answered on the basis of existing data. Answering these questions in a certain, predefined sequence will take you from the tree's root to one of its leaves, which in the end provides you with a statement about the decision you want to make.

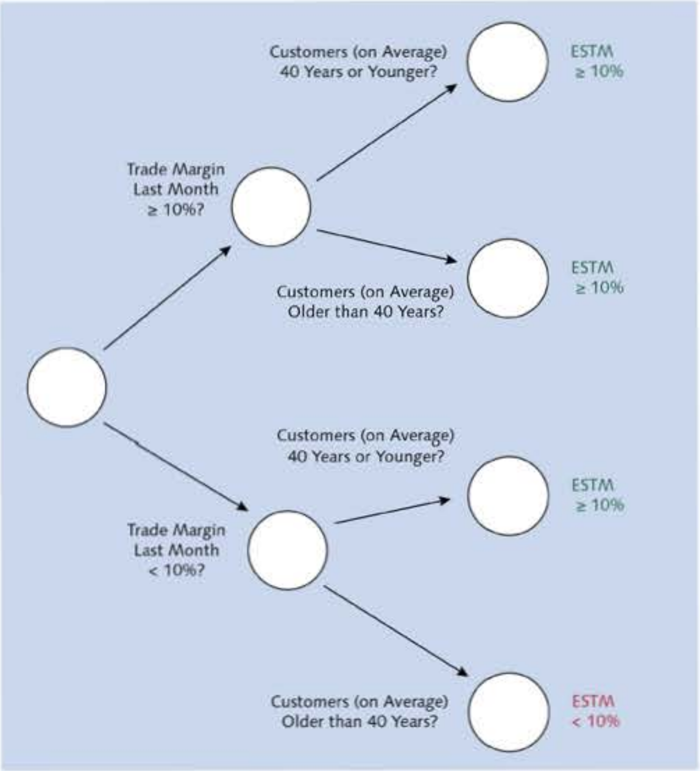


Figure 6.3 (Binary) Decision Tree (Sample)

Decision tree for SYS

A possible decision tree for SYS could look like the one shown in Figure 6.3. Let us take you through the tree from left to right:

- First, SYS asks whether the trade margin of the article group under review was higher or lower than 10% last month.
- Next, SYS asks whether the average buyer of articles within that article group was younger or older than 40.
- After both questions are answered, the decision tree delivers a forecast for the article's ESTM—that is, its long-term trade margin—that indicates whether it is expected to be below or above 10%. As 10% is SYS's threshold for kicking article groups out of their markets, this statement is in the end identical to a recommended decision.

The decision tree shown in Figure 6.3 is called a *binary* decision tree because each question has only two possible answers, each of which then takes you to another, alternative node (the next question or leaf).

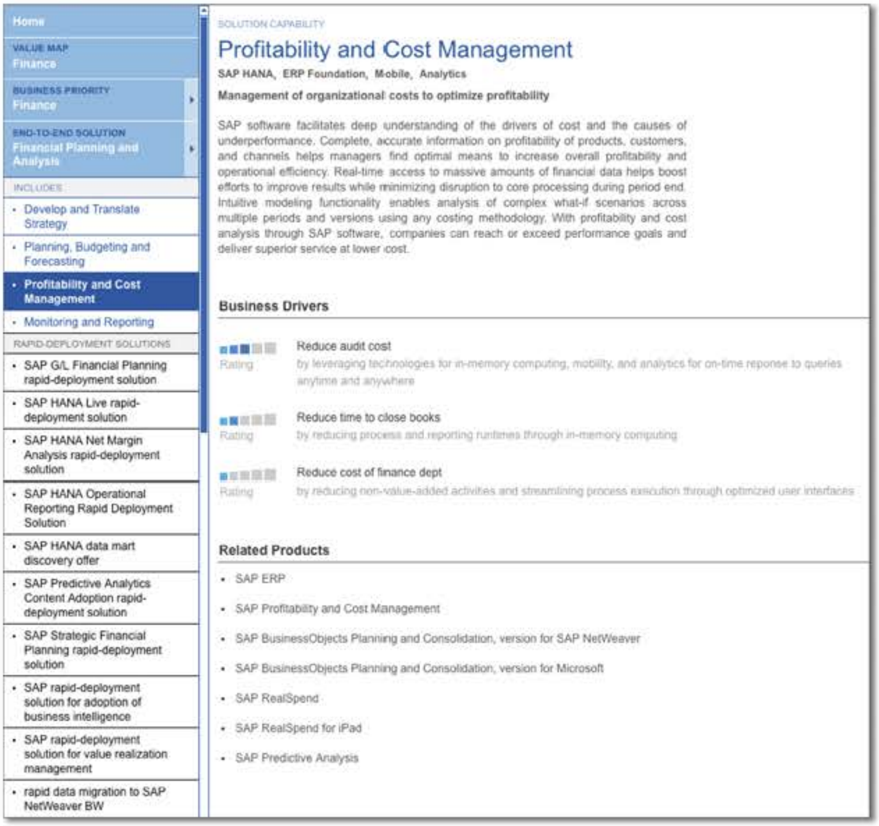
Decision trees don't have to be binary

Decision trees don't necessarily have to consist only of binary junctions and are normally a lot bigger and a lot more complex than the one shown in Figure 6.3. Their validity can be evaluated using test data. For obvious reasons, such test data should not be the same as those have been used to construct or train the decision tree.

6.4.1 Related Value Maps in SAP Solution Explorer

Value maps related to trade margin

When looking at the business processes within our example, you might be tempted to turn to the value matrix for retail mentioned in Chapter 7. The trade margin is, however, a brainchild of the SAP Profitability and Cost Management solution; details about this solution can be found via Finance • Financial Planning and Analysis (see Figure 6.4).



Home

VALUE MAP
Finance

BUSINESS PRIORITY
Finance

END-TO-END SOLUTION
Financial Planning and Analysis

INCLUDES

- Develop and Translate Strategy
- Planning, Budgeting and Forecasting
- Profitability and Cost Management
- Monitoring and Reporting

RAPID-DEPLOYMENT SOLUTIONS

- SAP G/L Financial Planning rapid-deployment solution
- SAP HANA Live rapid-deployment solution
- SAP HANA Net Margin Analysis rapid-deployment solution
- SAP HANA Operational Reporting Rapid Deployment Solution
- SAP HANA data mart discovery offer
- SAP Predictive Analytics Content Adoption rapid-deployment solution
- SAP Strategic Financial Planning rapid-deployment solution
- SAP rapid-deployment solution for adoption of business intelligence
- SAP rapid-deployment solution for value realization management
- rapid data migration to SAP NetWeaver BW

SOLUTION CAPABILITY

Profitability and Cost Management

SAP HANA, ERP Foundation, Mobile, Analytics

Management of organizational costs to optimize profitability

SAP software facilitates deep understanding of the drivers of cost and the causes of underperformance. Complete, accurate information on profitability of products, customers, and channels helps managers find optimal means to increase overall profitability and operational efficiency. Real-time access to massive amounts of financial data helps boost efforts to improve results while minimizing disruption to core processing during period end. Intuitive modeling functionality enables analysis of complex what-if scenarios across multiple periods and versions using any costing methodology. With profitability and cost analysis through SAP software, companies can reach or exceed performance goals and deliver superior service at lower cost.

Business Drivers

Rating
Reduce audit cost
by leveraging technologies for in-memory computing, mobility, and analytics for on-time reponse to queries anytime and anywhere

Rating
Reduce time to close books
by reducing process and reporting runtimes through in-memory computing

Rating
Reduce cost of finance dept
by reducing non-value-added activites and streamlining process execution through optimized user interfaces

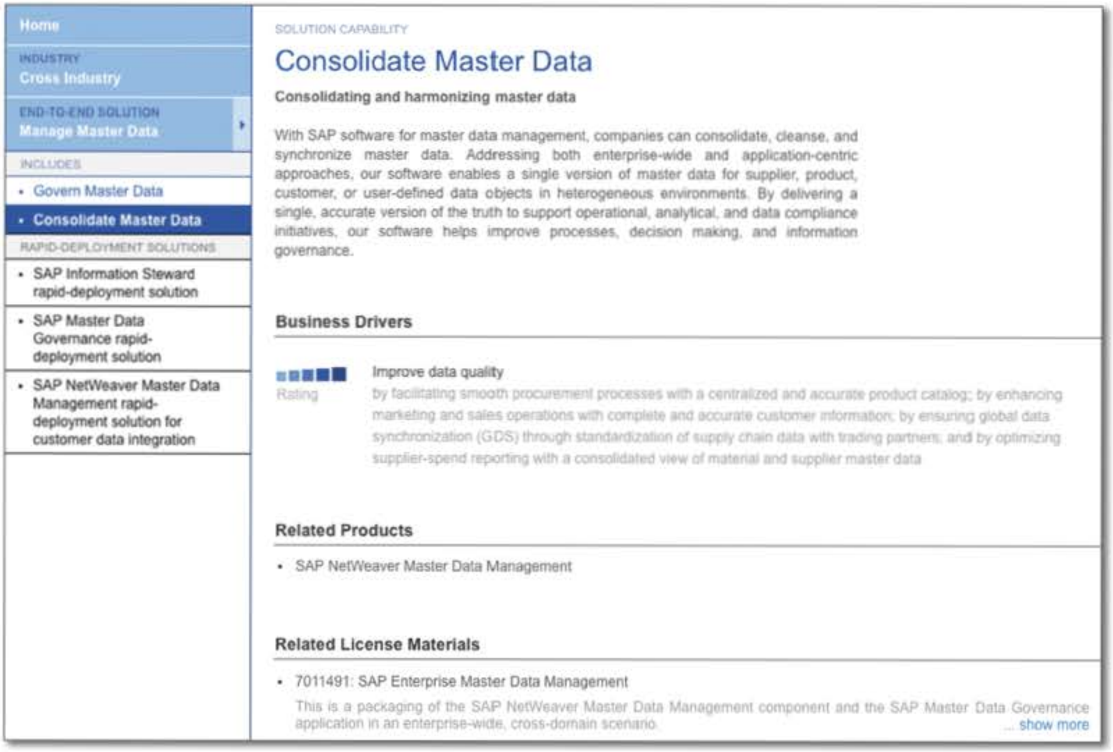
Related Products

- SAP ERP
- SAP Profitability and Cost Management
- SAP BusinessObjects Planning and Consolidation, version for SAP NetWeaver
- SAP BusinessObjects Planning and Consolidation, version for Microsoft
- SAP RealSpend
- SAP RealSpend for iPad
- SAP Predictive Analysis

Figure 6.4 Profitability and Cost Management Solution

Value maps for metadata

Potentially, you will also need data stored in other, article-related attributes; one example of this are characteristic values that could be used for grouping to enrich your data. Such material master data are managed within SAP Master Data Governance, which can, for example, be found in SAP Solution Explorer under ALL SOLUTIONS (set at the top right using SELECT VIEW) and then CROSS INDUSTRY • ENTERPRISE INFORMATION MANAGEMENT • END-TO-END SOLUTIONS • ENTERPRISE INFORMATION MANAGEMENT • MANAGE MASTER DATA • CONSOLIDATE MASTER DATA (see Figure 6.5).


Home
INDUSTRY
Cross Industry
END-TO-END SOLUTION
Manage Master Data
INCLUDES
• Govern Master Data
• Consolidate Master Data
RAPID-DEPLOYMENT SOLUTIONS
• SAP Information Steward rapid-deployment solution
• SAP Master Data Governance rapid-deployment solution
• SAP NetWeaver Master Data Management rapid-deployment solution for customer data integration
SOLUTION CAPABILITY
Rating
... show more

Consolidate Master Data

Consolidating and harmonizing master data

With SAP software for master data management, companies can consolidate, cleanse, and synchronize master data. Addressing both enterprise-wide and application-centric approaches, our software enables a single version of master data for supplier, product, customer, or user-defined data objects in heterogeneous environments. By delivering a single, accurate version of the truth to support operational, analytical, and data compliance initiatives, our software helps improve processes, decision making, and information governance.

Business Drivers

Improve data quality
by facilitating smooth procurement processes with a centralized and accurate product catalog; by enhancing marketing and sales operations with complete and accurate customer information; by ensuring global data synchronization (GDS) through standardization of supply chain data with trading partners; and by optimizing supplier-spend reporting with a consolidated view of material and supplier master data

Related Products

- SAP NetWeaver Master Data Management

Related License Materials

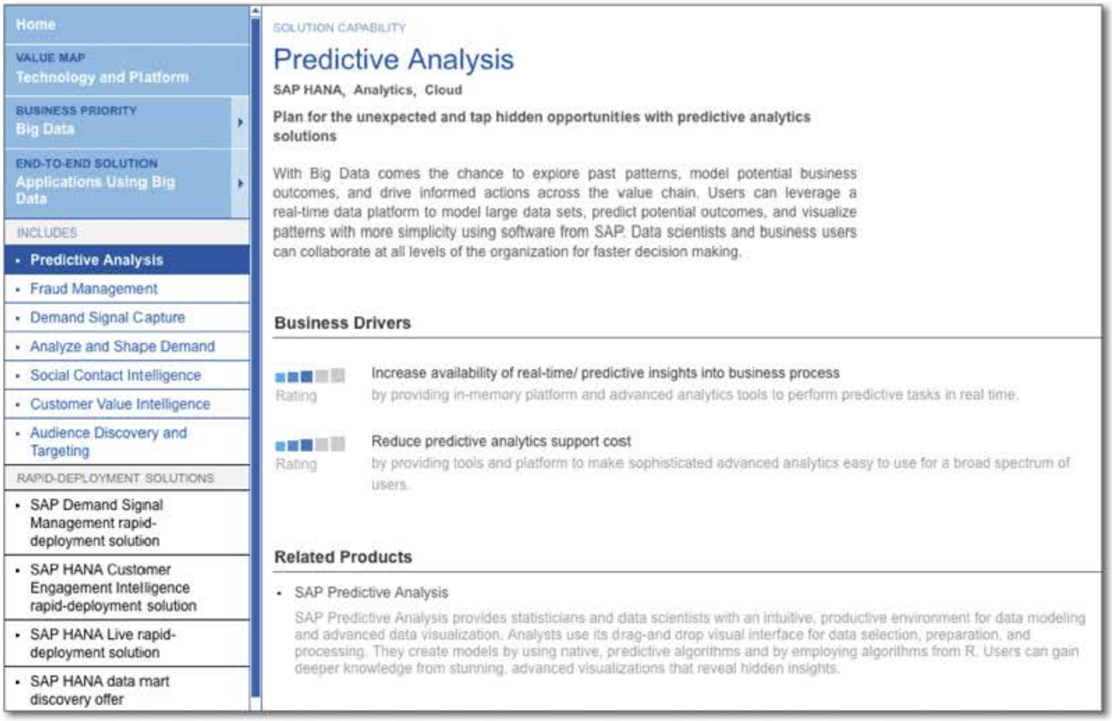
- 7011491: SAP Enterprise Master Data Management
  This is a packaging of the SAP NetWeaver Master Data Management component and the SAP Master Data Governance application in an enterprise-wide, cross-domain scenario.

Figure 6.5 Consolidate Master Data Solution

Another set of SAP tools in enterprise information management that could be relevant here are SAP's products for data management, such as their SAP Information Steward rapid-deployment solution: CROSS INDUSTRY • ENTERPRISE INFORMATION MANAGEMENT • END-TO-END-SOLUTIONS • ENTERPRISE INFORMATION MANAGEMENT • INFORMATION GOVERNANCE • SAP INFORMATION STEWARD RAPID-DEPLOYMENT SOLUTION.

Value map for our solution

The majority of the algorithms we will address in Section 6.4.3 are implemented, for example, in SAP PAL; in the SAP Solution Explorer, SAP PAL can be found within the TECHNOLOGY AND PLATFORM value matrix under BIG DATA • APPLICATIONS USING BIG DATA • PREDICTIVE ANALYSIS (see Figure 6.6).


Home
VALUE MAP
Technology and Platform
BUSINESS PRIORITY
Big Data
END-TO-END SOLUTION
Applications Using Big Data
INCLUDES
• Predictive Analysis
• Fraud Management
• Demand Signal Capture
• Analyze and Shape Demand
• Social Contact Intelligence
• Customer Value Intelligence
• Audience Discovery and Targeting
RAPID-DEPLOYMENT SOLUTIONS
• SAP Demand Signal Management rapid-deployment solution
• SAP HANA Customer Engagement Intelligence rapid-deployment solution
• SAP HANA Live rapid-deployment solution
• SAP HANA data mart discovery offer

SOLUTION CAPABILITY

Predictive Analysis

SAP HANA, Analytics, Cloud

Plan for the unexpected and tap hidden opportunities with predictive analytics solutions

With Big Data comes the chance to explore past patterns, model potential business outcomes, and drive informed actions across the value chain. Users can leverage a real-time data platform to model large data sets, predict potential outcomes, and visualize patterns with more simplicity using software from SAP. Data scientists and business users can collaborate at all levels of the organization for faster decision making.

Business Drivers

Rating — Increase availability of real-time/ predictive insights into business process
by providing in-memory platform and advanced analytics tools to perform predictive tasks in real time.

Rating — Reduce predictive analytics support cost
by providing tools and platform to make sophisticated advanced analytics easy to use for a broad spectrum of users.

Related Products

- SAP Predictive Analysis

  SAP Predictive Analysis provides statisticians and data scientists with an intuitive, productive environment for data modeling and advanced data visualization. Analysts use its drag-and drop visual interface for data selection, preparation, and processing. They create models by using native, predictive algorithms and by employing algorithms from R. Users can gain deeper knowledge from stunning, advanced visualizations that reveal hidden insights.

Figure 6.6 Predictive Analysis Solution

6.4.2 Functional Requirements

Data flows designed inductively with SYS

In the future, SYS would like to design data flows for decision support without prejudice. This means that data flows should no longer be built deductively, founded on unproven assumptions, but instead should be generated inductively from data (see also Chapter 5, Section 5.4). To put the icing on the cake, these newly generated data flows should also be standardized and consistent, unambiguous, transparent, and easily changeable across the whole group of companies within SYS.

Seven-Point Plan for Metadata

To get there, SYS has set up a seven-point plan:

1. **Taxonomy of key figures**
   As a starting point, SYS plans to establish a taxonomy of key figures. To get an idea of the result, look again at our trade margin-related example in Section 6.3.4.

2. **Taxonomy of characteristics**
   In parallel, SYS would like to build a taxonomy of characteristics. This taxonomy is intended to tell people which characteristics are in fact subcategories of others. In the case of SYS, a hierarchy of characteristics could look like the following:

   - Articles are grouped into article subgroups
   - Article subgroups are grouped into article groups
   - Article groups are grouped into (business) segments

   There can be an *n*:1 relationship between characteristics on two neighboring levels of the taxonomy. There can, however, also be more complex relationships:

   - Segments may not only be derived from article groups (*n*:1) but also from a combination of article group and country. If there are *x* article groups and *y* countries, we could be dealing with up to $n = x * y$ combinations—that is, an $x * y$:1 relationship. The technical term for this kind of setting in SAP BW is *compounding*.
   - As well as its use with the segment, the country could also be relevant for determining a region, in which case country is used together with the article group to determine a segment and on its own to define the region.

   There are also characteristics for which there is neither an *n*:1 nor an $x * y$:1 but instead an *n:m* relationship. One example of this is the model and the color of a car. Each model can come in a number of colors, and each color is used with various models. Such relationships cannot be put down in a simple hierarchy but instead may need a matrix or even a multidimensional equivalent of a matrix for filing.

3. **Allowable key figures**

   A key figure can only be calculated if a data record contains the data required for it. This may mean that certain characteristics or key figures will have to be delivered but could also mean that certain characteristics have to take specific values—such as *(country = Japan)* $\vee$ *(country = China)*—for the key figure to make sense:

   - A trade margin can only be determined for a data record if both net sales price and cost of acquisition are contained in it or can be determined from it.
   - A trade margin could be calculated at various levels; it may be meaningful for sales documents, article groups, or articles but not for legal entities (called *company codes* in SAP terminology), or calculating it may only make sense for *some* legal entities but not for others (such as for IT departments set up within their own company codes).

   The field delivered by a data record and the contents of these fields therefore have an impact on which key figures can be (theoretically) calculated. Thus, for each combination of key figures, characteristics, and characteristic values, there is a set of key figures that are theoretically determinable. When thinking about this set, one faces both mathematical and business restrictions. The former are semantically neutral, and the latter refer to the meaning or contents of data.

4. **Internal consistency**

   SYS would like to continuously check taxonomies of key figures and characteristics for their internal consistency, making sure they are unambiguous and free from contradictions. In the case of discrepancies, a newly formed data-governance team is informed immediately.

5. **Replicating data**

   SYS plans to implement a new big data solution that produces analyses for decision support. In SYS's new metadata repository, every field mapped within that new solution will be assigned to one of three categories: dimension, attribute, or fact. In addition, other metadata will be collected for fields; with key figures that are mostly facts, this would, for example, include their levels of measurement. All metadata are intended to be collected and updated in real time, and the metadata repository will perform the respective checks for internal consistency on them. Inconsistencies will trigger alerts with data governance or with the person creating the object, the respective data owner, the responsible data steward, or all of them.

6. **Generating data flows**
   The taxonomies defined in steps 1 to 3 will change all the time. Based upon the assumption that all changes are updated within the solution and within the metadata repository without delay, the new big data solution is intended to find out which of the theoretically determinable key figures should be calculated for which combinations of characteristics and characteristic values and what the data flows doing that should look like.

7. **Eliminating redundancy**
   Data flows that were generated (semi) automatically also need to be checked for redundancy. Some of them might serve very similar purposes or might want to do things that are already catered for by other existing data flows.

[«]

Fact

In a data warehousing context, a *fact* is a value or a measurement referring to an entity. In return, entities are defined by a number of characteristics values. In OLAP's classic *star schema*, a number of dimensions are arranged around one or more fact tables in a star-like manner.

Each dimension uniquely identifies specific objects or entities via a specific ID or key. Entities in data modeling are concrete (tangible) or abstract (intangible) objects that can be positively identified. One example of a concrete object is a customer (John Doe); an abstract object could be a customer case (identified by its case ID) in an IT support system. Data warehousing facilitates collecting, storing, processing, and evaluating facts about such objects.

Together, the IDs of all dimension tables comprise the *concatenated key* of the fact table, which in turn contains the data you are really interested in. Data within the fact table often refer to transactions in which all entities have been involved and which can therefore be identified using the combination of all dimension IDs. Such transactions (that is, a combination of dimension IDs) can be seen as entities in their own right (e.g., a sales transaction); this is why fact tables are also called *fact entities* in IBM systems and *central entities* in SAP HANA.

Dimensions and facts within the star schema

Figure 6.7 shows a sample star schema for our scenario. The first characteristic within each dimension table (e.g., credit card number) serves as a unique key to identify a specific entity (e.g., customer). That said, we are obviously assuming that people only use their own credit cards and don't "borrow" them from Mom or Dad. The keys of some dimensions (e.g., article) consist of more than one ID (e.g., article code or batch). Dimension tables may also contain supplementary attributes (e.g., age of customer); attributes are properties that describe the objects/entities within a dimension table.

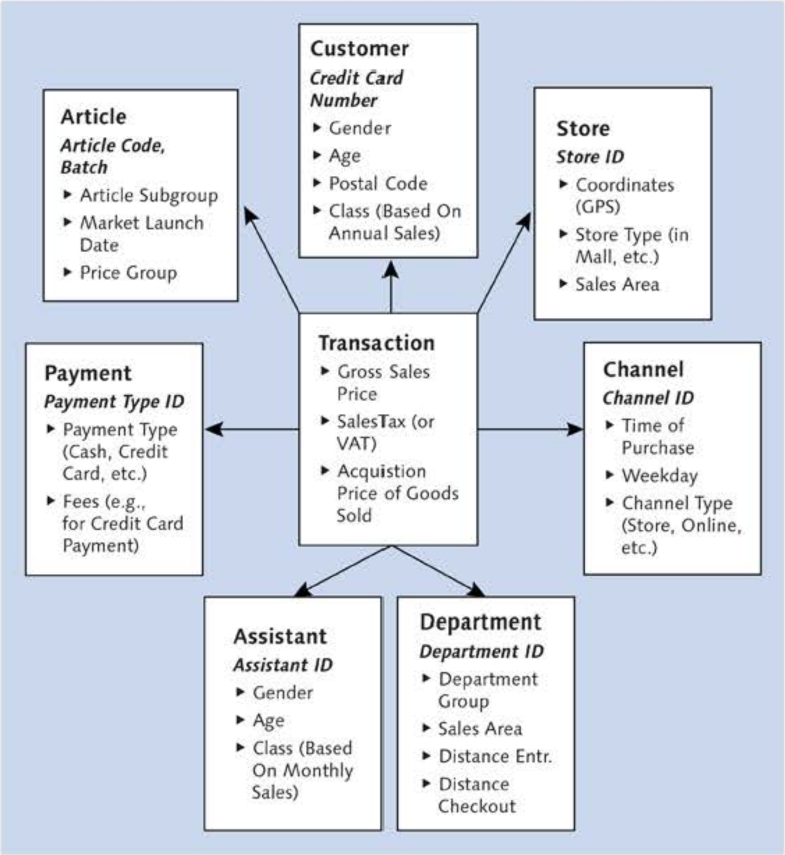


Figure 6.7 Star Schema with Dimensions and Facts

In our example, all dimensions together are used to define a sales transaction as a fact entity. The fact table then contains facts or values about this sales transaction, one of which could be the gross sales price. We sometimes use the terms *fact* and *key figure* synonymously; in reality, there may be facts that are not key figures at all (e.g., payment default, yes or no). In SAP BW, the object used to implement a star schema would be an InfoCube and/or a MultiProvider; in SAP HANA, you would use an analytic view.

Sample star schema: notes

Some annotations for Figure 6.7 are as follows:

- **Article**
  - The market launch data might tell you how novel or fashionable a product is. In this case, the market launch data would be an attribute of the article.
  - The article group, on the other hand, can be treated as an attribute of the article that has to be stored in the dimension table but can also be derived during processing from a taxonomy that arranges articles by article groups.
- **Customer**
  - SYS is segmenting its customers into classes by annual sales.
  - The customer class is updated after each sales transaction.
- **Store**
  - The store type tells you whether a store is located within a shopping mall or in a greenfield commercial area.
  - The sales area defines the size of the floor space used to sell goods.
- **Channel**
  - The time of purchase comes into the equation because SYS is treating different timeslices as different channels.
  - A day is split into 96 intervals of 15 minutes each.
  - Also, two purchases taking place at the same time but on different weekdays are treated as transactions via different channels from SYS's perspective.
- **Department**
  - SYS also knows how much space has been assigned to each department.

  - Two other attributes of a department are its distance from the store entrance and the distance from its center to a checkout.
- **Assistant**
  - Shop assistants—like customers—are described by gender and age.
  - Unlike for customers, their classification is not driven by annual but by monthly sales.
- **Payment**
  - The payment type defines how customers have paid for their purchases.
  - Payment transaction fees can be stored here (a percentage as an attribute) or can be determined while storing a transaction.

Allowable key figures

Let's return to our idea of allowable or determinable key figures, introduced under "Seven-Point Plan for Metadata" in Section 6.4.2. The metadata collected for items 1 through 5 of SYS's seven-point plan define something that looks like an *n*-dimensional *space of options*, that is, all allowable or meaningful combinations of dimension IDs, attributes, and facts.

You can imagine that this *n*-dimensional space can become quite large, a lot larger than anything even SAP HANA could handle, which means that when modeling you'll have to focus on certain extracts from it, extracts that are, for example, defined by only looking at one or just a few value drivers. Also, an organization's data doesn't contain all theoretically possible combinations of dimension IDs, characteristic values, and key figures. Therefore, the data to be processed themselves help you reduce the size of that *n*-dimensional space; you'll only have to look at combinations that can be found in your data instead of pondering all combinations that could theoretically occur. Furthermore, the process described in this chapter can also be performed by using samples of your data instead of chewing through all the data that you have collected.

Determining Attributes and Facts

Aggregating or calculating facts

The values of attributes and facts, such as a customer's age or the gross sales value, are either contained in the data delivered by your source systems or have to be calculated from these data and all other information you may have (such as taxonomies). If your source systems deliver data, the key question you'll have to answer is whether they can be aggregated and how; a percentage, like the relative trade margin, cannot simply be added across article groups. If the data you need are neither delivered by your source systems nor computable via a simple aggregation, you will need formulas or algorithms within a routine, a procedure, or a function, which will—hopefully—be reusable.

Re-usability determines approach

Sometimes, both paths can be taken; the trade margin across a number of article groups can either be recalculated from their total net sales and costs of acquisition or obtained via an aggregation by calculating the weighted average of all individual trade margins. Or, if it's more convenient, you could use the difference between net sales price and costs of acquisition per article group, add these differences up, and then divide them by the total costs of acquisition for all the article groups you are looking at. When it comes to building very flexible big data solutions, the best option is not the one with minimum performance requirements but the one with maximum reusability.

[Ex]

Reusability and Abstraction

Reusability is not new in IT at all. Even in the good old mainframe days, programmers used subroutines that are by definition reusable. Reusability and flexibility also play an important role in object-oriented programming, in which two key concepts related to reusability are polymorphism and late binding. In this book, however, we want to push the idea of reusability a bit further. As this concept is key to a couple of chapters, we want to spell it out a bit via an example.

An insurance company like the one mentioned in Chapter 8 uses sensor data from vehicles to estimate the risk appetite of individual drivers and to calculate their insurance premiums on that basis. Let's take a driver's cruising radius (how far they travel away from home as the crow flies) during the last billing period and assume that it can be used as an indication of the probability that this driver will have an accident in a certain cost category throughout the coming year. Maybe drivers with a wider radius are more experienced and primarily use highways, whereas those with a smaller radius are the typical Sunday drivers roaming around inner-city streets, and therefore the latter group is more prone to accidents. Or maybe it's the other way around; perhaps Sunday drivers are usually elderly but also extremely cautious drivers. Regardless, such data can easily be collected via GPS devices.

The same insurance company is also gathering the movement data of its field staff (claims adjusters, salespeople, etc.)—at least in countries where this is legally possible. These data are used for scheduling the appointments of claims adjusters and for route optimization with sales. Analyzing these data in different ways with an agile BI approach has shown that the cruise radius is pretty useful here as well. Employees with a narrower cruise radius are traveling less and are able to squeeze more appointments into a working day than those working in rural areas and driving longer distances. This distance is determined monthly for employees and annually for customers.

Most customers we have met so far would have satisfied both requirements via separate routines residing in separate data flows. However, although we may be looking at a different data source and different periods with customers and employees, both routines do roughly the same thing.

The same formulas and algorithms might be of interest in other, yet unexplored, applications. Maybe the cruise radius, collected and uploaded via a free emergency call app that has been recently made available by the insurance company and that is also used by prospects who are not yet customers, can help predict which kinds of insurance policies users will be interested in.

Extended reusability

When talking about reusability, we mean that exactly the same actual routine (and not a copy of it) is used in all cases; this means that the code in question only exists once within the whole organization. To make this possible in a maximum number of cases, three conditions have to be fulfilled: routines need to be abstract, free of semantic aspects, and purely processing oriented. In our example, we would need code that can determine the geographical center using *n* sets of coordinates and then calculate something like a cruise radius on that basis. Furthermore, this routine should be decoupled from data sources and business requirements, meaning there should at least be one layer separating it from both data sources and data marts.

[Ex]

"Center" Can Mean a Number of Things

When talking about a geographical center, one could refer to the center in terms of line of sight, road distances, driving times by car, or travel times by public transport. All these definitions will lead to different algorithms and formulas when calculating the center. Even if you knew you needed the center determined as the crow flies, you still have more than one option.

With more or less the same effort that many organizations keep spending on reinventing the wheel again and again, they could easily implement a number of these algorithms and let the solution itself find out which one works best (in terms of predicting accident frequencies). In an ideal world, all these algorithms would only exist once globally as a web service; interested firms could just use them for a fee. We are still a good way away from this vision becoming reality for many reasons.

Theoretically possible data flows

For all routines, procedures, and functions, the attributes and facts they serve and the kind of input data they need (not in terms of contents but in terms of semantically neutral data properties) should be clearly defined. Metadata, therefore, also need to embrace information such as what is considered suitable input data for procedures or the aggregations that are allowed on these data.

Returning to the image of a city map, this kind of metadata will define, for example, which routes you could take by car, factoring in one-way streets, roadblocks, and real-time traffic data. Following this analogy, the task of calculating optimum data flows is similar to determining routes via a navigation device. Each of these data flows has only one objective: building a bridge between real or potential values of attributes and facts (such as article group and trade margin last month) and value drivers (e.g., ESTM). The only difference between street navigation and data flows is that street navigation works with two dimensions, whereas data flows have to handle *n* dimension.

But even for home users there are software solutions and web services that can handle 3-D route optimization, so why not add a couple of dimensions now that we have a toy like SAP HANA?

When it comes to figuring out whether we should deliver judgments on the fate of products at an article group or segment level or whether such decisions should be based on previous trade margins or customer age, SYS doesn't have to rely on instinct or be hampered by historical practice any longer. It can simply let the system determine suitable data flows by trial and error. This is one of the key benefits of SAP HANA.

Tasks for Big Data

Constructing and evaluating data flows

Assuming that all the data and metadata needed are on the table, our big data solutions can take over four more steps in our workflow:

- **Draw samples**
  Even if we applied Occam's razor and only looked at that part of the space of options opened up by the data and metadata we have, we would still be taking huge strides. This is why we should only work with samples instead of all the data we have at hand.

  The other advantage of working with samples is that the rest of our data can be used to test the quality of the data flows the system is suggesting. Such quality tests can use a lot more data, because we are not exploring the whole space of options but only evaluating the routes the system has come up with.

- **Calculate hypothetical (allowable) attributes and facts**
  For these sample data, as many allowable attributes and facts as possible have to be determined, but because even big data solutions are not able to handle *unlimited* amounts of data, we should limit our scope.

  The areas we plan to examine should be linked to the value drivers we are interested in. The link between certain attributes and facts on the one hand and value drivers on the other is, however, hypothetical; hence, there is no way round a bit of deduction here. However, as processing power will keep growing (remember Moore's law), we will, step-by-step, be able to reduce the damage caused by human ignorance and/or hubris.

- **Inductively build decision trees**
  The preceding step will provide us with data records in which not only fields provided by source systems but also all attributes and facts that can be derived or calculated from them are filled. In our sample scenario, such a data record at the article-group level could contain the following characteristics and key figures:

  ```
  (Article Group; Trade Margin Last Month; Average Age of Customers; ESTM)
  ```

  In the preceding example, ESTM would be the value driver used to make decisions; the trade margin and the average age of customers buying items from that article group would a key figure and an attribute used to predict the behavior of that value driver.

On the basis of such records, SYS would expect its new solution to construct decision trees. Such decision trees are designed to answer questions of the following nature:

- Which attributes or facts have more discriminatory power (trade margin last month or average age of customers)?
- How many article groups should be used to classify articles?
- How many age groups should be used to classify customers?
- Can we expect an ESTM of more than 10% for a specific article group? If yes, how sure are we of this?

Alternatively, SYS could feed its algorithms with information like the following:

```
(Article; Trade Margin Last Month; Average Age of Customers;
ETSM)
```

In this case, the kind of information would be the same as before; only the granularity would be different. Once again, we could answer the same four questions related to discriminatory power and number of classes.

The preceding two data records are just two very simple samples from SYS's space of options. Nevertheless, each of them could be used to build at least one decision tree that could then be converted into a series of processing steps—that is, a data flow. Voilà!

Both data flows can be implemented automatically using virtual objects. They can then compete against each other, meaning that you can test their predictive power in terms of ESTM using the rest of your historical data. At the end of your comparison, you will know which data you need to look at and at what level of detail you should process them. If a higher or lower granularity is needed in the future, using it will be a piece of cake.

- **Consolidate decision trees**

  Building decision trees corresponds to step 6 in SYS's seven-point plan. After applying the preceding quality tests, SYS will be left with one decision tree/data flow per value driver. To the best of SYS's knowledge, this data flow will represent the best possible path through the space of options; "best possible" in this context means the one with the greatest discriminatory power.

Because we were building data flows per value driver, some of them may be pretty similar or even identical. There may therefore be a bit of room for improvement by merging data flows. To do this and to identify these data flows, we need to define what is meant by *similar*. We won't do that here, though.

[+]

Historical and Future Data

When constructing and evaluating decision trees/data flows, we are working with historical data, which means that we know not only the values of attributes and facts we could base our decisions on but also the outcome in the past resulting from the way those decisions were made.

Once we start implementing data flows, we obviously don't know the results of the decisions that will be based upon them. Instead, we must work with statements such as "ESTM for article group xyz is going to be > 10% with a probability of 65% and > 10% with a probability of 35%."

6.4.3 Building Blocks of the Solution

Required algorithms

In addition to a well-integrated, real-time metadata repository, we will need a couple of other tools to perform the job established in Section 6.4.2.

Draw Samples

Separate data into training and test data

The space of options can become huge, and a couple of factors, like dealing with continuous instead of discrete values, can make it even bigger. As we said previously, although in-memory databases have quite a bit of processing power they are not all powerful, despite the fact that a salesman might tell you that they can end all wars and end famine on the planet. Even if they were, we should not gobble up all our data to build decision trees, because this would leave us empty-handed when it comes to testing and rating them. With decision trees—and in general with *machine learning*, the high-level concept behind decision trees—data used to build a model or to teach it how to behave are called *training data*. Under no circumstances should the data used to train a system also be employed to evaluate it.

For these reasons, periodically drawing samples from the data replicated into our big data solution seems like a good idea. To help select data records for your sample, you can use tools such as the ones mentioned later in Section 6.5.1.

Calculate Hypothetical (Allowable) Attributes and Facts

Columnar data storage a perfect dream

For the data within your sample, you can then go ahead and calculate all possible or allowable attributes and facts. A columnar-storage approach, as used by SAP HANA, comes in very handy here; instead of picking parts of individual row-based records and laboriously rearranging them, a columnar database can simply pick the columns needed, combine them easily, and also quickly calculate aggregated values for each of them.

With each individual data record, routines, procedures, or functions that employ algorithms and formulas are used to determine the values of attributes and facts. Whatever language you use for this, be sure to keep the coding simple, go for a declarative programming paradigm, and document the result of your efforts within the metadata repository, or use a more abstract language, such as RDL, in which the coding itself represents the data model. The question of which attributes and facts you will have to determine will depend on your space of options, as defined under "Seven-Point Plan for Metadata" in Section 6.4.2.

Inductively Build Decision Trees

Algorithms for decision trees

As you may have figured out for yourself by now, in this chapter we are once again going for an inductive approach with the bare minimum of deduction when defining the space of options. When it comes to letting the system build alternative decision trees (data flows), we once again have the torturous task of selecting algorithms. The following ones are classic algorithms doing exactly that:

- **C&RT (also known as CART)**

  C&RT stands for Classification & Regression Tree, an algorithm developed in 1984. C&RT can only generate binary decision trees.

  One of C&RT's key advantages is that it is a hybrid algorithm that is able to process continuous data on a ratio scale as well as discrete or nominal facts. C&RT would, however, be of limited use for SYS, because it is restricted to producing only binary trees.

- **ID3**

  Iterative Dichotomiser (ID3) was developed two years after C&RT. This algorithm is primarily used to analyze huge volumes of data with many different key figures. It often leads to suboptimal, rather wide trees, especially when dealing with facts on a ratio scale.

- **CHAID**

  Chi-squared Automatic Interaction Detectors (CHAID) is an algorithm from the 1960s and is the oldest of the algorithms listed here. CHAID can process facts on ordinal and nominal scales and automatically stops the tree's growth before it becomes too big. Like C4.5 (discussed next), CHAID can construct decision trees with more than two children per node.

- **C4.5/C5.0**

  C4.5 is a successor to ID3. It was devised by the same developer (Ross Quinlan) and features some improvements over ID3—for example, it can handle continuous as well as discrete facts, and it also has the ability to deal with missing data. Trees generated by C4.5 are pruned automatically at the end of the process and therefore are more manageable.

  C5.0 is the successor to C4.5. It is a lot faster, needs far less memory, and leads to smaller trees, automatically excluding facts with little merit.

[+]

Selecting Decision Tree Algorithms

The nature or the properties of your data have a major impact on selecting a suitable decision tree algorithm. Some of them look at each attribute or fact in isolation, selecting the one that has maximum discriminatory power at each junction, whereas others can use more than one within a single junction.

Let's assume SYS needs an algorithm that can build decision trees for ESTM, like the one shown in Figure 6.3. In this case, you could either first consider historical trade margins and then the customer's average age, or do it the other way around, first looking at age and then at trade margins. Furthermore, the attribute "age" and the fact "trade margin" are both values on a ratio scale that can either be processed as they are or that can first be classified into an arbitrary number of groups (based on their low, medium, and high values, for example), leading to decision trees with a greater or lesser number of branches at each level.

Selecting the right variant in terms of the sequence of decision criteria and the optimum number of branches per level should be left to the system, the algorithm, and the discriminatory power of attributes and facts.

Furthermore, a creative data architect could have the idea to calculate a new key figure *(trade margin / customer age)²* that, after a couple of tests, turns out to be superior to every other criterion you were looking at so far in terms of its discriminatory power. In such a case, in addition to looking at individual attributes and facts, you'll also have to consider combinations of these or dependencies among them. If such dependencies are to be taken into account, the next question is how much effort you will invest. Do you only want to look at linear dependencies or formulas, or would you like to check polynomials of a second order (like the preceding example), polynomials of higher orders, or something even more complex? As you can see, your data scientists will not run out of work.

All of that may sound like a lot of backbreaking work, but how much time and money are you currently spending on fixing things? Believe us, a fraction of that effort will work wonders when directed toward new insights!

Consolidate Decision Trees

Decision trees for the same value drivers

From among those decision trees/data flows that are supposed to help us make decisions founded on the same value drivers, you can pick the best one by comparing, with hindsight, the decisions they would have led to. As defined in Section 6.4.2, the "best" decision tree is the one that would have led to those decisions that maximize shareholder value. We do admit that this evaluation is based upon historical data, but unfortunately this is all you have.

Decision trees for different value drivers

After that, you will still end up with one decision tree per value driver, but, as stated previously, many of the resulting data flows will be pretty similar. Using the framework we defined under "Seven-Point Plan for Metadata" in Section 6.4.2, all metadata related to your decision trees will be available right away. Even better, information about what they do and how they do it will be available in a consistent and structured format based upon one and the same reference system, your space of options.

As a result, you will be able to automatically compare the corresponding data flows, quantifying their similarity—something you can probably only dream of doing with your existing data flows. In addition to static measures of similarity, you could also think about using the clustering algorithms like k-Means, mentioned in Chapter 2, Section 2.1.1, and if you find two or more doing the same thing you might want to consolidate data flows or related domains and layers.

6.4.4 Potential Benefits and Value Drivers

Value drivers related to data flows

The value drivers we are interested in in this section are not ones such as ESTM that SYS needs in order to make decisions about its future assortments. This book is not about product mix policies in retail but about making decisions about big data business cases. The value drivers relevant here can be derived from the problems discussed in Section 6.3.

If metadata are always up-to-date and generally available, if data flows evolve on the basis of automatically generated decision trees, and if they can be adapted to new circumstances quickly or even automatically, then all of that will have at least four positive effects:

- **Accuracy when depicting reality/flexibility**
  In Chapter 7, we will show that the environment companies are operating in has become extremely unstable. Decision-relevant frameworks are no longer changing over decades or years but monthly, daily, hourly, or even every minute. A world in which decisions are made on the basis of static reports or data flows is fast becoming obsolete, a cloud-cuckoo-land for management that has very little relevance to the world out there. Modifying data flows manually is taking longer and longer; sooner or later, organizations that rely on such processes will reside in a world of delusion, no longer able to get their bearings in reality.

  So far, most organizations are trying to resolve the issue by using more and cheaper offshore resources, in a way becoming addicted to outsourcing, needing ever-higher doses of it. However, easing cost pressures by outsourcing will hit an inevitable finite limit, and if a growing percentage of your IT resources are more or less continuously trying to reconcile numbers, you will no longer have the capacity for real progress, regardless of whether you outsource or not.

  The approach presented here—developing metadata models that generate data models—will free you from these constraints. This value driver increases shareholder value by helping you get a clear view of reality, by adapting to changes cheaper and faster, and by providing you with a sustainable and scalable model for future growth. It will enable you to use new opportunities faster or even see them for the very first time.

- **Lost income/expenses due to incorrect decisions**
  When talking about how we view reality so far, we have focused on how flexible data models are and how fast they can adapt to changes. A better perception of reality will, however, also help you avoid costly mistakes.

  Even if your systems can adapt to a changing reality quickly, we still don't know whether they have adapted to the *right* reality. Maybe you have adapted to an illusion instead. With big data, you are able to consider many conceivable realities in a very short time, thus giving you a better chance to hit the right one. One example of this is as follows:

  - All decision tree algorithms listed under "Inductively Build Decision Trees" in Section 6.4.3 are controlled by certain parameters.
  - If you are able to set a very large number of possible values for these parameters within a very short period of time, then you will end up with a huge number of decision trees and data flows serving the same purpose. This widens your horizon dramatically.
  - When making decisions, you are then able to take into account a huge number of possible realities; your chance of finding the right one are probably higher than if you only used one set of controlling parameters you just happened to come up with.

  We assume that shareholders will reward a business's ability to consistently make fewer mistakes, to avoid related costs, and to take advantage of opportunities that others aren't aware of. That is why we are considering a better perception of reality a value driver.

- **Reconciliation costs**
  The man-hours needed to reconcile different results from different redundant reports, data marts, and data flows will disappear. There will only ever be one data flow that determines values for attributes and facts, and this data flow will be clearly documented in your metadata repository. What's more, every value driver relevant for decisions will be clearly assigned to underlying data flows.

Burdens of the past, such as redundancies, can be eradicated by cross-system comparisons that measure the similarity of data flows using text analysis (see also Chapter 8); new unwanted duplication can be avoided by automatically checking generated decision trees for similarities right from the outset.

All of these advances increase shareholder value by reducing personnel costs in IT and finance.

- **Development costs**

  Data models that are free of redundancies, consistent, and to an increasing extent generated automatically not only help you improve your reaction times but also reduce adjustment costs. You will benefit from two effects:

  - Higher transparency through unified procedures and better metadata
  - Automation of what was previously manual development work and at the same time automatic updates of related metadata

  Once again, shareholder value can be raised by reducing personnel costs, this time primarily in IT and mainly related to improving existing data models and developing new ones.

Automation in production

The industrial revolution started with machines enabling new manufacturing processes and then taking them over altogether. In parallel, these manufacturing processes were standardized, and modeling, designing, and improving production processes became a discipline in its own right. Quite a while ago, things went even further; many decisions needed in the course of a production process are no longer made by humans but by algorithms within manufacturing execution systems.

Automation in administration

In a similar way, computers made it possible to support administrative business processes using IT solutions. Subsequently, computers took control of the whole process, as is the case with most payrolls, for which they not only calculate salaries but also send out paychecks electronically without any human intervention. Today, administrative decisions, such as when and how to send out payment reminders, are also made automatically, mostly based upon a static set of rules.

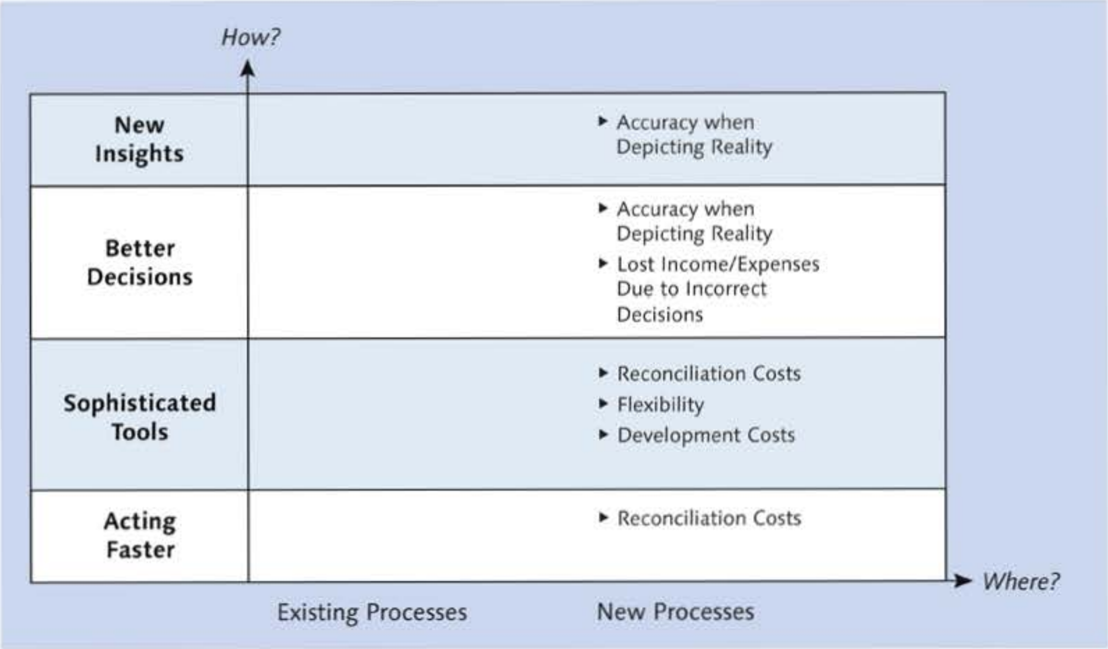
Automation in system design

Big data is bringing the next logical step into range: automating the design and the development of decision-supporting systems like products for data exploitation and products for data generation.

System design and software development, once an art, became an engineering science a few decades ago, but when talking about systems designing systems we are not talking about declarative languages, modeling tools, or metalanguages that can generate code. What we are talking about is automation at the very heart of system design. With the vision we have in mind, developers no longer design or implement data models; instead, they design metasystems, solutions that can generate and, if needed, modify data models. In such a scenario, resolving capacity issues via outsourcing would be as foolish as deploying more bricklayers to build a skyscraper faster.

Assigning value drivers

Figure 6.8 provides you with an overview of the value drivers we just discussed, assigning them to the cells in our benefit–value driver matrix. We have decided to put all value drivers on the right-hand side under New Processes. Although data modeling mostly deals with existing business processes, the idea of at least partially automating the generation of such data flows would be a new concept and thus a new business process for most IT departments.



	Existing Processes	New Processes
New Insights		▸ Accuracy when Depicting Reality
Better Decisions		▸ Accuracy when Depicting Reality ▸ Lost Income/Expenses Due to Incorrect Decisions
Sophisticated Tools		▸ Reconciliation Costs ▸ Flexibility ▸ Development Costs
Acting Faster		▸ Reconciliation Costs

Figure 6.8 Data Models Benefit–Value Driver Matrix

6.5 Implementation Scenario and Framework Architecture with SAP HANA

Criteria to select implementation scenario

When selecting an implementation scenario for SYS, three considerations come to the fore:

- First, metadata play an important role in our concepts. Hence, SYS will need a metadata repository that is very well integrated with all of its applications and also with SAP HANA.
- Second, developing data models automatically and then implementing the resulting data flows by hand doesn't seem very clever to us, at least not in the long run, so a close integration between SAP HANA and OLTP systems is key.
- If SYS already has a data warehouse like SAP BW, and if most relevant reports are produced there, this data warehouse should be used. Creating a parallel universe in SAP HANA by utilizing, for example, the data mart scenario would be counterproductive.

Blurring the boundary between OLAP and OLTP

The second point is based on the following thought: data flows generated automatically and set up in SAP HANA are not only there to provide the data for reports but also to control business processes. Hence, SAP HANA and operational systems like SAP Business Suite working in close harmony is possible when correctly designed. In Chapter 7, Section 7.5.1, we will discuss how to implement this idea ("Products for Data Generation").

6.5.1 Implementation Scenario and Framework Architecture

Scenario for SAP customers

For customers using SAP Business Suite, the SAP Business Suite on SAP HANA scenario (see Figure 6.9) is the obvious choice. Non-SAP customers who only want to use SAP HANA for the time being will at least temporarily have to follow the data mart scenario.

Extending the scenario

As with other case studies, you will not get around some amendments and add-ons to the vanilla implementation scenario, which is why we will now take a closer look at Figure 6.9.

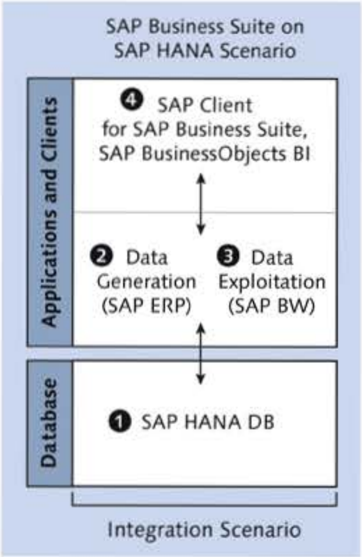

SAP Business Suite on SAP HANA Scenario
Applications and Clients
❹ SAP Client for SAP Business Suite, SAP BusinessObjects BI
❷ Data Generation (SAP ERP)
❸ Data Exploitation (SAP BW)
Database
❶ SAP HANA DB
Integration Scenario

Figure 6.9 Data Models Implementation Scenario

Databases ❶

Consistent data and data models

One of the main advantages of the SAP Business Suite on SAP HANA scenario is that all the data needed are stored in one single, integrated SAP HANA database. In our case, this not only leads to great computing performance in SAP Business Suite but also creates the ideal habitat for consistent data and data models. A single database can also contain redundancies and inconsistencies, but our experience shows that data model pests thrive a lot better in heterogeneous systems that have many hidden, dusty old corners that haven't been swept out for ages.

Products for Data Generation ❷

SAP Business Suite generates the data

As mentioned at the start, we are primarily focusing on customers who are using SAP Business Suite, which in this case is also the primary product for generating data. In particular, the option to implement decision tables in SAP HANA makes merging data generation and data exploitation a very compelling option. Areas ❷ and ❸ in Figure 6.9 are therefore candidates for merging in the future. For obvious reasons, buying both products from the same supplier facilitates this process.

In this integrated implementation scenario, deciding whether decision logic is held within SAP HANA using decision tables or whether SAP HANA is providing data to be used to fill customizing tables within SAP Business Suite becomes less important. The first variant is certainly more flexible; it will, however, also necessitate changes within SAP Business Suite (adaption/development by SAP or via customer enhancements).

Products for Data Exploitation ❸

As stated in Section 6.2, SYS uses SAP BW as its data warehouse for all its subsidiaries (SAP BW is not shown in Figure 6.9 because it is not necessarily contained in the pure version of SAP Business Suite on SAP HANA as defined by Figure 2.17). We assume that this data warehouse delivers the majority of all operational, strategic, and legally required reports and that it will therefore continue to exist in all possible future scenarios.

SAP BW's capabilities not sufficient

Unfortunately, SAP BW comes with very little functionality suitable for fulfilling the requirements listed in Section 6.4.3. SAP BW's data-mining tool, Analysis Process Designer (APD), is capable of constructing decision trees but can only do so via one single algorithm. We have yet to meet a real-world customer using Analysis Process Designer for demanding data-mining applications.

Data exploitation within SAP HANA

Within SAP HANA, on the other hand, there is a whole arsenal of readily implemented tools for data exploitation (Figure 6.9, ❸):

- **Draw samples**
  In Chapter 7, Section 7.5.1, we will extensively discuss how to properly size and draw samples via SAP PAL or via R.
- **Calculate hypothetical (allowable) attributes and facts**
  When determining or calculating hypothetically determinable attributes and facts, we are computing all allowable supplementary and derived attributes and facts for the data records within our sample. Which attributes and facts have to be determined is defined by the metadata mentioned under "Seven-Point Plan for Metadata" in Section 6.4.2 and is defined also by the data themselves—that is, the attributes and key figures delivered and the values sitting in these fields.

To create the required data structures and procedures, the new language RDL, mentioned in Chapter 2, Section 2.1.3, is a strong contender; we will take another look at this option in Section 6.5.2. For simple algorithms, we would recommend using SQLScript due to its lower level of abstraction. You might also need additional tools, such as SAP PAL, R, Calculation Engine, and Planning Engine, when determining values for attributes and facts. The SAP HANA Planning Engine is a collection of functions often needed for business planning.

- **Inductively build decision trees**
  SAP PAL supports the construction of decision trees via CHAID (`CREATEDTWITHCHAID`/`PREDICTWITHDT`) and via C4.5 (`CREATEDT`/`PREDICTWITHDT`). In R, your options go far beyond these two algorithms. A couple of packages provide you with everything you could ever wish for. C&RT comes via package `rpart`, C4.5 via `RWeka`, and C5.0 via package `C50`. But because decision trees are only one special case of machine learning, R does have a lot more to offer; an impressive overview can be found at *http://cran.r-project.org/web/views/MachineLearning.html*.

- **Consolidate decision trees**
  Once decision trees have been built, the remaining data that have not been used so far can be selected to compare a couple of trees, helping you with decisions related to the same value driver. To do this, you let your decision trees predict (in SAP PAL, `PREDICTWITHDT`) what could have happened in the past, compare these predictions with historical reality—for example, using a chi-squared test for goodness of fit—and select the decision tree with the best predictive performance.

  Alternatively, you may use quality criteria to evaluate decision trees. Quality criteria are used to quantify the forecasting performance of decision trees; there are an almost endless variety of them. In R, apart from statistical tests and quality criteria, you also have metaalgorithms that can be used to optimize the parameters controlling various underlying algorithms (package `e1071`, function `tune`).

  In addition to all of this, you still face the challenge of consolidating decision trees linked to different value drivers. For this, you can use clustering algorithms (in SAP PAL, `KMEANS` or `SELFORGMAP`, in R, function `kmeans` or package `kohonen`, and so on). Because your decision trees/data flows for different value drivers will have little in common in terms of depth and width, classifying them as a whole will not take you very far. When categorizing data flows, it would be better to rely on semantically neutral properties (such as the number of layers or domains they pass, properties of the procedures within them, and so on). You probably won't need text analysis for those that have been created automatically as you do with manually created data flows.

Clients ❹

Dashboard(s) for the control room

In this chapter, we are designing a more or less autonomous system. You will therefore need some kind of control room to help you monitor this level of automation. Such a control room—maybe in the guise of one or more dashboards—will be able to accept and process data that are related to the automated generation of data flows. Examples of such data include the following:

- Control parameters for algorithms that generate decision trees
- Number of decision trees generated/consolidated
- Aggregated quality criteria (before and after consolidation)

IT in the future

We are still quite a long way from the vision of an IT department being as deserted as a cold-rolling mill in the steel industry. In this chapter, we are trying to anticipate new opportunities and perspectives that may become an everyday reality in some 20 years or so. So far, you still need people to do the following sorts of things:

- Check and evaluate proposed data flows not just individually but within your organizational context
- Implement new data flows or at least supervise their implementation
- Check and judge the quality of metadata

To be up to that job, your experts will need clients to display or modify metadata and data models. Examples of such clients include the following:

- The SAP Information Steward frontend (when dealing with metadata).
- The ARIS (architecture of integrated information systems) platform; a cheap and easy way to dip your toes into data modeling is via ARIS Express, available for free at *www.ariscommunity.com/aris-express*.

- Any other tool used for data modeling (for example, tools for entity-relationship modeling [ERM], one of many ways to define data models); ideally, you should use tools that leave you with only a small gap to bridge in terms of implementing these models—for example, RDL.

6.5.2 Data Architecture

Two domains to start with

To get closer to the ideal of self-developing systems, we need an architecture that rests on at least two pillars, domains, or groups of domains:

- **Abstract space of options**

  In Section 6.4.2, we introduced the concept of a space of options spanned by all dimension IDs, attributes, and facts that are either contained in your source data or can be derived or calculated from them using whatever procedures you may have implemented.

  This space of options can be precisely defined using modeling tools or metalanguages, such as RDL. RDL can, for example, be used to declare the following object types:

  - Entities
  - Derived entities—that is, entities based upon other entities
  - Associations between entities, like *n*:1 or *n*:*m*
  - Actions on entities, describing the logic needed to determine attributes and facts from existing data

  Have you ever engaged in entity-relationship modeling, or have you ever created a database in Microsoft Access? If you have, then some of the previous terms may sound familiar to you, but if not, then look for books on ERM or OLAP modeling from your favorite online bookseller. Even better, find a subordinate who likes this sort of thing and ask him or her for a summary, and go and have a beer.

  Exactly which specific objects (that is, which entities, associations, or actions) are required to span the space of options can be found from the information in your metadata repository. Essentially, you need everything listed under "Seven-Point Plan for Metadata" in Section 6.4.2. Depending on the layout and complexity of your metadata repository, you could even think about generating the required RDL code automatically.

- **Concrete space of options (instantiated)**

  If you had the previously mentioned kind of code, spanning your space of options in RDL, it would be compiled as soon as you activated it. In the course of the compilation, the required objects, such as tables, procedures, or services for data access via OData, would be created in SAP HANA. Using the implemented data model, you could now draw a sample from your source data and have all allowable attributes and facts calculated for that sample, as described in Section 6.4.3.

  The tables generated by RDL within SAP HANA will be used to store data that have been extracted from source systems (dimension IDs, such as the article number, or facts, such as net sales) or that are to be determined within SAP HANA (attributes, such as the article's subgroup, or facts, such as last month's trade margin) via procedures that have also been generated by RDL. Thus, the abstract concept of a space of options leads to a specific instantiated concept that uses the data of your sample. With this approach, you no longer need to manually create the required objects within SAP HANA or modify them with every change you notice in your environment.

Using the concrete, instantiated space of options as a starting point, you can now have SAP HANA build, test, and consolidate decision trees. Whether you then test them using yet another sample or all remaining data within your population will only depend on the computing power at your command. If you opt for another sample, you need to make sure that both training and test data are *disjoint* (meaning their intersection is the empty set).

Sample architecture

Figure 6.10 shows a proposal for an architectural model corresponding to the preceding ideas about two domains and how to use the data within them. The individual processing steps we talked about are implemented in separate layers. Please keep in mind that this is nothing but a rough draft skeleton. RDL, for example, offers a lot more than the simple elements we discussed here. In the following sections, we will explain some of the layers within Figure 6.10.

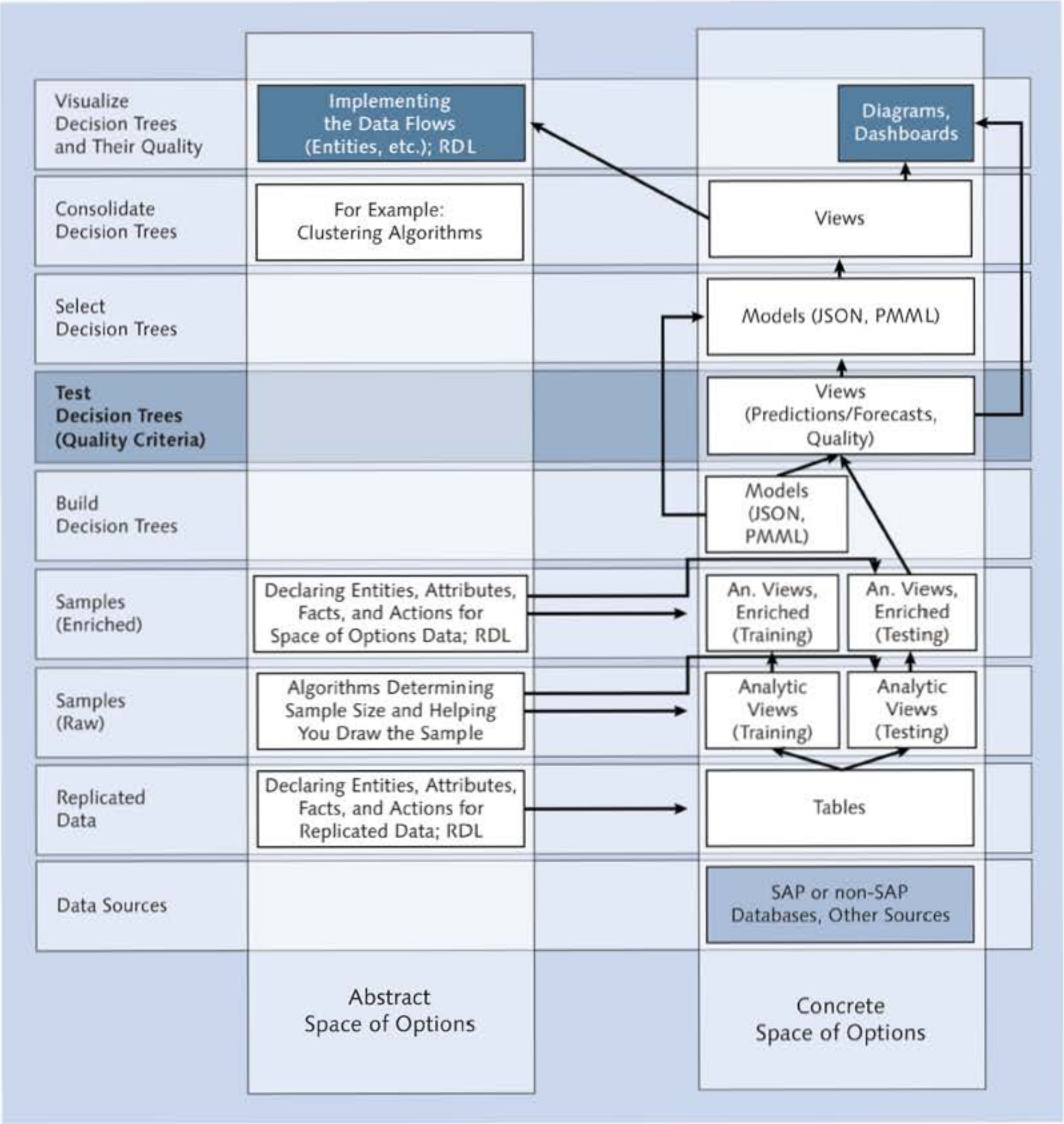

Visualize Decision Trees and Their Quality
Implementing the Data Flows (Entities, etc.); RDL
Diagrams, Dashboards
Consolidate Decision Trees
For Example: Clustering Algorithms
Views
Select Decision Trees
Models (JSON, PMML)
Test Decision Trees (Quality Criteria)
Views (Predictions/Forecasts, Quality)
Build Decision Trees
Models (JSON, PMML)
Samples (Enriched)
Declaring Entities, Attributes, Facts, and Actions for Space of Options Data; RDL
An. Views, Enriched (Training)
An. Views, Enriched (Testing)
Samples (Raw)
Algorithms Determining Sample Size and Helping You Draw the Sample
Analytic Views (Training)
Analytic Views (Testing)
Replicated Data
Declaring Entities, Attributes, Facts, and Actions for Replicated Data; RDL
Tables
Data Sources
SAP or non-SAP Databases, Other Sources
Abstract Space of Options
Concrete Space of Options

Figure 6.10 Data Models Data Architecture

Data Sources

Any kind of source

We will not discuss data sources and data acquisition in detail here (further information about this can be found in some of the following chapters).

This is why we are only talking about SAP and non-SAP databases or other sources in general terms. By "other sources," we mean all data outside of classic, relational databases—for example, Apache Hadoop via SAP Data Services, SAP Event Stream Processor, and so on.

Replicated Data

Persistent tables

In this case study, we assume that all source data needed to build decision trees/data flows/data models are replicated regardless of where they come from or whether they are already in SAP HANA or not. The only reason for this is traceability.

This is why we allowed for persistent objects within this layer. The structure of these tables can be derived from the source data's metadata. Keep in mind that some fields might not (yet) need to be replicated into SAP HANA, although you should be generous rather than sparing in that respect. Creating the instantiated corresponding objects in SAP HANA can be done via RDL.

Samples (Raw)

Analytic views to look at data in samples

You then draw samples from your replicated data (see Chapter 7, Section 7.5.1 for applicable algorithms). Some of these samples will be used to build decision trees and others to test and evaluate them, which means we have opted for testing via samples rather than testing the whole population in this example. For these samples, we don't need persistent data; analytic views will do. Within SAP HANA, an analytic view is the equivalent of an OLAP cube; the structure of an analytic view corresponds to the star schema shown in Figure 6.7. Figure 6.11 shows a sample analytic view derived from the sample star schema in Figure 6.7.

Example for an analytic view

In the center of Figure 6.11, you'll find a line item table (CE1IDEA) from CO-PA (acting as the fact table). Above that table, we have arranged two possible dimensions (article/PRODUCTS and customer/CUSTOMER) that are connected to the fact table via their respective keys (PRODUCT_ID, CUSTOMER_ID). The relations among these three tables are defined via a LOGICAL JOIN (window area SCENARIO). When defining the database (called DATA FOUNDATION), you could have selected only those data that belong to your sample via a field such as BELNR (document number). One practical tip in this context: you need to mark your fact table as CENTRAL ENTITY under PROPERTIES • GENERAL so that SAP HANA recognizes it as such.

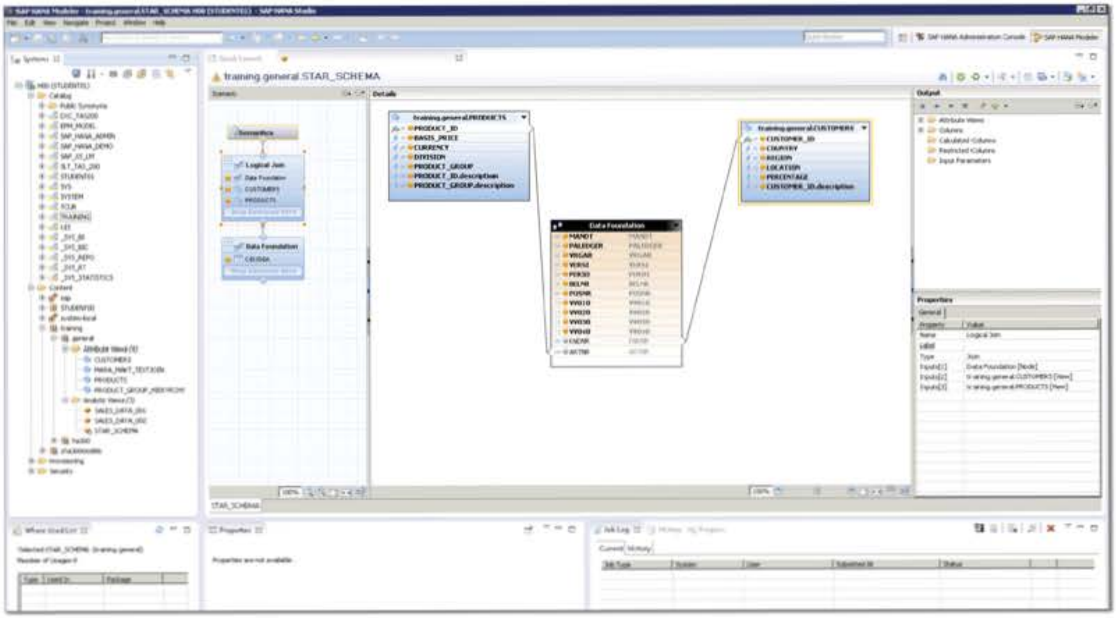


Figure 6.11 Analytic View with Fact Table and Dimensions

Samples (Enriched)

The sample data from the preceding layer are now enriched with the attributes and facts that cannot be found in the source data. This means that you'll have to create and fill other, extended data structures within SAP HANA. To determine attributes and facts that don't come from the sources, you could use actions in RDL.

Test access to RDL development environment

If you would like to play with RDL, you can get a free seven-day access to a development environment by registering under *http://sap-river.com/*. At the moment, this only works with Google's Chrome browser.

Build Decision Trees

Models stored in JSON or PMML

Enriched sample data will be used to create decision trees via the algorithms presented under "Inductively Build Decision Trees" in Section 6.4.3. If you are using SAP PAL for this, the results will be models that can be saved in the data format JSON (JavaScript Object Notation) or PMML (Predictive Model Markup Language). JSON is the format of choice if you want to use the models to make predictions using SAP HANA/SAP PAL; PMML would be used to hand your models over to other statistics solutions.

In most cases, you will need JSON in SAP PAL, and PMML is an additional option. R can produce and process both formats as well; JSON would be generated via package `rjson`, and for PMML you would use package `pmml`. Apart from these two special formats, R can also deliver the parameters of a model as a table.

Test Decision Trees (Quality Criteria)

Check predictive power

To compare a number of decision trees/data flows that are meant to support you with regards to the same decision or value driver, the models stored in JSON are fed with the test data from one of your test samples. This will lead to views containing *ex-post forecasts* (that is, predictions with hindsight) for the relevant value driver or value driver–related key attribute/fact. These ex-post forecasts are then checked against the actual values also contained in your test data, calculating quality criteria and quantifying the predictive power.

Select Decision Trees

Exchange decision trees as and when needed

The decision trees with the highest predictive power are used to create data flows, whereas worse performing alternatives for the same value driver or decision are rejected. At this point, you'll once again see the flexibility of our proposal: perhaps the criteria (attributes and facts) that best identify article groups with a high potential ESTM today are different from the criteria you should use tomorrow and different again from the ones that were suitable last week.

In a classic environment, you would hardly ever notice this change, let alone be able to implement it quickly. If you were to become aware of it, you would face lengthy reconstruction work on your data model. With our approach, you simply need to trigger the modeling algorithm from time to time.

Consolidate Decision Trees

Filtering out redundant data flows

Each of the decision trees/data flows you are left with at this stage does refer to a specific value driver or decision to be made, but some of them might still be pretty similar, meaning that you could consolidate them to make your data model leaner. Trying to automate this whole process would be a bit ambitious, but automatically identifying some of the more similar ones does not mean reaching for the stars. As mentioned before, you may use clustering algorithms from SAP PAL or R for this and then manually cross-check candidates for consolidation, or you can forget about this step altogether; if you know that many data flows will change in the future anyway, then it might not be worth the effort.

One of the main hurdles of using clustering algorithms is that they cannot process models stored in JSON or PMML. You therefore first need to convert the models into a format these algorithms are able to digest and also establish assumptions that determine under which circumstances two data flows are considered similar.

Attention! This is not only pretty tricky, but you are also getting into deductive thinking at this point!

Visualize Decision Trees and Their Quality

Check automatically generated data flows

Probably only a few of us are willing to blindly trust a self-developing system. This is why you would probably like to inspect automatically generated decision trees and data flows before implementing them and allowing your SAP Business Suite to send out payment collection letters on that basis. Maybe you also want to know why and on what basis—meaning using which quality criteria—the algorithm chose a certain decision tree over another one.

For this, you need appropriate clients. These clients are solutions for data modeling, which are a subset of the products for data management, as defined in Chapter 2, Section 2.1.3; these clients can visualize data flows and also process input in PMML or XML. On top of that, you'll want dashboards to serve as some kind of control center for your system, which is why we have added a client layer/domain for human users in the top-right-hand corner of our data model.

Implementing data flows

Furthermore, the resulting data flows still have to be implemented. We have also established this functionality on the client layer. However, because the recipients for this functionality are not humans but machines that we would be talking to in metalanguages (such as RDL or XML), you'll find it in a different domain in the top-left-hand corner of Figure 6.10.

Brute force

Does our general approach look familiar to you? Perhaps you have encountered something like it before, maybe in one of your computer classes some 30 years ago? That's right! What we are describing here resembles—at least in some respects—the good old brute-force method. This was an approach that was used in the early days of computing—for instance, with computers playing chess. Indeed, we assume that the brute-force approach is still doing a good job in terms of cryptography for many secret services all over the planet.

[»]

Brute-Force Search

Brute-force search, sometimes called by its more friendly names *exhaustive search* or *generate and test*, refers to a technique that simply goes through all action alternatives for a certain decision (like the next move in a game of chess), evaluating them in some shape or form and then picking one of them on that basis.

Brute-force search may look a bit primitive on the face of it, but it still has its practical use in cryptography or, more precisely, in cryptanalytic attacks, in game theory, in computer games, and generally in complex optimization problems. One important application area is route planning for delivery services or field staff: the traveling salesman problem.

The great advantage of brute-force search compared to less performance-demanding heuristic algorithms that use estimates, assumptions, or approximations is that if you really go through all and not just some options you can be sure to find the best possible solution, not getting stuck with the second- or third-best one. Mathematically speaking, you are going after the global optimum instead of local optima.

In principle, our idea is not revolutionary. The chief attraction is the fact that we are using brute force for a job that wasn't solvable automatically at all, not too long ago. The main problem with exhaustive search is that it pushes systems to their limits unless you restrict yourself in terms of the desired *depth of calculation* (for example, the number or reach of potential next moves you are precalculating or considering in a game of chess).

SAP HANA enables implementation

Big data does not remove that limitation but definitely moves the limits of what is feasible or reasonable. This also leads to a new insight about SAP HANA; due to the appliance's enormous performance, brute force can now be used for many problems that only had heuristic solutions before or none at all. Quite a few heuristic algorithms that have been very useful so far will soon join logarithmic tables and slide rules at the museum of ancient technologies.

What is a cynic? . . . A man who knows the price of everything and the value of nothing.

Oscar Wilde, Lady Windermere's Fan

7 Managing Customer Behavior

Derek was impatiently moving around in his seat. Since early morning, his group had been in their van, shaken to pieces on the gravel roads through the Namib Desert. To be at the dunes in time for sunrise, they had to get up at 3 a.m. This was one of the main reasons why the strong coffee had been well received; Derek had washed down his muesli with three large cups. The caffeine had woken him up, but those brown beans also had a lot of acid and digestive hormones attached to them. With the dunes, his stomach had already started to grumble. Images of a brand new motorway station with giant, shiny bathroom facilities were swirling in his head.

Focusing hard, he kept look-out for a couple of trees or a bush, being well aware of the fact that this could become a little tough in the middle of a desert. What irritated him a bit though was the thought that the odd African puff adder might also have thought of a joyride beside the track and that the eyes of all the other passengers would follow him to his place of solace.

A sudden reduction in the van's speed called Derek's attention to a tent-like construction. George—their local driver—was about to stop right in front of it. It soon became clear this was indeed the only public convenience within a radius of 170 miles. Clearly, George had spotted the misery in Derek's eyes and therefore pulled over without Derek having to ask.

A few minutes later, Derek returned, clearly relieved but still not in high spirits. On entering the tent, he had learned that the national park's management had set the price for using the toilet at $10, about 15 times the average hourly wage in Namibia. After listening to Derek's rant, George—whose main occupation was not a bus driver but a management student at Windhuk—had cheerfully responded that the value of the service determined its price.

Figure 7.1 Public Convenience at Sossusvlei, Namib-Naukluft National Park, Namibia

After a short cool-down phase in the air-conditioned bus and some time to think about it, Derek indeed had to agree that the values of goods and services weren't based only on the products themselves; customer needs were key. When waiting at an airport, he often treated himself to a beer, well aware of the fact that once he got to his destination, he could buy the very same beer at a discount store for a tenth of the price. "Customer needs" to him suddenly seemed like a simple expression to describe a pretty complicated concept. His willingness to pay for drinks was determined by where he was, which other alternatives there were within a certain future timeframe, the amount of other food or drinks he had consumed within a certain time in the past, how much he wanted them, and what his mood was like. Intrinsic product attributes such as "quality" and "taste" were well down the list.

The better businesses are at understanding the factors that determine their customers' needs, the more profitable they can become. Inwardly, Derek had to admit that, a few minutes ago, he would have gladly handed over $150.

Understanding purchasing decisions

Purchasing decisions are, first and foremost, decisions, which doesn't necessarily mean that they are rational in any shape or form. When we have to decide something, our minds often travel in mysterious ways.

Hidden opportunities

In this chapter, we will first of all explain why people strive to influence the decisions of others. Then we will introduce you to our sample company for this chapter, a retail group called Leech Oil. Leech Oil spends a lot of money on market research and has gone a long way in that respect.

Although the company is often hit by unexpected customer behavior that does not fit into existing models, it is just this sort of situation that can contain hidden opportunities.

In this chapter's solution section, we will once again analyze affected business processes within SAP Solution Explorer, suggest functional requirements, discuss algorithms that are independent of whichever products you use, and have a closer look at potential benefits and value drivers. Finally, we will select applicable implementation scenarios and see whether the algorithms we suggested are available within SAP HANA. We will close this chapter by looking at a few more data architecture-related aspects, this time focusing on the transfer of decision logic from business software into SAP HANA.

7.1 Understand, Predict, and Manage Customer Behavior

Objective: influence behavior of others

The urge to understand the behavior of other individuals, to then predict it using behavioral models, and finally, to control it is probably as old as mankind itself. Over the centuries, scientists from various disciplines (economics, sociology, psychology, and so on) have wrestled with the questions of *why* people do what they do and why they do (or indeed sometimes don't) act at all. A few scholars might have been striving for nothing but wisdom for wisdom's sake, but a lot of that research was driven by the desire to understand, model, and influence human behavior.

7.1.1 Example: Demand Curve

Price and demand

The demand curve is a classic example of such a model.

[«]

Demand Curve

A *demand curve* expresses the dependency between a couple of parameters and demand in a mathematical format. In its most simple form, the demand curve depicts a linear dependency between an item's price and the quantity demanded on a certain market. It then can be expressed as $d = a * p + b$. In this formula, d stands for the demand and p for the price. a and b are market- or product-specific control values (see Figure 7.2).

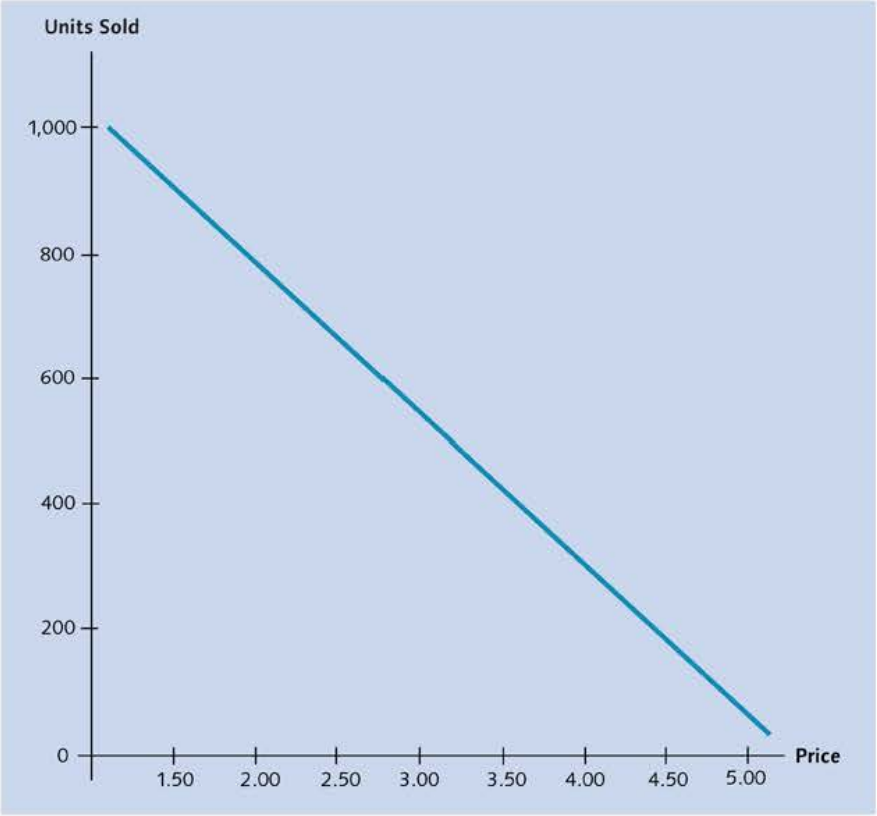


Figure 7.2 Linear Demand Curve

Nowadays, economists and psychologists have long left behind the simple model shown in Figure 7.2, mainly because they observed a lot of things this model cannot explain. For example:

- Consumer enthusiasm for Apple's smartphones and tablets cannot be explained by these products being the cheapest ones on the market.
- We often pay more for gas than we have to because we simply don't know that a gas station two blocks away is offering the same product at a much lower price (two keywords in this context are incomplete information and market failure).
- A gas station in the middle of the Australian outback can offer its product at more or less whatever price it likes without having to fear any negative effect on demand. Customers probably won't have

enough fuel left to drive to the next station, and even if they did the costs of doing so would not justify the effort (the keyword here is transaction costs).

7.1.2 Better Models through More Parameters

More data = better models?

It seems obvious that models become more realistic if more parameters are taken into account. However, you have to presume the following:

- The parameters you are adding do indeed have some kind of explanatory power.
- You have enough computing capacity to handle more parameters/ data.
- You have properly judged the impact of these parameters on whatever you are trying to predict. (We discussed this extensively in Chapter 4, Section 4.4.)

Hotel: catalog versus reality

Imagine you were working with a tour operator. Based upon a given price per night, you will have to forecast how many of your customers are going to book that specific resort next year. A bit of research on travel portals such as TripAdvisor or HolidayCheck takes you to videos recorded by former guests that show unappetizing breakfast buffets and stained lounge chairs. A look at a satellite picture on Google Earth reveals that a six-lane highway separates the hotel from the beach and that you can only cross this street via a pedestrian tunnel five miles to the west of the hotel. Had you visited the hotel, you would have also found out that homeless migrant workers are using the tunnel as a place to sleep, that the beach is littered with garbage, and that giant mosquito swarms reside at the beach toward the end of summer. Nevertheless, some of your customers might still want to stay there, maybe because two pitchers of sangria per night are included in the price—but the demand curve and the price sensitivity of these customers may be slightly different from the target group you originally had in mind.

Models explain the past

You cannot build models that are based upon the future, because you don't know what the future will be like. What you can do, however, is to move your models a bit closer to "now"; taking into account as many parameters as you can could be one step on your path. The closer you get to real-time modeling, the better you will become at managing customer behavior via intelligent pricing or communications.

This in turn means that big data and SAP HANA are going to open a door to a brand-new world of modeling customer behavior, a world you didn't even know existed before. In the blink of an eye, you can now process data that would previously have kept your systems busy for days. You can use brute force (as described in Chapter 6), plus trial and error, which means that data analysis can suddenly follow an experimental approach, interacting more closely with your customers.

All of that sounds great already, but we have only talked about first steps so far. Wait until you see the rest of the chapter!

7.1.3 Dynamic or (Inter) Temporal Customer Segmentation

Using up-to-date data

In the past, contemporariness was one factor that was often ignored due to seeming neither important nor valuable when developing models that represent customer behavior. When analyzing data records—for example, from point of sale (POS) systems or billing solutions for telecommunications or electricity firms—people are dealing with relatively high volumes of data, millions or billions of records per day. Even very fast and scalable data warehouses like SAP BW could, up to now, only handle such volumes in batch mode.

[»]

POS System

The term *point of sale* refers to the physical location at which a retail purchasing transaction is taking place. When using the term POS system, we refer to all IT solutions that collect data about sales transactions or interact with customers. Such systems may reside in a store, within a vending machine, or on a website; they can interact with customers directly or indirectly—that is, via sales staff.

Behavioral patterns are changing

Because of batch processing, insights into customer behavior were not available before the next morning, and building whole new models under these circumstances wasn't a job that could take only a couple of hours; rather, it was a project absorbing weeks or months, which means that the behavior of customers in January might have been modeled by the middle of the year, at which time behavioral patterns that were derived and modeled had most likely changed. The results were of nothing but historical value for data archaeologists. Predicting future customer behavior on such a basis was no longer acceptable.

Questionable assumptions of stability

The processes determining customer behavior are often not stationary (see Chapter 4, Section 4.4.3 for a definition). We may be dealing with different customer groups at different times, or customers may behave differently in the afternoon than they do at 8 a.m. Behavioral patterns are a lot less stable than most organizations would wish. They don't only change from week to week or day by day but often within hours or even minutes. This is exactly the problem we would like to explore in this chapter's case study. If you don't factor in time, then money spent trying to react to customer behavior is wasted in the fast-moving times in which we live.

7.2 Scenario: Setting Prices in Gas Station Kiosks

Leech Oil Pty Ltd (LO) runs a network of about 1,200 gas stations on the Australian continent. Due to the country's sheer size and the distances between cities, most gas stations also include a small supermarket (called Leech Essentials, LE) and a shop selling alcoholic drinks (Leech Lifeblood, LL). Both chains, LE and LL, are run by a subsidiary of LO; the question of how closely all three shops are integrated is mainly determined by state legislation.

Electronic price tags

To save personnel costs of price marking, LO has put electronic price tags on the shelves of all their shops in all three parts of the group. The project went live a couple of months ago, some teething problems have been overcome, and everything is working smoothly now.

Chance findings of the project

During go-live, Leech Oil gained some fascinating insights. Due to an error in an SQL statement, some articles were being marked with prices that were consistently too high. When trying to evaluate the resulting damage in terms of lower demand and revenues—LO was planning to assert a claim against its IT service partners—Bud Wiser, the controller with LL, noticed that customers reacted to overpriced articles differently at different times of the day. In the morning, consumers tended to be clear headed and more price sensitive than in the evening. Furthermore, there were some time slots during which behavior was strikingly different; indeed, on some days, from a certain time on, drinks were even bought if they were marked with a price that was more than 100 percent too high.

Interviews with employees explain anomalies

Intensive interviews with employees in affected stores delivered a couple of interesting explanations:

- On one of the days in question, the Australian national rugby team (the Wallabies) had given their arch-rivals, the All Blacks from New Zealand, a good thrashing. After the game, fans invaded LL's stores to replenish their beer stocks and celebrate the victory. This only happened because the triumph came unexpectedly, so most of them a) didn't have enough booze on hand and b) were even more excited than usual.
- On another day, heavy rainfalls forecasted for the evening had failed to materialize. Temperatures had remained over 35 degrees Celsius (95 degrees Fahrenheit) even after sunset. This led to a lot of spontaneous barbecue parties being organized for which people needed wine and beer. At the same time, sales of barbecue accessories and meat had gone up for Leech Essentials. A more thorough analysis showed that, again, a number of factors had come together. Temperatures were unusually high for that time of the year, it got darker a bit earlier than in midsummer, and the respective day was a Friday, so most customers didn't have to get up for work the next morning.

Short-term behavior almost unpredictable

Although there were good reasons for customer behavior, and of course it could be explained quite easily in hindsight, this behavior would be very hard to consistently predict and use for forecasting in the long term. After major sports events, customers sometimes buy drinks to celebrate, but sometimes they might need it as a source of comfort if their favorite team went down the drain.

7.3 Static Customer Segmentation: Costs, Risks, and Opportunities

Taking time into account

Because LO doesn't produce or refine oil itself but is only a vendor of petrochemical products, their oil, grocery, and liquor businesses are structured in very similar ways, and market research has always played a central role in each of these business areas.

Leech is proud of the hardware and software that it uses and of the high level of expertise of its employees in various departments; many of these employees hold doctorate degrees in mathematics or psychology. As a consequence, Leech is also aware that customer behavior depends a lot on the time of day. Leech's experts have therefore split the week into *7 * 24 = 168* time slices, leading to up to *365 * 24 = 8,760* time slices per year (with a few extra in leap years). These time slices are used when segmenting customers or deseasonalizing time series.

7.3.1 Problem: Walking on Thin Ice/Data

Low data volumes, long periods under review

When segmenting customers at this level of detail—using data from just one time slice—LO kept hitting the same problem: the company itself has only been around for about 30 years, and its supermarkets and liquor stores were only set up in the last decade. At best, therefore, Leech has data for some 30 years, which is pretty meager for some statistical algorithms. Furthermore, the company's mathematicians don't get tired of emphasizing that applying stationarity for such a long period of time is typical IT pie-in-the-sky, muddle-headed, simple-minded, I-know-best rubbish.

Market research is skeptical

Accordingly, the head of manufacturing was skeptical about the controller's observations. His insights were certainly right and interesting, he admitted patronizingly, but in the end, these were just one-off events without any real informative or predictive value. On every day analyzed, there had been a unique cosmic gathering of seemingly random facts that had dramatically changed customers' price sensitivity; one could almost believe that their buying decisions were being driven by the stars! Even if Leech could successfully capture such events by continuously monitoring weather forecasts or intermediate results of rugby games, the impact of these events on customer behavior would still remain somewhat inscrutable. Below the line, by working at a higher resolution or granularity, Leech would simply exacerbate the problems it had already in terms of its meager stores of data.

7.3.2 Numerical Example

Unexplored potentials for revenue

The concerns from market research should not be dismissed too lightly. Still, the idea that one could easily double an article's price and get away with it scot-free has gripped the heart of the controller like his first love in high school did. Although his insights were chance finds, he cannot resist the temptation to have a closer look at this topic. Some rough estimates have led to the following:

- Discounters are selling the Castlemaine's XXXX lager jumbo carton with 24 cans at about 60 AUD, resulting in a very low margin. LL is selling it at 75 AUD, generating a margin of 20 AUD per carton.
- If one could increase the sales price to 95 AUD and get away with it, the margin would increase to 40 AUD per carton. As long as such a price increase would not lead to a sudden drop of more than 50 percent in demand, the higher price would still have a positive effect on LL's total margin. In other words, whether increasing the price does or doesn't make sense will depend on customers' price sensitivity, which can be measured by the gradient of the demand curve; mathematically speaking, you are looking for the first derivative of the demand curve.
- On the day of the rugby game, one LL store near Kiwirrkura missed the price increase due to a data-transfer error. In this store, sales of the Castlemaine XXXX jumbo pack jumped from 50 to 110, a 120 percent increase, which is even more impressive considering that the total population there is only some 250 people. This resulted in margins climbing from 1,000 to 2,200 AUD, which is an increase of 120 percent.

  In other stores, the price was doubled, bringing up the per-carton margin from 20 to 95 AUD; sales had, however, still increased by about 100 percent. With a store that normally sold x cartons, this brought up the total margin by $((2 * x * 95) - (x * 20)) / (x * 20) = 170 * x / (20 * x) = 8.5 = 850$ *percent*.
- Even better is the fact that margins generated by price increases come at lower costs than margins generated by volume increases due to savings in terms of storage capacity and shipping costs.

Opportunities with other article groups

Similar potential could be slumbering in other article groups for Leech's gas stations (LO) and supermarkets (LE). With gas, the customers' willingness to accept higher prices will also depend on the time of day at which they fill up.

Dealing with the skeptics

Nevertheless, three major doubts were still gnawing away at Bud's enthusiasm:

- The guy from market research isn't an idiot; his reservations are justified. From a statistics course at university, Bud remembered that there is a dependency between the size of a sample and the reliability of conclusions drawn from it.
- Every combination of circumstances he observed was indeed unique. Even he himself could hardly imagine how anybody could have anticipated the relationships between these events and the customers' price sensitivity.
- Spontaneous price increases of 100 percent or more could just as easily kill sales stone dead. Marking up prices just a little might go unnoticed by consumers, but once they get the impression that you are trying to rip them off, they might vote with their feet and avoid your store for quite some time. Or maybe accidently let it catch fire during a storm and watch it burn.

7.3.3 Conclusion: Cause–Effect Relationships are Irrelevant

Forget about the why

To resolve Bud's dilemma, we need to revert to an insight from Chapter 1, Section 1.4.2. With big data solutions, the why is often irrelevant. You want to detect and utilize dependencies, and you want to notice in time if they are no longer applicable; understanding *why* they occur is not as high up your priority list, because doing so might take too long.

Insights valuable but too late

This doesn't mean that you should not try to gain insights into cause–effect relationships. If Bud could prove that the weather does indeed have a systematic impact on beer sales, he could factor this relationship into sales planning and even production planning (as discussed in Chapter 4) or at least use it to monitor sales. However, if a forecasted thunderstorm fails to materialize, then you would only know this at the time it was meant to hit, so we are not talking about a dependency between a forecasted event and sales volumes, but about a prediction that could turn out to be wrong at the time during which events were meant to happen. In such a case, that insight arrives far too late to be taken into account for production or sales planning.

[+]

Reaction Time Determines Focus

With electronic price tags or online sales, one can adapt prices within a fraction of a second. When advertising online with Google, you can exchange an image or a design instantaneously, picking the one that works best with your target group. In neither case do you have the time to analyze why.

Hence, whether or not you are interested in understanding cause–effect relationships depends a lot on how fast you need to or are able to react!

Rare events

To make things worse, in our example many of the relevant dependencies seem to be one-off phenomena. This is one reason that mathematicians were saying that Leech's database was meager: 30 years of experience does indeed correspond to more than 250,000 time slices, but coincidences such as the ones Bud is thinking about may still only occur once or twice during these 30 years. Even if you understood customer behavior, if such a combination of events doesn't occur again this century, then what's the benefit of having analyzed it?

7.4 Solution: Dynamic-Empirical Algorithms

In a nutshell, LO's controller would like to react to changes in customer behavior that in practical terms are unforeseeable, changes that he is not able to understand in the short run, and ones that may occur only once. Therefore, "react" in this context can only mean doing things that can be done in real time.

Options for action in real-time processes

One of the few parameters LO can change more or less instantaneously is the prices of their products, whereas businesses that are selling online have a lot more controls at their disposal. For example, a content management system can modify a website's design or content from one second to the next, and messaging applications like the ones discussed in Chapter 9 can send out SMS messages or tweets to customers whenever they want.

Sending messages to customers that have a customer loyalty card with Leech and whose mobile phone numbers the company may therefore know could, however, also be an option for LO. Furthermore, Leech could use credit card or customer loyalty card numbers to identify customers at checkout, offering personalized promotions or discounts and displaying these on the cashier's screen during payment.

Three unknowns with LO

In the end, LO faces the following uncertainties:

- They don't know what events might trigger changes in customer behavior.
- They don't know when such events are going to occur next.
- They don't know exactly how customer behavior will change.

Even if Leech knew the events it was looking for and the exact times of these events (for example, that the forecasted thunderstorm at 6 p.m. on February 3, 2015, is not going to take place), it still wouldn't know how customers would behave—that is, for example, how price sensitive they are going to be or from which threshold onwards they would take umbrage and not come back ever again. In the end, this boils down to modeling customer behavior.

Trial and error

The three unknowns listed previously can only be dealt with experimentally, much like Leech did inadvertently when introducing the new electronic price tags. Thanks to big data, it's no longer a problem to evaluate the outcome of such experiments in real time.

7.4.1 Related Value Maps in SAP Solution Explorer

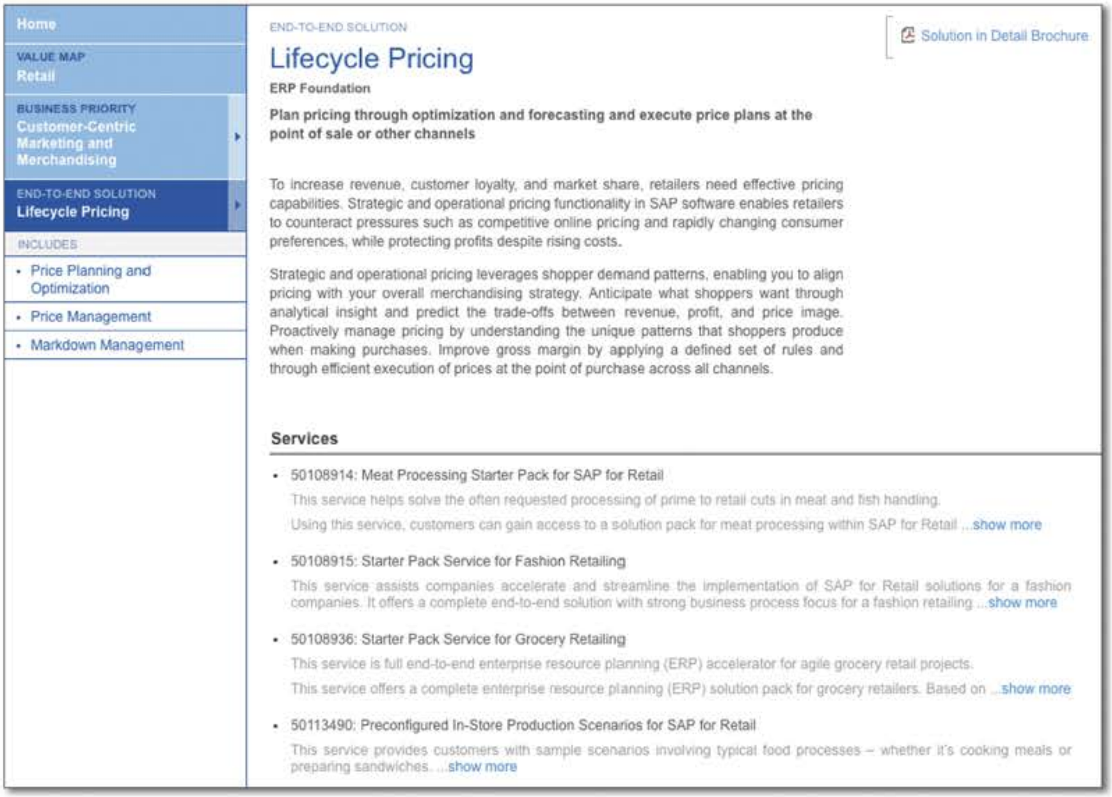
Many appropriate SAP solutions

Because there is a lot to be gained by using big data in the context of sales and distribution, you will find a couple of related solutions in the respective value maps within SAP Solution Explorer. Take a look at the value maps for retail (for example, Promotions and the Shopper Insight RDS or Customer Interaction and Personalization and SAP Real-Time Offer Management [RTOM]); there are also things to be found in the cross-industry overviews related to sales (for example, Sales Performance Management and the SAP HANA CRM Analytics RDS).

Example: Lifecycle Pricing

Because we are only looking at pricing in this scenario and because we only need relatively simple analyses of operational data, we will use the value map for retail as an example; from that value map, we have selected the Lifecycle Pricing end-to-end solution (see Figure 7.3). For our purposes, the following solutions could be useful:

- **Price Planning and Optimization**
  Price Planning and Optimization is used to automatically determine prices at the level of stores and stock-keeping units (SKUs).
- **Price Management**
  Price Management is also used for pricing; unlike Price Planning and Optimization, its main purpose is to help businesses adapt prices, reacting to temporary or local changes in customer behavior.
- **Markdown Management**
  Markdown Management completes Price Planning and Optimization and Price Management with a solution that helps you manage discounts—for example, during seasonal sales. One key objective here is to optimize the timing of markdowns to increase sales quantities without sacrificing too much margin.



END-TO-END SOLUTION

Lifecycle Pricing

ERP Foundation

Plan pricing through optimization and forecasting and execute price plans at the point of sale or other channels

To increase revenue, customer loyalty, and market share, retailers need effective pricing capabilities. Strategic and operational pricing functionality in SAP software enables retailers to counteract pressures such as competitive online pricing and rapidly changing consumer preferences, while protecting profits despite rising costs.

Strategic and operational pricing leverages shopper demand patterns, enabling you to align pricing with your overall merchandising strategy. Anticipate what shoppers want through analytical insight and predict the trade-offs between revenue, profit, and price image. Proactively manage pricing by understanding the unique patterns that shoppers produce when making purchases. Improve gross margin by applying a defined set of rules and through efficient execution of prices at the point of purchase across all channels.

Services

- 50108914: Meat Processing Starter Pack for SAP for Retail
  This service helps solve the often requested processing of prime to retail cuts in meat and fish handling.
  Using this service, customers can gain access to a solution pack for meat processing within SAP for Retail ...show more
- 50108915: Starter Pack Service for Fashion Retailing
  This service assists companies accelerate and streamline the implementation of SAP for Retail solutions for a fashion companies. It offers a complete end-to-end solution with strong business process focus for a fashion retailing ...show more
- 50108936: Starter Pack Service for Grocery Retailing
  This service is full end-to-end enterprise resource planning (ERP) accelerator for agile grocery retail projects.
  This service offers a complete enterprise resource planning (ERP) solution pack for grocery retailers. Based on ...show more
- 50113490: Preconfigured In-Store Production Scenarios for SAP for Retail
  This service provides customers with sample scenarios involving typical food processes – whether it's cooking meals or preparing sandwiches. ...show more

Figure 7.3 Lifecycle Pricing End-to-End Solution

Other SAP and non-SAP products for POS/pricing

All three solutions mentioned previously are used for pricing. Hence, the application we are thinking about on behalf of LO needs to supply one or more of them with data or rules needed to determine prices. In addition, Leech may want to deploy the SAP HANA Operational Reporting RDS, POS systems, such as SAP Point-of-Sale, and software products that can send data to electronic price tags. The latter are simply needed to ensure that whatever prices Price Planning and Optimization, Price Management, and Markdown Management generate will actually end up on the front of the shelves.

7.4.2 Functional Requirements

Focus: prices and real-time promotions

As mentioned in Section 7.3.3, Leech Oil is considering a solution that can react very quickly. The group could, for example, aspire to refine or review decisions such as the following in the future:

- Should prices in a certain store rise or be cut? By how much? Which products?
- Should LO, LE, or LL send special ad hoc offers to individual customers based upon the customer's location (if available), the date/time, or both? To which customers at which locations and dates/times should these offers be sent, when should the offers be sent, and what should the content of the offers be?
- Should customers be treated to personalized special offers or discounts during checkout—for example, via a message displayed on the cashier's screen—once they have been identified?

Enhancing, rather than replacing, models

Why did we say "refine" or "review"? According to the introduction to this chapter, Leech is fairly good in terms of market research. We can therefore assume that they are already using a couple of pretty sophisticated yet static models, which are in principle able to make pricing decisions. This is just to emphasize that we are not developing (see Chapter 5), monitoring, or enhancing (see Chapter 4) models here. Instead, we are trying to find out whether deviating from a model's recommendations in individual cases or against the background of special events might make sense. Building an SAP HANA-based app that can support you in that respect is the more general task we want to help you resolve.

In terms of the required reaction speed, what we wish to achieve resembles the outcome of the use case described at *www.saphana.com/docs/DOC-1834*. There also is a related video on YouTube (see *www.youtube.com/watch?v=McDaSXCs5ZQ*).

Example: Bigpoint

Back to the use case! Bigpoint is a German-based developer of browser-based online games. Their business model is as follows: games are made freely available with no charge to play; when playing, users are presented with context-specific offers they then can choose to pay for. If a seven-headed dragon is the only thing standing between you and the tower holding a beautiful maiden, you might be tempted to invest 1 EUR in the heat of the moment to buy a magic sword.

SAP HANA, the dragon, and the maiden

This is exactly the point at which SAP HANA comes into the equation. As games are evolving dynamically based upon the interaction between humans and machines, you are unable to predict the scenario that a gamer is going to face at a certain point in time. You may, however, be able to guess which objects he might choose in a certain situation. If you haven't got the data you need to make an intelligent guess, you could just give it a try and learn from your mistake.

Online games are a controlled environment

One major difference between the use case with Bigpoint and the scenario defined in Section 7.2 is that with LO, we are dealing with a business that operates points of sale (stores) in the real world; LO therefore has to handle not only far more data but also much more heterogeneous data that are not too easy to get ahold of.

Think about a customer's location: Bigpoint could use the player's IP address for this or—if people are playing on mobile devices—gather geodata via the game app. LO would have to invest more gray matter into figuring out how to gather your location information and could also factor in a lot of other things that might not be so important for Bigpoint. One example of this is that LL might have data on and evaluate its customers' past drinking preferences and habits. Bigpoint would probably neither have these data nor be that interested in them (after all, the company is too ethical to tempt gamers into buying things when they are not sober).

Parallels with medical studies

By now, some of you may have noticed an interesting analogy. Our task has a lot in common with medical studies—that is, with checking the effectiveness of a drug or a therapy. As with medical studies, Leech has four requirements at which we are will take a closer look:

- Drawing samples
- Conducting experiments
- Making and executing decisions
- Analyzing experiments in detail

Draw Samples

Select stores, articles, and customers

As with other case studies, we'll need two samples. For one sample, we will change one factor, such as the price of one single article, and for the other sample, we will leave things unchanged, leaving the price as determined by any of the applications mentioned in Section 7.4.1.

When gathering these samples, we need to consider three things:

- The sample size needed.
- Which combinations of store/article/customer and so on to include in our sample (there are different selection methods in sampling).
- We need to assume that with both samples all other things have remained unchanged during our experiment (this slightly adventurous assumption is called *ceteris paribus*).

Conduct Experiments

Do things differently one time

Once samples have been generated, LO will start experimenting. From a technical perspective, this means that prices have to be changed within the systems involved, messages have to be sent to customers, and so on. Some of this takes place outside of the application we are designing here or—as with sending new prices to electronic price tags—outside the SAP world. In this chapter, however, we are not going to look at interfaces; we simply assume that SAP HANA will receive whatever data it needs via appropriate products for data management in real time. This includes data about the samples, the design, and the outcome of experiments.

Make and Execute Decisions

Evaluate tests

Via a quick ad hoc assessment of its experiments, Leech needs to answer two basic questions:

- Has the change—for example, an increase or decrease in price—led to a different outcome, or could the differences seen between the two samples also be explained as random fluctuations?
- If there is a different outcome, is the new result advantageous (or not) from LO's perspective?

No need to try suicide

If, for example, an increase in demand triggered by lower prices does not compensate the company for the per-unit margin lost, then this measure might not make sense. We assume that Leech's experiments are designed in a way that keeps SAP HANA from doing stupid things. By "stupid things," we mean actions that will by definition lead to commercial damage or to LO committing financial suicide. For example, you probably don't want to test reducing the price of an article below its cost of acquisition at all stores, even if that might tease out some extra demand. Offering customers the odd bargain makes sense in certain cases—for example, to get them into the right kind of shopping mood—but selling your goods below their cost of acquisition would actually be illegal in a number of countries. Although this particular detrimental measure can be excluded upfront, other things may turn out to be harmful after the experiment has been conducted.

Telling operational systems what to do next

Once the results have been evaluated, it's time for a decision. If the differences between both samples were both significant and advantageous, you would probably want to extend the change you have made to other, or all, stores or customers. If they weren't, you may want to reset things to normal.

In both cases, your solution—implemented using SAP HANA, for example—is supposed to tell affected operational SAP and non-SAP systems what to do next, communicating with them via appropriate real-time interfaces. One example of such an interface is SAP HANA XS (see "Other Products Databases, Platforms, Technology, and Services" in Chapter 2, Section 2.1.3). This could happen in push mode (sending messages as and when required) or pull mode (the external app queries your SAP HANA database).

Naturally, if you are of a more cautious temperament you could also go for a step-by-step approach, extending your decision to a larger audience (more stores, articles, or customers) in a number of phases, each time comparing the subset in which the change has been made against the rest of your population.

Analyze Experiments in Detail

Worry about models later

After completing a successful experiment, LO may want to do some further analysis and try to understand what really happened. As with drugs, many humans are not satisfied with just knowing that something works; many of us will not rest before we also think we know *why* it works, regardless of the fact that our understanding is often nothing but a temporary illusion of knowledge. This new insight would also help Leech to question, refine, improve, or rebuild existing models. However, we will not address such desires in this chapter.

If you are interested in verifying existing models, then take a look at Chapter 4. If you haven't got any models yet and would like to develop them, then turn to Chapter 5. If you need to build a structure that is very flexible in terms of implementing ever-changing models, then Chapter 6 is the place to go.

7.4.3 Building Blocks of the Solution

(Statistical) algorithms needed

To keep the length of this chapter under control, we will only look at two of the four requirements listed in Section 7.4.2: drawing samples and making and executing decisions.

Sampling

Why use samples?

Samples are used to make statements about a much larger population, such as all stores, all articles, or all customers in Australia. In most cases, experimenting with the whole population would be costly, dangerous, and irresponsible, which is why we are looking for special cases that deviate from what our models tell us when using two or more samples. To begin with, this raises the question of how many and which objects our samples should contain.

Sample Size

Acceptable error probabilities

The sample size depends on what kind of probabilities for type I and type II errors Leech is willing to accept.

[»]

No way around deduction here

Type I and Type II Errors

Let's assume LL changed the price of Castlemaine's XXXX family pack in 100 stores, leaving it unchanged in another sample of 100 liquor stores. We now want to test the hypothesis that the price change does *not* have any effect on demand; mathematically speaking, our null hypothesis is "expected values within the populations that the samples were drawn from are identical." When accepting or refusing that null hypothesis, four things can happen:

- We reject the null hypothesis although it is true. This is called a *false positive* or a *type I error*, often denoted by the Greek letter α. In our case, this would mean that our analysis leads to the conclusion that changing the price does make a difference, but in reality the differences we saw were due to random variations.
- We reject the null hypothesis and rightly do so because it is wrong. This is a correct outcome or a *true positive*.
- We do not reject the null hypothesis and are correct to do so because it is true. This is another correct outcome or a *true negative*.
- We fail to reject the null hypothesis although it is false. This is called a *false negative* or a *type II error*, denoted by the Greek letter β. In our example, we would judge that the price change didn't lead to significant changes in demand although it actually did. We simply picked the wrong samples.

In Chapter 5, Section 5.4, we emphasized that you should work inductively rather than deductively. Strictly speaking, however, statistical tests are always deductive. The purpose of a statistical test is to check a null hypothesis, and in many cases this hypothesis has been derived deductively. Nevertheless, we still use statistical tests in this chapter for four reasons:

- We cannot work inductively, because we don't have enough data to do so.
- The tests used here are abstract and generic. They check whether two parameters are statistically dependent without saying much about the type of that dependency. Or more specifically we are checking whether two parameters, such as customer demand in a test group and customer demand in a control group, behave in the same way or not without making any other statement regarding customer behavior.

- In Chapter 5, we handled developing models without making any initial restrictive assumptions. In Chapter 6, we once again picked up this concept, this time extending it to complete decision trees/data flows instead of only looking at two parameters. In this chapter, we aren't questioning or redesigning data flows like the ones developed in Chapter 6, adapting them to a new, changed environment. Instead, we are assuming that the environment as such has not changed; we are hunting for opportunities created by exceptions. Any of the experiments we conduct in this context could be considered deductive.
- The faster our system is, the greater will be the variety of experiments we can conduct. We will still work deductively and also try to be as open minded as possible.

Once LO has defined the probabilities of error it can live with and its market research department has delivered some additional data—for example, the estimated standard deviations for the relevant parameters—the sample size can be calculated.

By the way, the probability of type II errors is also used in *power analysis*, a statistical method that measures how well a statistical test performs. The *power of a statistical test*, sometimes also called its *sensitivity*, is defined as *power = 1 – probability of type II errors*.

Select Sample Data

Stratified sampling

Intuitively, most of us would probably tend to let chance take its course; we would throw a die or generate random numbers to select the objects (stores, articles, or customers) that we want to include in our sample. However, chance might play tricks on us, and we could end up with a sample that contains a much higher percentage of women than our population, for example. In such a case, *stratified sampling* might be a better strategy; we first find out what the percentage of women and men in our population is and then go for the same distribution in our sample, selecting that predefined percentage of women and men there.

Use existing knowledge

Maybe we also have other kinds of information about our customers, such as knowing some determining factors that help identify those that are particularly susceptible to special offers at checkout. If we do, it would also make sense to use this knowledge when sampling, and we would exclude customers that we know we couldn't influence by pricing.

Many sampling algorithms

Our thoughts about the value of samples are nearly endless. Costs of sampling, for example, are sometimes important, and there are far more sampling algorithms than you may think.

Make and Execute Decisions

Two-sample and multisample tests

When evaluating whether a drug is more effective than a placebo, researchers often conduct *two-sample tests*, a special case of *multisample* or *multiple-sample tests*. A two-sample test checks the hypothesis that a certain parameter in the two populations that the samples came from don't differ; a multisample test does the same for more than two samples.

In medical research, experts often use three kinds of tests that could also work well in our example:

- **t-test**
  Student's t-test is the simplest two-sample test; it only checks whether the means of two samples are the same. A disadvantage of the t-test is that it is a *parametric* test, meaning that it is based upon the mathematical assumption that the population both samples were drawn from is normally distributed; in Chapter 4, Section 4.4.3, we explained why this is a limiting assumption. If you aren't sure about this, you may want to work with the next two tests.
- **Mann-Whitney-Wilcoxon test**
  The Mann-Whitney-Wilcoxon (MWW) test, also known as Mann–Whitney U test, Wilcoxon Rank-Sum test, or Wilcoxon–Mann–Whitney test, checks whether two samples can be assumed to have come from the same population. It is more flexible than the t-test; for example, you don't have to assume that data are normally distributed.
- **Kruskal-Wallis test**
  The Kruskal-Wallis test—also called the H-test or Kruskal–Wallis One-Way Analysis of Variance—is an extension of the Mann-Whitney-Wilcoxon test used when more than two samples need to be compared.

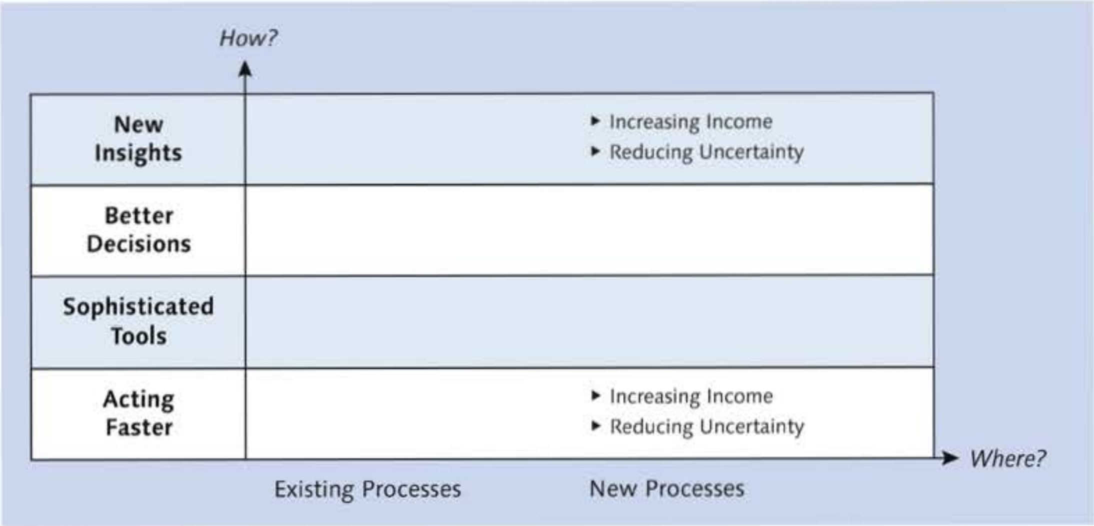
7.4.4 Potential Benefits and Value Drivers

Trade margins and sales volumes

With the requirements defined in Section 7.4.2, Leech Oil is pursuing a very simple objective; they want to use irregularities in customer behavior, which their models cannot predict, as an opportunity to realize higher absolute trade margins by setting higher prices, generating higher sales volumes, or both. At the same time, they also want to limit the risk of losses caused by such irregularities.

Reducing risks

Within this case study, we are only focusing on these irregularities; we are therefore ignoring potential benefits that stem from, for example, questioning models that are no longer suitable. When implementing the solution, we are talking about a new business process that creates value because LO can act faster and make better decisions—that is, using insights that are not limited by the scope and predictive power of its models alone. With static, rule-based systems, LO would never be able to exploit such exceptions in real time. Figure 7.4 shows the benefit–value driver matrix for our example.



	Existing Processes	New Processes
New Insights		▸ Increasing Income ▸ Reducing Uncertainty
Better Decisions		
Sophisticated Tools		
Acting Faster		▸ Increasing Income ▸ Reducing Uncertainty

Figure 7.4 Customer Behavior Benefit–Value Driver Matrix

We are not going to explain the two value drivers within the matrix in detail; they are covered extensively in Chapter 1, Section 1.4.3 when discussing generic value drivers.

7.5 Implementation Scenario and Architecture with SAP HANA

Interfaces with SAP systems

Our application will be surrounded by quite a few peripheral solutions, some of them based upon SAP products and some of them supplied by other vendors. There will be numerous interfaces between our application and the systems around it:

- SAP HANA will need data about the populations (stores, articles, customers, etc.).
- On the basis of these data, SAP HANA will have to size and draw samples.
- When conducting experiments, SAP HANA will have to tell the surrounding solutions which parameters to change—for example, which prices to increase.
- In the aftermath of these experiments, SAP HANA will need the results.
- Our application will then tell other solutions whether the respective change should be applied to a larger number of stores, articles, or customers, whether the maneuver should be aborted, or whether changes need to be made in customizing.

When looking at solution and data architectures, we will ignore all interfaces to non-SAP systems. SAP customers could use standard platforms like SAP Gateway to communicate with non-SAP products.

7.5.1 Implementation Scenario and Framework Architecture

Scenario depends on data sources

To implement what was described in Section 7.4.2, either of the following two implementation scenarios could be appropriate:

- **Accelerator scenario**
  If integrating LO's application with SAP CRM or SAP ERP is not a significant factor or if non-SAP solutions are already in use, we would recommend choosing the accelerator scenario. In this case, only a few selected data from other solutions are replicated from the SAP CRM or SAP ERP solutions and analyzed within SAP HANA.

These analyses then lead to insights about customers' microbehavior that in return result in instructions sent from SAP HANA to products for data generation. Once these instructions have been executed, new data are produced and handed over to SAP HANA, initializing another loop.

SAP has assigned their own RDS for customer segmentation via SAP HANA to this scenario. Customer segmentation is one building block of a solution called SAP CRM Audience Discovery and Targeting.

- **SAP Business Suite on SAP HANA scenario (with extensions)**
  If you intensively use SAP CRM and SAP ERP, a substantial amount of the data we are talking about here will come from these packages. Furthermore, if you have deployed standard solutions like SAP Gateway or SAP Process Orchestration and manage other processes like mobile communications via SAP Mobile Platform with SAP, then a higher level of integration makes sense. In this case, you would also revert to SAP's standard interfaces for communication between products for data generation and SAP HANA.

  You should, however, keep two things in mind:

  - The SAP Business Suite on SAP HANA scenario often comes with SAP BW on SAP HANA. For our case study, it doesn't really matter whether LO uses SAP BW or not. The only thing that is important is making data from products for data generation like SAP CRM or SAP ERP available within an SAP HANA database.
  - Neither SAP Business Suite nor SAP BW nor SAP HANA contain all building blocks listed in Section 7.4.3 as standard. We will therefore definitely have to develop additional applications.

Focus: accelerator scenario

In the end, Leech may therefore choose a combination of both the accelerator scenario and the SAP Business Suite on SAP HANA scenario, using the latter with or without SAP BW as a product for data exploitation. The following considerations, however, are based on the accelerator scenario (see Figure 7.5).

New SAP HANA apps scenario for non-SAP customers

Customers using neither SAP CRM nor SAP ERP may also consider the new SAP HANA apps scenario, as presented in Chapter 2, Figure 2.18. A key difference between the accelerator and new SAP HANA apps scenarios lies in the fact that the latter does not care much about what is happening outside of SAP HANA. Again, as in the accelerator scenario, the data for Leech's app would come from third-party software solutions; such third-party software solutions would also have to be provided with data from SAP HANA. To integrate all the products involved, even non-SAP customers could still resort to SAP platforms.

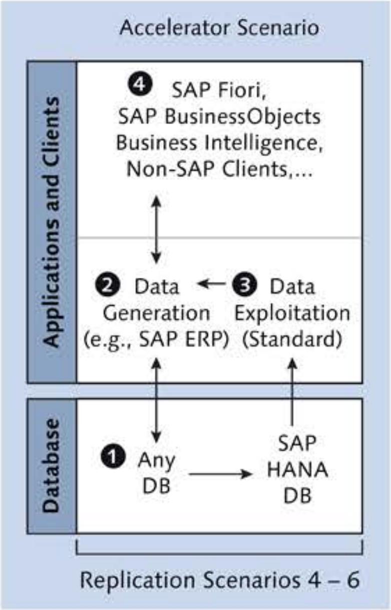

Accelerator Scenario
Applications and Clients
❹ SAP Fiori, SAP BusinessObjects Business Intelligence, Non-SAP Clients,...
❷ Data Generation (e.g., SAP ERP)
❸ Data Exploitation (Standard)
Database
❶ Any DB
SAP HANA DB
Replication Scenarios 4 – 6

Figure 7.5 Customer Behavior Implementation Scenario

In the end, the choice between the accelerator scenario and the new SAP HANA apps scenario would depend on how closely your new application would have to be, or could be, integrated with your existing products for data generation.

Back to the accelerator scenario: as before, let's take a closer look at the individual layers shown in Figure 7.5.

Databases ❶

No processing on database layer

As before, we may not only have to harness classic databases as sources of data but also APIs or streams. Because LO has very high expectations in terms of processing and reaction speed, SAP HANA should not be required to waste too much time on preparing or enriching the data—for example, on geocoding store locations. Whenever possible, the delivering applications should hand over clean, enriched, and filtered data to SAP HANA. Two possible data logistics solutions (products for data management) in this context are SAP Data Services and SAP Replication Server. In Section 7.5.2, we will briefly look at another option specifically for POS data.

Products for Data Generation ❷

No external data modification

All products that Leech's new application receives data from or sends data to are to be found in this segment. Apart from SAP or non-SAP CRM or ERP solutions, this also includes the aforementioned POS systems or smartphone apps that collect data for LO. SAP HANA may read whatever it needs directly from the respective solution's database but should not update such external data itself. As a general rule that is valid even outside the big data space, data should only be updated by the application that owns them and knows how to handle them properly.

SAP HANA may kick off modifications

However, this does not mean that SAP HANA cannot trigger the required modifications (using standard methods, such as a `CALL TRANSACTION` statement within an ABAP program that could be created just for this purpose).

[+]

Relocation of Decision Tables to SAP HANA

Another interesting alternative for SAP HANA to interact with other applications is the relocation of decision tables into SAP HANA. Further information about this can be found in *ABAP Development for SAP HANA* (Gahm, Schneider, and Westenberger, SAP PRESS, 2014). The SAP Community Network also contains information about this subject.

In the architecture model presented in Section 7.5.2, we assume that products for data generation are going to use OpenSQL (a proprietary variant of SQL developed by SAP) to access data in SAP HANA decision tables. If so, then SAP HANA does not have to send anything to products for data generation; these products will help themselves to the data they require.

Products for Data Exploitation ❸

Samples, experiments, and decisions

Within Figure 7.5, this is the segment in which Leech will have to implement all the wonderful functionalities mentioned in Section 7.4.2. We also know that they will need a Swiss army knife, or some other clever box of tricks, that can size and draw samples, trigger experiments, evaluate their results, make decisions, and tell other applications what to do next.

As mentioned previously, the most elegant variant when it comes to communicating to the outside world is not to communicate at all but instead to provide decision tables that other applications can use. For LO's other requirements, their Swiss army knife will be comprised of the following tools:

- **Sizing samples via R**
  SAP Predictive Analysis doesn't have functionalities for sample sizing. R, however, has quite a lot to offer in that respect. Two examples are the `pwr` package developed by Stéphane Champely, which comes with nine different sizing functions, and the package `samplesize` originally developed for medical research and also of interest for the tests mentioned in the final item in this list.
- **Selecting sample data via SAP PAL**
  Within a category called "Preprocessing Algorithms," there are a couple of functions in SAP PAL that can be used to select sample data or to preprocess data that have been picked during sampling. This includes functions such as `SAMPLING` (a function used to draw a sample via eight different sampling methods, including stratified sampling), `SCALINGRANGE` (to normalize data when it comes to, for example, comparing sales volumes of smaller and bigger stores), or `SUBSTITUTE_MISSING_VALUES` (to supplement data that are incomplete due to transmission errors, for example).
- **Selecting sample data via R**
  In R, there is an extremely powerful package called `sampling` that embraces over 50 individual functions. For purists who want to use R only, the SAP PAL functions mentioned previously are also included. Not only simple things like normalization (via function `scale`) are available but also an almost inexhaustible treasure trove of ideas to help you deal with missing data (search for words such as "impute" or "imputation").

- **Making decisions via SAP PAL and R**
  SAP PAL gives you some important statistical tests (`VARIANCETEST`, `VAREQUALTEST`, and so on) but doesn't cover a lot of ground in that respect. For more sophisticated tests—like those mentioned under "Make and Execute Decisions" in Section 7.4.3—you should turn to R. In R, you'll be able to use the t-test (`t.test`), the Mann-Whitney-Wilcoxon test (`wilcox.test`), and the Kruskal-Wallis test (`kruskal.test`) out of the box, plus many others you can download for free via supplementary packages.

Different Scopes of SAP PAL and R

[+]

The functional requirements discussed in this paragraph clearly highlight the differences between SAP PAL and R. SAP PAL is easier to handle, but has a limited functional scope. In contrast, there is hardly a thing you wouldn't find an R function or an R package for.

On the other hand, the expertise needed to select the right tools and the statistical knowledge needed to handle them are clearly higher with R. Nevertheless, you should assume that you'll have to deal with this language. There is no reason for SAP to reinvent something in SAP PAL that is already available for free in R, and SAP is working on some measures that will further improve the performance of R code embedded into procedures.

Hence, you should probably assume that many of the products for data exploitation you are going to buy or build will consist of procedures containing R code and feed data into calculation views.

Clients ❹

Clients with customers

Any clients used by Leech's customers, such as apps on smartphones, fall into the category of surrounding and peripheral systems from our perspective, so we are not considering them to be clients that belong to the solution we are designing here. The only remaining question is whether the product for data exploitation that Leech will have to build needs some special kind of client itself.

Clients for Leech

Fundamentally, we are trying to design a system that is going to act more or less autonomously. This means that we are trying to get rid of clients, so the client layer is less important here, regardless of which scenario (accelerator, SAP Business Suite on SAP HANA, or new SAP HANA apps) is chosen. The only thing you may need is a control desk or switchboard for monitoring and managing the whole system. A couple of dashboards and some kind of alerting/messaging application should do the job. You can build these dashboards using SAP BusinessObjects Dashboards or SAP BusinessObjects Design Studio.

7.5.2 Data Architecture

Technical requirements

Let's refocus and at the same time translate the functional requirements within Section 7.4.2 into SAPanese:

- We want a solution acquiring store-, article- and customer-related attributes and facts, plus facts about quantities sold in Leech's stores, in real time. Some of the required data (like store, article, or customer master data) may as well be delivered via batch processes.
- Using SAP PAL and/or R, the solution is to determine appropriate sample sizes and to select sample data.
- Based upon these samples, Leech would like to run experiments that are related to pricing. For this, the solution is expected to trigger price changes, such as general increases, reductions, or rebates, via the solutions that are used for pricing—that is, you may want to change customizing or condition tables there or make the solutions use SAP HANA decision tables via customer enhancements. These changes should be limited to objects included in the samples.
- Using statistical tests in R, the solution should determine whether the experiments did indeed lead to significant differences in customer behavior.
- Depending on that outcome, the changes triggered in pricing solutions are either to be reversed or to be extended to other objects. The latter could mean that samples are extended step-by-step or that we go for the whole population straight away.
- This cycle will go on and on, trying different things on different objects every time.

Figure 7.6 shows a possible data model for this solution. We will take a closer look at its components in the following sections.

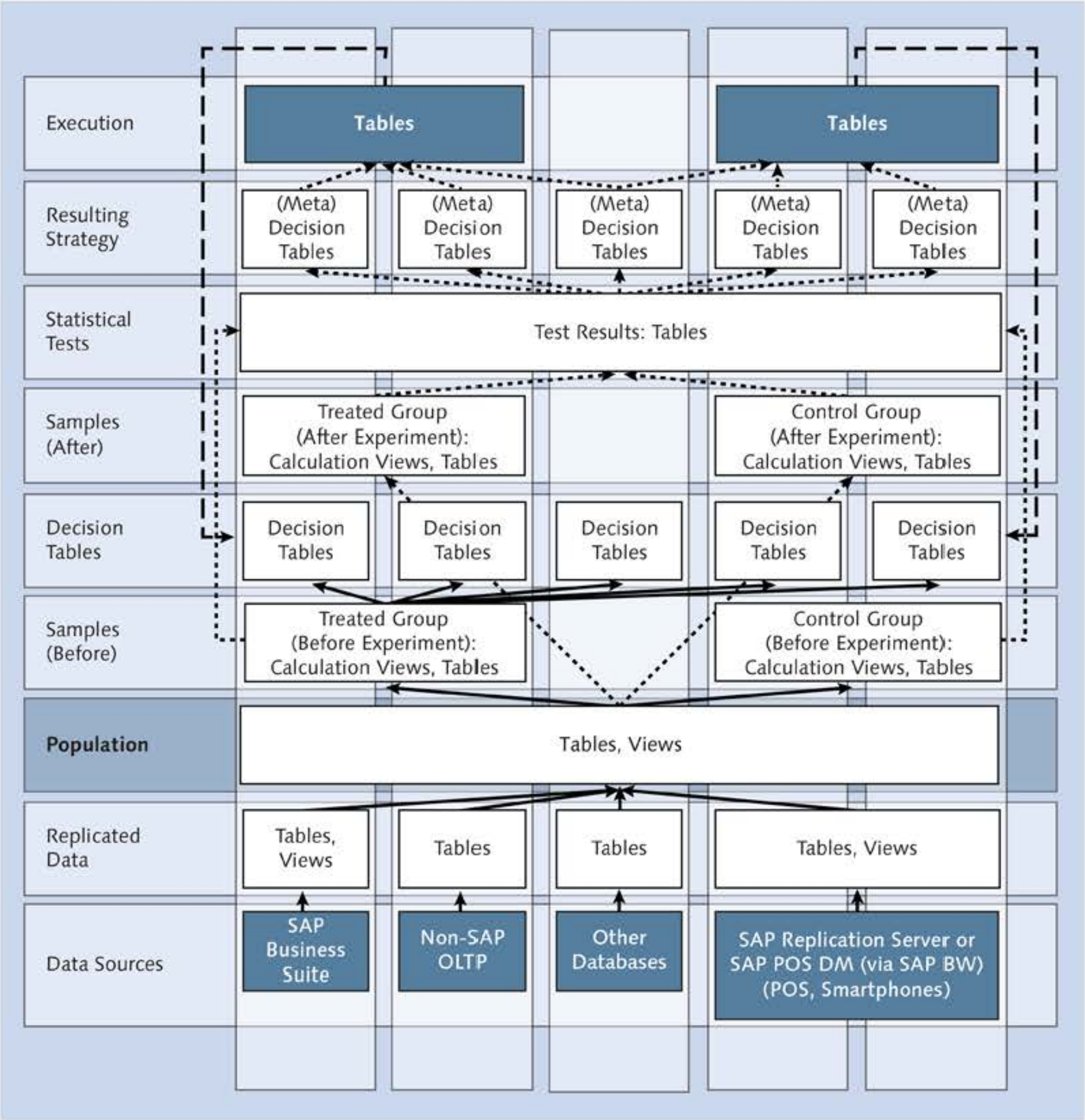


Figure 7.6 Customer Behavior Data Architecture

Data Sources

A variety of data sources

We are assuming that our solution gets its data from a number of sources. For SAP customers, the most important one will be SAP Business Suite, SAP CRM, and SAP ERP plus the solutions addressed within the value maps mentioned in Section 7.4.1. The data of these solutions are either already in SAP HANA—as in the SAP Business Suite on SAP HANA scenario—or can easily be brought there via SAP Data Services or SAP Replication Server.

SAP Data Services for external data

On top of that, SAP customers as well as non-SAP customers will probably use other, more specialized or proprietary solutions, especially in retail for which customers have always been creative in terms of tweaking SAP products. (Based on our experience, industries such as retail, financial services, telecommunications, and public utilities are the ones that struggle the most with the upgradeability of their systems.) As a general rule, data from more or less all products for data generation can be acquired via SAP Data Services (take a look at the impressive list in Chapter 2, Section 2.1.3, under "(Meta) Data Integration with non-SAP Products."

SAP Data Services will probably also be your preferred tool in terms of getting hold of data in other databases. Apart from data in classic databases, SAP Data Services can also get information from Apache Hadoop. Another advantage of SAP Data Services is that data can be cleaned, quality checked, enriched, and transformed on their way. With transformations, we do, however, urge caution: keep in mind our comments about SPOTs in "Replicated Data" in Section 6.5.2 and under "Replicated Data" within the following section.

Data from POS systems

Finally, we will have to manage data from real-time sources, chiefly data from POS systems. Such data can be collected in a central database and picked up from there via SAP Replication Server; many airlines use such an option for their booking data. Alternatively, SAP can also deal with POS data trickling in from the checkout (trickle feeds). The respective solution is called SAP Point-of-Sale Data Management; SAP POS Data Management comprises the POS Inbound Processing Engine, which can receive data via BAPIs, IDocs, or an input interface (`Point Of Sale Transaction In`). Because POS Data Management can also be implemented on an SAP HANA-based SAP BW system, even data that are trickling in are available instantaneously.

Replicated Data

Mirroring source system tables

Data that have to be acquired from external systems via SAP Data Services are first saved in (persistent) tables on the SAP HANA side of things. The structure and contents of these tables should more or less mirror the corresponding tables in delivering systems. There are two reasons for this:

- If you are filtering, pruning, or modifying data in the course of your data-acquisition processes, later on you won't know exactly what your insights are based upon—that is, which source data they were derived from.
- Do you really know which data might interest your organization in a couple of weeks, months or years? One of the basic concepts in big data is that *all* data are going to be useful sooner or later. This is precisely the reason why giants like Amazon, Facebook, and Google have turned into data octopuses, collecting everything they can get their tentacles wrapped around.

Views instead of replication

With data from SAP Business Suite or SAP POS Data Management that are already within SAP HANA, you'll need nothing but views. With these views, you will only include fields and records needed for further processing. Remember that you are working with virtual objects only and therefore don't have to worry that much about additional storage capacities. Be generous!

From a technical perspective, you don't even need the views; all downstream layers can directly access source tables within SAP HANA. Using views does, however, make your structures more transparent and decouples sources from downstream processing. One caveat, though: if the data in your source system are not stable and are subject to ongoing change, such as sales documents in SAP ERP, then you may still want to replicate them.

Population

Structured according to analytical requirements

All data within the layer called "Replicated Data" are to be considered our population for further analysis. The structure of these data—that is, the domains—are driven by the structure of the source systems. This means that we should first restructure them according to our analytical requirements.

In our data model, we have only provided one layer for this; in reality, this process usually stretches across a couple of layers. According to the principles put down in Chapter 5 and Chapter 6, we need to say goodbye to hypotheses about the data and create data structures that are as flexible as possible in terms of incorporating new insights. In Chapter 8, we will focus on this topic.

Calculation views

In this chapter, we are, first and foremost, conducting experiments on customer behavior. We would like to change pricing-relevant parameters and check what consequences this may have. The fact that we are not trying to build permanent, self-adapting models but instead volatile, temporary experimental setups provides the opportunity for this layer to mainly consist of calculation views.

Samples (Before)

Mapping experiments using procedures

In Section 7.5.1, we gave you a short overview of the functions that are available in SAP PAL and R for sample sizing and sampling. Regardless of which tools you would like to use for this, you are going to end up with procedures in SQLScript. These procedures will then flow into calculation views and/or tables; which one you are going to use depends on how virtual or how persistent you want your samples to be. Normally, you don't have to keep these data once your experiments have been evaluated.

Consider LO:

- We would like to change prices in real time and then learn how customers react.
- If the customers' price sensitivity is high, these price increases will be reversed more or less immediately and therefore won't cause a lot of damage.
- If, however, customers don't react by curtailing their spending, then the change will be extended to all articles step-by-step.
- In both cases, the sample will only be used to conduct one single experiment, serving its purpose after a short time. It does, therefore, often make sense to use views instead of persistent structures for samples.

Treated group and control group

Figure 7.6 shows two different groups on the SAMPLES layer. Much as in medical research, you will need two different groups: a treated group in which you make the change, such as a price increase, and a control group in which everything remains the same.

Decision Tables

Changing values within decision tables

For objects within the treated group (again, defined by a combination of store, article, customer, and so on), you should change one single detail. In our scenario, this happens by modifying values in control tables, which are either classic customizing or condition tables within SAP or decision tables within SAP HANA. Products for data generation use such tables to manage operational processes. With the control group, customizing/condition/decision tables will remain unchanged.

Alternatively, you could also consider sending messages to affected applications, asking them to make corresponding changes and to confirm these changes with an exact timestamp.

Samples (After)

Process new data

Once the adaptions in decision tables or within products for data generation have been made, your products for data generation will produce new data that will be sent to SAP HANA, ideally in real time. Once again, you can draw samples from these new data using the same procedures as for the SAMPLES (BEFORE) layer. In Figure 7.6, we indicated this by inserting dotted lines between the population and the SAMPLES (AFTER) layer. The data flows related to evaluating the experiment can be reused as well, which is why they are also dotted.

Duration of the experiment

The duration of your experiment depends on the amount of data that have to be collected to draw reliable conclusions—that is, the sample size. When making a decision about this, there are a number of options to consider:

- One can assume that the population after the change behaves exactly like the population before the change regarding certain parameters—for example, the standard deviation. In this case, you can decide that the size of a sample after the change should be the same as the size of a sample before the change; the only thing you'll have to do is wait until enough data have arrived in SAP HANA.

- If you are making changes that could have negative consequences, you may want to limit the damage. In this case, you might want to limit the duration of your experiment and be content with the data collected during that time span. This does, however, come at a price; you may avoid further damage, but the risk of type I or type II errors is pretty high.
- You can go for the silver bullet, considering your population after the change as a whole new database for which sample sizing needs to be redone.

Statistical Tests

Are key figures/ value drivers affected?

Using the algorithms listed in Section 7.5.1, you are now going to perform statistical tests using the data of your treated and control groups before and after the experiment. These tests are used to find out whether the change you made also led to a change with value driver–relevant key figures or with value drivers themselves. Furthermore, you want to know whether that change suited you or not. The tests will be the basis for making decisions about next steps. Such next steps could be any of the following:

- Stop the experiment, reversing all changes to control tables
- Extend samples, repeat test, then decide again
- Implement change everywhere immediately

Obviously, these decision-making processes can also coexist and be implemented in parallel for different series of experiments. Furthermore, you can run more than one experiment at a time; Leech has enough stores to do so.

Resulting Strategy

Implementing decision strategies in procedures

In the end, your decisions need to be executed by procedures. Actually, these procedures can be identical to the ones that have been used to make the experimental change before, the only difference being that this time the change is applied to different objects or more objects.

The information—that is, which of these procedures (increase price, reduce price, and so on) is meant to be executed with which parameters (5%, 10%, and so on) for which objects (stores, articles, customers, etc.)—can be stored in decision tables on the RESULTING STRATEGY layer. These decision tables are different from the ones in the DECISION TABLES layer. Here, we are talking about control tables for our concept of metaprocedures. These metaprocedures, for their part, call procedures that modify the DECISION TABLES layer and also provide them with the input data needed; the purpose of the RESULTING STRATEGY layer is to store these input data.

Execution

Providing input data via metaprocedures

The procedures that maintain the DECISION TABLES layer need input data. They have to know which (existing) data in decision tables have to be replaced by what (new) data. Such input data can be provided in separate tables via metaprocedures. In Figure 7.6, we have only inserted two domains on the EXECUTION layer; from there, two dashed lines indicate changes to decision tables.

Running through experimental cycles again and again

Once the solution has run through a complete cycle from the bottom to the top of our data model (in reality, this should only take a couple of seconds), a new experiment could be started. It would, for example, be possible to specify that one experiment can only change the prices of 1 percent of your articles by not more than 10 percent. By running through such a cycle again and again, you would find out where the pain thresholds of your customers are.

Autonomous bots

Many such cycles looking at different parameters could run in parallel. One of them could test price changes per article, and another one could check price changes per store, town, or region. A third one might handle promotions (such as three for the price of two) that could have an impact on settings within the Markdown Management solution mentioned in Section 7.4.1.

Finally, you'll end up with a system of autonomous bots that are able to adapt important control parameters to your environment 24 hours per day, seven days per week in real time. Compare this to your SAP customizing projects so far.

[+]

Bots Out of Control

With all that enthusiasm for big data, you should not forget that we are talking about highly complex systems here. Due to their sheer speed and complexity, such systems cannot be easily monitored by the humans who design or use them.

This does bring with it certain risks. In stock trading (in which relying on such algorithms has been common for years), we have seen a couple of problems with these systems. On May 6, 2010, there was a so-called flash crash, the causes of which have not yet been unveiled. One does, however, suspect that it was a result of computer-controlled trading algorithms reacting to each other and thus spiraling the situation to uncontrollable heights.

Another example: on Amazon, a book called *The Making of a Fly* (Lawrence, Wiley-Blackwell, 1992), nothing but a biological textbook, was offered for about 24 million USD, because pricing algorithms of two booksellers were bidding against each other.

When building such complex systems that are meant to act more or less autonomously, you therefore have to make sure there is some kind of control room, and make sure it is designed well—that is, better than the ones at Chernobyl or Fukushima.

Everything is connected somehow. If you pull out a hair on your bum, your eyes water.

Dettmar Cramer, former German football player and trainer

8 Analyzing Sensor Data Automatically and Generating Metadata

Derek sighed. The beach seemed to be endless, his walk more so, and lunchtime was receding into the distance. He was getting very hungry. When he parked his RV at the southern-most point of the bay, he had taken a close look at the area on his hiking app and seen that the line of sight distance between there and the other end was not much more than three miles.

Figure 8.1 Beach near Fréhel, Département Côtes-d'Armor, France

As the beach wasn't straight like a French baguette but instead shaped like a croissant, he had generously added some 50%. So his destination—the cliffs—should have been about 4.5 miles away from where he started, taking about 97 minutes to cover at his normal trekking speed. Nevertheless, another look at the map showed that he had only covered half the distance after 95 minutes. In about half an hour, the last bus for today would depart *from the Northern end of the bay heading back to the camping ground. The very same place where his crispy baguettes were slowly getting softer, and soft croissants had probably become rock-hard cookies by now.*

His hiking app obviously didn't seem to know that making headway on wet sand, in the snow, or walking through long grass was a lot more arduous than marching on tarmac or a good footpath. He had been hiking in rough country before and had recorded these journeys, but ten percent or so of more arduous terrain did not make a real difference when calculating a ten-year average, nor had most of his ramblings taken him over ground that was exclusively either firm or muddy.

On his last expedition through the Scottish Highlands, Derek had, for example, first walked uphill on graveled side roads, then crossed a wide plain on a mixture of well-trodden sheep paths and marshy pastures, and climbed up a mountain, negotiating steep, boulder-strewn slopes, just to finally find himself knee-deep in a snowdrift. Depending on the mix of conditions that he encountered, the average speed calculated by his app could be used as an approximate value for future undertakings but was quite often totally way off.

Unfortunately, his hiking app could only compute one average instead of a number of them. Upon reflection, he thought this restriction was pretty hard to understand. After all, the app was running on a device that included GPS data, map data, and an altimeter, so in theory all the information it needed for calculating meaningful average values (classified by gradient and topography) was in this little black box that he now so desperately wanted to throw into the approaching waves.

In practice, all these data were as separate and distant from each other as he and his destination, the conspicuous rock formation to the north. Brought back to earth (or at least sand) with a bump, hungry and tired, Derek stuffed the useless high-tech toy into his backpack and soldiered on.

Sensors are everywhere

Our environment is full of sensors that can measure each and every kind of parameter. Our fridges, ovens, thermostats, air conditioners, and notebooks all contain temperature sensors; GPS receivers accompany us all the time in smartphones, tablets, or cars; and when we travel by plane, a multitude of measuring instruments keep an eye on important flight parameters, such as altitude, outside temperature, the plane's bank angle, and the functionality of its engines.

Large sets of diverse data

All of these deliver measurements every day, hour, minute, or second or even more frequently. If you want to gain insights from sensor data, then you need to bring together quite a few large and diverse databases. That sounds like the typical habitat for big data, doesn't it?

In this chapter, we are going to review a couple of issues that are specific to analyzing sensor data. We are not really interested in technical details but would like to take a higher-level view. Furthermore, we are not trying to identify dependencies or develop models (for this, see Chapter 5) or to build data flows that link sensor data to value drivers (see Chapter 6).

Specific metadata when analyzing sensor data

Our main topics here are the specific, metadata-related challenges that arise when processing sensor data. What kind of metadata do you need, how can you get hold of them, and how can you ensure that they are always up-to-date? One of the more specific questions we are going to encounter is whether measured values refer to a specific instant in time or to a time interval. From an analytic point of view, these two cases have to be treated in entirely different ways.

Machine-to-machine applications

To illustrate these somewhat more abstract thoughts, we are going to use an example from the automotive industry. Although this example is fictitious, it does depict a scenario that became reality under the name usage-based insurance (UBI), also known as pay as you drive (PAYD), quite some time ago. For Vodafone, the well-known telephone company, UBI is just one of many services within their machine-to-machine (M2M) area of business.

Acquiring/extracting metadata

Having sparked some thoughts about sensor-specific metadata in your brain, we will shine a light on how to use big data not only for analyzing data from sensors but also for extracting the metadata needed for such analytical systems. We are going to present a partially automated solution using speech recognition and text mining for this. As we go along, we will focus exclusively on this topic. The data model at the end of this chapter will deal only with metadata (and not with processing sensor data).

Simplifying communication

A solution like SAP HANA—potentially introducing additional data marts and new virtual objects—raises the bar in terms of data governance and metadata management. At the same time, however, SAP HANA's performance and its analytical capabilities (like text mining or machine learning) can also be used to turn the tables, partially automating the task of extracting and updating metadata in the first place. Such capabilities would—probably for the first time ever—put you in a position in which metadata actually reflect what's really going on in your systems, databases, and programs. In this chapter, we are therefore going to explore why SAP HANA may make metadata management and data governance easier instead of more complicated.

Having defined the scenario for this chapter, we are now going to concentrate on the specific issues and benefits for our hypothetical client. The key question is this: How can departments within a company, let alone organizations within different industries, keep each other informed about their metadata when they have very different knowledge bases and when these metadata refer to systems that are spread all over these organizations? To answer this, we are going to introduce you to one of the key ideas within this book: separating content-related metadata (like information about the name or the purpose of a sensor) from semantically neutral ones.

[»]

Dipping our toes into language philosophy

Semantically Neutral Metadata

Think of two common sensors in your car; one of them recognizes when the headlights are switched on, whereas the other tells you if the driver's door is closed properly. The sensor for the lights simply determines whether power is flowing to the respective bulb, and the sensor in your door could be a so-called *Reed switch* (a sensor to measure the proximity of two objects via a magnetic field).

Even if the Reed switch wasn't in your car but on the door of your fridge, *semantically neutral metadata* would be identical in both cases. Both sensors—the one in the car and the one in the fridge—capture the state of type "yes" or "no," both deliver data not periodically but any time they are "questioned," and for both the value delivered does not refer to a time interval but to a point in time. If the door of your fridge is closed now and if it was closed 10 minutes ago, this doesn't mean it has stayed closed for the intervening nine minutes and 59 seconds.

Another example: If you are taking or sending digital photographs, then there are metadata attached to them. Such metadata include the time and location of the photo, details about your camera, the lens you used, the aperture and the shutter speed you chose, and a histogram detailing whether your picture is over- or underexposed. You can add information to the picture's contents as well (such as "Aunt Agatha's birthday party"), which would also be considered metadata. The metadata about the time the photo was taken and the aperture would be semantically neutral; the metadata about Aunt Agatha is not.

Semantically neutral metadata describe the behavior or the processing options for data in a more general way. They are usually not directly related to the data's origin or contents but to (often mathematical) properties that become relevant when analyzing them. Which metadata are semantically neutral and which aren't does not necessarily depend on the data themselves. In the end, it's all about what you are planning to do with the data now or in the future. If you would like to build a routine to delete all pictures with red-haired Aunt Agatha or with your ex-girlfriend in them, then the information about who is in your picture is still content-related and not semantically neutral; it does, however, suddenly become relevant for processing. A corresponding semantically neutral field that is a more abstract version of "Aunt Agatha in it" could be something like "attitude toward content." Do you see the difference?

Benefits of semantic neutrality

After our short and regular excursion into SAP Solution Explorer—once again identifying appropriate solutions and products from SAP—we are going to return to the idea of semantic neutrality. We will add a couple of examples related to our scenario, explain the benefits of semantic neutrality in functionally heterogeneous, highly dynamic environments, and suggest some algorithms or data logistics solutions (products for data management) that can be used to collect and extract content-related as well as semantically neutral metadata.

Focus in terms of value creation

Subsequently, potential benefits and value drivers will again be in the spotlight. When looking at benefits and value drivers, we are mainly interested in those related to our modeling approach and not in benefits or value drivers whose reach is limited to special instances of applying that approach (like usage-based insurance). The concepts discussed in this chapter are meant to help our fictional car manufacturer realize the rewards from its new applications with a reasonable amount of effort.

Acquiring metadata versus processing sensor data

By now, you should be familiar with the structure of our case study chapters. As before, we are going to finish with a look at potential implementation scenarios and data architecture. With data architecture, we will zoom out a bit, trying to present an overall model for acquiring and handling content-related and semantically neutral metadata. Rounding out the picture, we will provide you with a few tips in terms of processing sensor data, also including types of data (e.g., RFID data) that are not relevant for this chapter's case study. In Chapter 9, we will move on to another industry, getting down to processing sensor data.

8.1 Handling Sensor Data

Sensor data controlling processes

Within manufacturing and especially in process industries, such as food or chemical industries (a cynic might consider these two the same), sensors have played an important role for decades. Sensors do more than monitor all systems on a plane, for example, making sure that they are operating as they are meant to; production processes are controlled by manufacturing execution systems (MES), which also heavily rely on data collected by sensors. It's not unusual for MES to interfere with, or even interrupt, production processes if the data of certain sensors exceed previously defined thresholds. In the printing industry, for example, sensors that detect a rupture in the paper running through a rotary printing press can stop machines within a fraction of a second, faster than any human ever could, avoiding further damage.

Combining data from different sensor systems

The greatest potential doesn't come from making individual sensors better or faster but from intelligently and flexibly integrating the data delivered by different kinds of sensors. Driver assistance systems in cars rely on the information delivered by radar and ultrasound sensors plus the images of cameras sitting behind the windscreen and identify traffic signs, road markings, other cars, pedestrians, and animals. Each and every carmaker, as well as the Internet giant Google, is working on autonomous vehicles—cars in which people sitting at steering wheels can work or read magazines (believe it or not, quite a few pilots on intercontinental flights do the latter), much as they would on a train. All of that is already technically feasible. Right now, what you can buy at your local car dealership is not restricted so much by technical feasibility as by legal regulations—that is, the question of liability if something goes wrong.

Next step: Autonomous, self-learning cars

It's great to have an autonomous system that allows you to take a nap behind the wheel in the early morning traffic on the Cross Bronx Expressway in New York, but wouldn't it be even better to have one that isn't static but can learn from past experience, avoiding that road in the future? One could extend that idea to the driver assistance system. It's great if it comes with brilliant algorithms, even greater if these are updated over the Internet on a regular basis, but the platinum standard would be self-improving algorithms. It might also be a good idea to collect and process sensor data from preceding or following cars in the future, thus eliminating the risk of running into the car in front in a traffic jam as you go around a bend or over a steep hill. Would the data model of the driver assistance solution be flexible enough for this? Well, not if it has anything in common with most data models we have seen in ERP or data warehousing environments so far.

Industry 4.0 as a strategic initiative

In terms of possible applications in manufacturing, even politicians—though they usually don't have a reputation as thought leaders in IT—have become aware of the opportunities. In 2011, a new strategic initiative called "Industry 4.0" based upon the Internet of Things has been set up in Germany. In this context, the German Fraunhofer research institute is, for example, looking at how to build self-configuring MES.

Correlations and inductive reasoning

When dealing with sensor data, the principles defined in Chapter 1, Section 1.4.3 and Chapter 5, Section 5.4 are particularly important:

- We are approaching data without prejudice, which means that we are not trying to reject or verify preconceived opinions, hypotheses, or models; instead, we want to let our data speak, scanning them for patterns.
- With dependencies, we are not just interested in the *why*, but in the *what*. If a driver who brakes significantly more often than others is more likely to get involved in an accident, then she or he is a higher risk from the insurer's point of view, regardless of why she or he hits the brakes so often or whether he or she caused the accident or was the unfortunate victim.

[Ex]

Sensor Data in Spectator Sports

A good example of inductive data analysis occurred in March 2014, when *The Register* reported (in the article "SAP: It Was Our Big Data Software Wot Won It for Germany") that the national football team was using SAP's big data solutions to analyze sensor data in preparation for the world championship in Brazil. Positioning chips were put into the ball and sewn into the clothing of the players, making it possible to analyze not only a player's overall performance, but even individual moves within a match.

The football addicts among you know what happened some four months later.

8.1.1 Sensor Data are Heterogeneous

(Mathematically) diverse data

Our initial thoughts about this topic indicate that we will have to deal with very different, very diverse attributes and facts. By "diverse" we don't mean diversity in terms of formats and data structures; these can easily be unified, and appropriate (industry) standards for this—such as XML—have been around for quite some time. We are referring to more fundamental differences between semantically neutral metadata. Such differences are related to, for example, the following factors:

- Accuracy (reliability)
- Level of measurement, dimension, and unit
- Granularity, time difference, and time dilatation
- Reference date and reference period
- Aggregateability and aggregation functions
- Basic versus calculated versus restricted key figures
- Completeness

Accuracy (Reliability)

Sensors measuring with differing accuracy

Measurements for the same data can come at very different levels of exactness. The readings of a barometric altimeter are influenced by many factors and can easily be 100 feet off. For a hiker, this does not make that much difference, but for an A380 landing at night and with thick fog over the runway it does. Laser or radar altimeters (the latter are used for autolanding) are much more reliable.

When searching for outliers or sizing samples, you need to know the normal and expected standard deviation due to errors in measurement of your data. If you underestimate how dispersed your data are, then the following mistakes can result:

- You consider normal fluctuations to be significant outliers.
- Your samples are too small (to lead to valid, robust conclusions).
- You are underestimating the standard deviation of your population; this often means underestimating risk.

Overestimating dispersion would, for example, make you select samples that are bigger than they have to be, thus causing additional costs or delays.

Maximum accuracy not always necessary

Faulty measurements are not desirable. On the other hand, it doesn't make sense to always go for the most expensive, most precise sensors as a matter of principle. From a data-modeling perspective, all you need to know is what level of accuracy you may realistically expect. Using a couple of low-price sensors that correct each other is often cheaper and safer than using just one highly accurate measuring device. However, bear in mind that no matter how precise a sensor might be, it can still be faulty. If you know the expected measurement errors of your sensor, then statistical methods will tell you by how much the total error can be reduced if you combine data delivered by more than one of these.

Another way of dealing with inexact data is to group them using categories (from value A to value B) to turn readings on a ratio scale into ordinal ones, thereby no longer allowing erratically fluctuating individual values to make you nervous.

Level of Measurement, Dimension, and Unit

Nominal, ordinal, interval, and ratio scale

You can only process sensor data correctly if you know their level of measurement—that is, their unit and their dimension. A measure's dimension defines what it is supposed to measure (pressure, temperature, altitude above mean sea level, volume, and so on), and the unit tells you how to interpret specific values (pascal, degrees Celsius/Fahrenheit, feet, gallons, and so on). There can also be measurements without dimension and unit (think of our examples with the closed car/fridge door or think of the number of apples on a tree).

Granularity, Time Difference, and Time Dilatation

Synchronizing time

Sensor data come in different granularities. Some sensors deliver hundreds or thousands of values per second, whereas others only deliver one result per hour or per day. Hence, a measurement usually needs a timestamp, and if the granularity is really high, meaning you are receiving quite a few values, then this adds another problem: synchronizing measurements with different devices in different locations. If one of these locations is a fast-moving spaceship or even a relatively slow moving satellite, then you will also have to deal with velocity and gravitational time dilatation, as described by Einstein, who discovered that as velocity and gravity increase they both slow down time. Clocks tick slower in a fast-moving spaceship than on earth (due to relative velocity time dilation), whereas time on earth runs slower than it does far away from our planet and its gravitation (due to gravitational time dilatation).

Relativity theory is practical and relevant

Such thoughts are more than mere intellectual speculation for geeks or science fiction authors. Your car navigation device would not work properly if time dilatation between the atomic clocks on board the GPS satellites and the counterparts of these clocks on Earth weren't taken into account when calculating your position.

Reference Date and Reference Period

Instant in time or time interval

The question of when, from where, and how often a sensor is generating data is only one time-related question that needs to be answered before processing such data. Another one is whether measurements refer to a certain point in time or to a period of time (think back to our example with the fridge).

If data refer to a point in time, then their timestamp will tell you which point in time that is. If a measurement refers to a period, then you also need to know when exactly that period began and when it ended. Imagine a sensor linked to a photoelectric barrier in a supermarket, counting the number of customers passing the barrier. Whether 2,000 customers are a lot or very few depends on the time of measurement (3 a.m. or 6 p.m.); it's also important to know whether this number refers to customers per hour, per day, per week, or per month.

Measuring when and how often

Summarizing what we have said so far, the time and frequency/granularity of your measurements will depend on your requirements and what is technically possible, but no matter what you do in that respect you will only be able to process your measurement data if you know the answers to the following three questions:

- Does the value you have refer to a point in time (e.g., air pressure or temperature), or does it describe a period (e.g., number of customers or amount of data transferred)?

- If a value refers to a point in time, exactly which instant are we talking about? Quite often, this is the time at which your sensor measured the respective number, but even if the sensor is not aboard a spaceship, that time does not have to be identical to the data's time of arrival in your system.

  Data can be transferred at the speed of light, but even at that speed a signal from Auckland, New Zealand to Casablanca, Morocco—some 12,200 miles—will travel for 0.07 seconds. But data rarely travel as the crow flies (or via the orthodromic distance, as defined in Chapter 5, Section 5.5.2), and even if they did 0.07 seconds is half an eternity for high-frequency traders on the world's stock exchanges. Thanks to platforms like SAP HANA, the definition of fast and slow is changing in the real economy as well.

- If a value refers to an interval of time, then which interval exactly? We need the interval a) to make sure that the periods the measurements refer to do not overlap and that objects are not counted twice and also b) to properly interpret data; remember our example of counting customers in a supermarket.

Aggregateability and Aggregation Functions

Summation might not work

Whether sensor data can be aggregated and in what way (for example, added up) depends on the type of data and on their reference period. With a photoelectric barrier counting customers, individual values can be added as long as their reference periods do not overlap. If we had a total for each day of a month, then totaling them would give us the number of customers for the whole month, but if we forget to reset the counter of the photoelectric barrier at the end of each day, then the values delivered will have overlapping (nondisjoint) reference periods. The first value delivered will be the customers of the first day of the month only, and the second value will be the total of customers on the first plus customers of the second day. If we simply keep adding up these data, the number of customers we'll end up with will be far too high; if the month had 31 days, the first day would be counted 31 times. Hence, there is a relationship between the point in time or the time interval a number is referring to and its aggregateability.

Example: air pollution (CO_2 concentration)

For some measurements, there is no meaningful way of accumulating them. If a sensor is measuring the air's CO_2 content on a busy road and comes up with a percentage, we cannot just add up these numbers. The values do not refer to a period but to a point in time, meaning that the reference period for each individual value is infinitely short, and summing them would lead to totals that are higher than 100%, which is clearly nonsense. Calculating an average might make more sense here, although this still does not solve the problem of different reference periods. In reality, it makes a difference whether the CO_2 level was high for 23 hours and low for just one or the other way around, even though a simple, nonweighted average of both numbers would be the same.

Totals determinable but not interpretable

For some data—for example, the number of employees per department—one can calculate a total across certain characteristics (e.g., by departments) but not for others (e.g., across months). In the latter case, choosing an average instead would be more sensible; and for the CO_2 value example, one could ask whether the fact that some months have 31 days and others only have 28 should also be taken into account.

From a purely arithmetic point of view, you could, however, also add up employees within departments month by month, but the number would be meaningless at first glance. Other data—for example, categories such as date or gender—can't be added at all, and it wouldn't make sense to try to do so. With nominal data, you don't get very far at all using classic aggregation mechanisms, such as average, minimum or maximum; other examples of nominal data with limited aggregateability are postcodes and names. Nevertheless, in all four cases (dates, genders, postcodes, and names), you could think about grouping your data (from value A to value B) and/or counting them individually.

Summary values from the bird's perspective

It is indeed often more useful to look at summary values—meaning aggregated values—rather than individual values. Seeing the world from a bird's-eye perspective, we often see links that worms (particularly the very intelligent worms) would never have seen. Summary data often provide better insight; with unprocessed, raw data, we might not see the forest for the trees. Classic aggregation functions for data are sum, count, maximum, minimum, average, standard deviation, and median.

Estimated time of arrival

Navigation devices in your car keep recalculating your estimated time of arrival. They don't, however, do this on the basis of your velocity during the last 10 milliseconds (the latter information is needed because a velocity cannot refer to a point in time but only to a time interval); instead they use average values or moving averages calculated over a longer period to continuously enhance their original estimates, which might only have been based upon road types and expected traffic.

Aggregation properties are stable

The good thing is that for most data their properties in terms of if and how they can be aggregated are relatively stable, which means that we are dealing with semantically neutral metadata; even better, the number of possible characteristic values we are confronted with when it comes to elements like aggregateability and possible aggregation functions is manageable. With aggregateability, there would just be two (yes or no).

Basic, Calculated, and Restricted Key Figures

Does a sensor deliver calculated values?

With measurements, one can differentiate between basic, calculated, and restricted values. The differentiation among these three types of facts is based upon the idea of basic key figures, calculated key figures, and restricted key figures in SAP BW. For example:

- A sensor can provide you with the distance to the car in front of you measured in feet. This would be a basic key figure.
- It could, however, also do some basic computations using your current velocity and delivering the distance in seconds—a calculated key figure—instead.
- If it only delivers that value if the distance was less than 2.5 seconds, then the sensor would prefilter the results, thus coming up with a calculated and restricted key figure.

All three values would have to be processed differently, and it is often useful to know which kind of value you are receiving.

Completeness

Expecting values to go missing sometimes

Capturing and transferring sensor data does not always go smoothly, so it is important to know when and if you should allow for data to be incomplete. If you should, then you need to know how to detect that something is missing. If you expect a sensor to deliver a measurement every 60 seconds and if each measurement has its own timestamp, then gaps within your data flow can be easily identified, but if a sensing element is only supposed to send something if a certain event occurs (for example, the fridge door being opened), then you can never know that it might have missed an individual event due to a malfunction.

Filling the gaps

Once you know that data records are missing, you then have to decide how to deal with that situation. You may simply decide to ignore absent data; alternatively you could try to fill the gaps with appropriate mathematical–statistical algorithms. SAP PAL comes with a `SUBSTITUTE_MISSING_VALUES` function that can replace missing values by the corresponding *mode* (the value that appears most often), mean, or median.

8.1.2 Interpreting Sensor Data within Their Context

Do not look at sensor data in isolation

When looked at individually, certain sensor data have very little or no relevance or value at all:

- **Walking speed**
  One example of this is Derek's average hiking speed in our introductory story: without knowing on what terrain, with which gradient, and under what weather conditions (sunny, windy, snowing, etc.) this average has been accomplished, its value for estimating the duration of today's tour is very limited.

- **Emergency braking**
  Another example can be found with modern driver assistance systems, which can recognize obstacles such as other vehicles, humans, or animals and if necessary initiate autonomous emergency braking if something gets in the way.

  If, however, the obstacle is not standing still but also moving, then it may be out of your way before you would have reached it; in this case immediate emergency braking would be an inappropriate reaction that could, in turn, cause someone to run into you from behind.

  But then again, if that moving obstacle suddenly stops in front of you like a rabbit in the headlights, you really need to hit the brakes. When you do that, and how hard you brake, depends on your reaction time and your braking distance.

  This means that to decide about emergency braking a driver assistance system will need data regarding objects you could collide with;

it will have to know whether these objects are moving and if they are in which direction, at what speed, and so on. Furthermore, they also require data about your own speed, the properties and condition of the road, your personal response time, and the stopping distance that is realistic based on this data.

- **Energy consumption**
  You want to evaluate the energy consumption of a cooling system that is required to generate a storage temperature of -18°C. Whether a certain level of consumption is to be considered high or low depends—among other things—on the outside temperature.

  If your cooling system is attached to a refrigerated container used to ship frozen produce around the world, then you can only judge the energy efficiency of different types of brands if you link their consumption to the outside temperature at the various, potentially exotic locations that the container has passed on its journey. Even air pressure or swell may have an impact on energy consumption. Sometimes such considerations are obvious; at other times you first need to detect that a certain outside factor should also be taken into account.

Only one sensor = more false positives/negatives

As shown by this refrigerated container example, sensor data usually have to be interpreted while keeping the appropriate environmental conditions in mind. Such environmental conditions are often data collected from other sensors. Admittedly, there are cases in which one single kind of measurement is sufficient to make a decision—for example, the printing press sensors or a fire detector in a hotel room. Nevertheless, relying on just one dimension can lead to a higher risk of making a mistake. Whether this is acceptable or not heavily depends on the costs of additional sensors and the costs of false positives and negatives.

A smoke detector that was afraid of toast

In a holiday home (the one at which the photo in Figure 8.1 was taken), one of us came across a smoke detector that was so sensitive that it even responded to a toaster. Every morning, it started to howl crazily and could only be appeased by assiduously waving a towel underneath it. We decided that bothering your neighbors might be preferable to suffocating in your sleep, and besides the holiday home was in the middle of the Scottish Highlands with the nearest neighbor far away.

As a rule, however, keep the following in mind: In most cases, you will have to combine series of measurements from different sensors to gain useful insights. Because this is becoming a lot easier as big data and sensors get cheaper all the time, expectations in terms of decision quality are increasing accordingly.

[+]

More than One Sensor versus More than One Dimension

Under "Accuracy (Reliability)" in Section 8.1.1, we mentioned that it is sometimes both cheaper and better to use 10 less accurate sensors than one precision device.

We would like to emphasize that we are talking about something different here—that is, not about having more than one sensor of the same type but about measuring more than just one factor (for example, the energy consumption *and* the outside temperature *and* the swell when monitoring the refrigerated container).

8.2 Scenario: Cooperation among Car Manufacturer, Telephone Company, and Insurance Firm

Car manufacturer under pressure

The Japanese car manufacturer Maneki-neko K.K. (MN; in English, "Beckoning Cat") makes family cars and is one of the global technology leaders in that market. Although some famous German brands are gradually starting to fit their cars with adaptive cruise control or heads-up displays (HUDs), often only including such options in high-end cars, MN has been offering these technologies even in its smaller vehicles for years. Nevertheless, MN, like other car manufacturers, is suffering from price wars and decreasing contribution margins.

Customers don't pay a premium for technology

The fact that MN's products come with more features does indeed make consumers prefer them at the same price, but it doesn't create any elbowroom for price increases. Some fainthearted tests in the United States have shown that in the case of price increases many people would choose cheaper, technically inferior cars from India.

Data gold as a new source of income

To tap into new sources of income, MN has started intensive discussions with a major insurer and a leading mobile telephone company. In the course of these talks, the partners developed the idea of selling cars that come with a mobile phone card and lifelong Internet access without any extra cost, very much like Amazon's Kindle PaperwhiteS 3G. The mobile phone connection is also there to transfer data from various sensors in the car to the mobile phone provider and the insurance company. Such data would include information about the car's speed, acceleration, braking, and distance to vehicles in front. It is proposed that the mobile phone provider will get these data free in return for supplying the lifelong telephone and Internet connections, but they expect the insurance company to pay for the same data.

New insights for engineering and marketing

MN hopes to be able to extend the agreement with the insurance company to similar contracts with other insurers, thus forming a new, lucrative business. Furthermore, the insurance company and the mobile phone provider have promised to provide MN with all lessons learned on the basis of these data. MN assumes it could use these insights to further develop its own cars and also for market research purposes.

Cars as data collection units

There is a reason that we selected a car manufacturer for this chapter's case study. Because cars today contain a variety of sensors, cameras, positioning devices, and mobile phone systems, a car need not only be seen as a means of transport; it also is a mobile data-collection unit, which is probably the reason that Google is promoting the development of autonomous vehicles with the same kind of verve that they invest in robotics—just another type of data-collecting device. We suspect that only a few makers of cars have fully understood how valuable their data can be and that they or somebody else might be able to generate higher revenues by selling data rather than by developing new propulsion techniques.

8.3 Exchanging Data: Costs, Risks, and Opportunities

No in-house resources for analysis

MN employs brilliant engineers and technicians, but its IT department is relatively small; most IT-related tasks have been outsourced. Within other departments, such as marketing, MN also lacks data artists, data scientists, mathematicians, and statisticians. On its own, MN would therefore be totally unable to cope with analyzing the data delivered by their cars, not only due to the giant volumes of data but also due to lack of knowledge and experience.

Joint venture and common team

This is why MN started to talk to potential partners. The mobile phone provider is experienced in processing large volumes of data—for example, thanks to handling call detail records (CDR; that is, data records that capture the billing-relevant information of a phone call). The insurance company already employs a few hundred actuaries (mathematicians specializing in insurance); for these employees, handling mathematical formulas and statistical algorithms is part of the daily grind. Quite a few of the insurer's employees also like the idea of doing something different from simply evaluating risks for a while. MN and its partners have therefore decided to set up a new joint venture and a common team. In the beginning, this team will comprise 15 people, to be extended to up to 50 members later on.

8.3.1 Problem: Partners Have Different Requirements

Before signing contracts, MN has to overcome one difficulty. Its partners are expecting that the data delivered by MN won't simply arrive as raw material. Neither the insurance company nor the mobile phone provider employ automotive engineers; both partners would therefore be challenged when it came to understanding the contents of the delivered data. Furthermore, the mobile phone company—having worked with similar amounts of data for quite some time—has already built a very abstract analytical system that has recently been migrated to SAP HANA. This analytical system blindly analyzes data, meaning that the possible results of the analysis are not restricted by *a priori* statements made by the developers and are therefore independent of the data's meaning or content. The approach of the mobile phone company therefore reflects our recommendations in Chapter 5 and Chapter 6.

Forwarding content-related metadata

To describe the meaning of data that are sent to partners in simple terms, MN's advertising department has devised a clever process. First, it plans to put together a list of all sensors assembled into vehicles plus the data these sensors are supposed to measure (some sensors measure more than one parameter). For each sensor, a short video will be produced that explains that sensor, its position within the car, its function within the system as a whole, and the values it is supposed to deliver. The soundtrack of this video will be transcribed into written text via a dictation system and will be linked to the video's time code. This link will supply the recipients with the ability to perform a full text search. People will be able to search the transcribed text for terms and jump to the place within the video that the term is being explained; all that will take is a simple mouse click on a word.

Tagging a video

The advantage of using speech-recognition functions compared to analyzing the soundtrack via a pure dictation system is that the recognized text will be directly linked to the time code of the video. This process is also called *tagging*. The link between the text and the time code makes it possible to jump directly to text within the video.

If you also had a suitable taxonomy of terms (for example, the one available in SAP Information Steward), one could also imagine designing a frontend that can not only search for terms that are contained in the video but also for related superordinate concepts.

Content-specific metadata upon request

The previously outlined approach to conserving and transferring knowledge is, however, only the second step within a new workflow to be set up among the partners. This is because the mobile phone provider and the insurance company will only need to semantically understand the contents of the data that have been delivered *after* the respective analyses have been conducted. Otherwise, both of them would have to waste a lot of time trying to understand stuff that turns out to be irrelevant later anyway.

Flexible data models with the partners

A lot more important for MN's partners are criteria like those mentioned in Section 8.1.1. Based upon this information, the partners are going to control their analytical systems. The mobile phone provider, in particular, is using a data model that takes no notice of preconceived dogmata but instead unveils unknown and unexpected relationships between data. To be able to process whatever kind of information there is, specific requirements (in terms of what input data have to be like) have been defined for each theoretically determinable fact; remember our concept of the space of options here. MN's sensor data can only be processed within such an environment if they are delivered with the respective metadata. For the time being, MN is struggling to deliver such semantically neutral, mathematical-statistical metadata.

8.3.2 Numerical Example

EBIT margin diving

The 89-year-old founder of MN, who is still actively involved in managing the business, considers all these considerations academic and theoretical codswallop. To finally convince the founder of the company, MN's CFO made a rough-cut calculation of the additional revenues that could be generated by this cooperative venture. At the moment, MN sells approximately 10 million vehicles per year, generating around $180 billion or on average $18,000 per car. Unfortunately, MN's EBIT (earnings before interest and tax) margin (*EBIT margin = EBIT / revenue*) has recently dropped to just 8% or $1,440 per car.

Partners are willing to pay for data

MN's potential partners from the insurance industry are willing to pay a commission of $1,500 to MN for every new car sold with a usage-based insurance package. For the insurance company, the resulting data would be useful in many respects:

- In the case of an insurance claim being made, the circumstances of the accident would be far easier to reconstruct. This in return would make it easier to reject claims from third parties and also from policy owners themselves. Enforcing claims against other drivers would also be facilitated.
- The data would be invaluable when it comes to calculating individual, driver-specific insurance premiums. By comparing the driving behavior of accident-prone drivers with those that have been driving without any accidents for decades, one could offer lower premiums to careful drivers, extend the market share within that segment, request more money from adventurous drivers, or let the competition deal with the latter altogether.
- Independent of individual cases, the data could be used to learn a lot about customers' driving behavior in general. Tracking movements and creating movement profiles would be a piece of cake, thus creating brand-new options when it comes to tightly focused sales pitches.
- Finally, working together with a car manufacturer provides the insurance company with a simple way to bind new customers to their products when they buy vehicles.

Data provision not costly

MN's CFO has also learned from his engineers that the costs for collecting and transferring the data would be relatively low. Because telecommunications costs would be paid by the mobile phone provider, the additional expenses for MN would be confined to a couple of minor vehicle modifications, resulting in an extra cost of about $100 per car. There would also be additional IT costs of about $5 million per year.

Breaking new ground in marketing

The marketing department has estimated that at least 15% of MN's prospective customers would take the new offering in the first year. To really hook them, the marketing guys suggest offering a rebate of 2% on the vehicle sales price on top of the advantageous insurance policy sponsored by MN's partner. Based on these numbers, the CFO has produced the following calculations:

- Due to required adaptions for the cars, the EBIT per car will decrease by $100. The 2% rebate is going to reduce it by another $360 on average, reducing it from the initial $1,440 to just $980. However, taking into account the payment from the insurance company ($1,500) the EBIT will again increase to $2,480 or almost 14% of the sales revenue.
- If the 15% of the customers mentioned previously represent purchases of 1.5 million vehicles per year, then the annual EBIT is going to increase from $14.4 billion to $15.96 billion—that is, by $1.56 billion. The EBIT margin for the business as a whole would increase by almost one percentage point.
- In the face of an additional EBIT of more than $1.5 billion per year, the additional IT costs of $5 million per year seem negligible. Even if these costs were 10 times higher than estimated and even if there were additional costs of the same order for consultants or conceptual work, then the project would still make sense.

Go from the patriarch

The CFO knows that MN's founder—though originally an engineer—wouldn't have gotten where he is today if he didn't have a soft spot for money. A presentation based around these numbers plus some notes from the discussion with the insurance company finally managed to convince even the patriarch to give these first preliminary considerations a thumbs up.

8.3.3 Conclusion: Semantically Neutral Metadata

Defining required metadata

To exploit the enormous potential yields, MN will first have to jump the hurdle described in Section 8.3.1. After joining forces with the CIO, the

CFO has developed a couple of ideas that he can discuss with the partners. The CIO proposes to contact the external consulting firm that developed the data model of the mobile phone provider. Together with this consultancy and two representatives from the IT departments of the insurance company and the telecommunication firm, a brainstorming workshop about semantically neutral metadata is suggested.

Partners also want to benefit from workshops

MN's partners have taken up a tough stance on that workshop. If they have to invest time and resources to help MN define their metadata, then they would also like to use that opportunity to clean up their own. Therefore, the following agenda has been defined for the workshop:

- As a starting point, 10 random data flows are to be picked from the mobile phone providers' products for data exploitation.
- These 10 data flows are then to be examined under the aspect of whether previous knowledge in terms of contents or the data they process seems to have gone into designing these data flows. If it has, then the respective processing steps, layers, and domains are to be modified, eliminating the impact of that previous knowledge.
- Subsequently, all parties would like to clarify whether these data flows contain any implicit assumptions about semantically neutral metadata, such as assumptions about the level of measurement or the probability distributions of the parameters processed by the data flows. At the same time, they should clarify whether there are any additional demands that should be made on input data but which have not yet been documented.
- The metadata list resulting from this exercise is to be structured and converted into a hierarchy. Furthermore, these hierarchies are to be used as the basis for another brainstorming workshop.
- Finally, all ideas that have been gathered so far are to be quality-checked by the external consulting partner, quantifying their level of narrow-mindedness exposed by each respective item of metadata—for example:
    - If the sensor is developing values of the type yes/no (for example, "Safety belt buckled"), such a value would sit on a nominal scale,

  meaning that addition and many other arithmetical operations cannot be performed with these data. Clearly, specifying the level of measurement within the metadata doesn't therefore lead to any major restrictions in terms of how the data can/will be processed.

  - If, however, metadata define that a certain measure on a ratio scale (for example, the braking effort applied by the driver) is always normally distributed, then a substantial number of insights and possible results is excluded right from the beginning. We wouldn't even think of using algorithms that assume that data are for example following a heavy-tailed distribution, for example.

- The metadata will then be cleaned once again and checked for completeness based on the recommendations from the consultancy.

Purpose of the Workshop

The main purpose of the exercise is to support MN's task of building the knowledge necessary to provide their partners with the metadata needed. In doing so, the partners will also have an opportunity to tidy up their own data flows, getting rid of content-specific metadata that hadn't been spotted before or that had crept in during the process of maintaining their data flows.

Furthermore, when it comes to semantically neutral metadata, all of the parties involved want these metadata to be exclusively explicit and transparent rather than implicit and presumed.

8.4 Solution: Extracting and Managing Sensor-Specific Metadata in a Big Data Environment

Which metadata are needed?

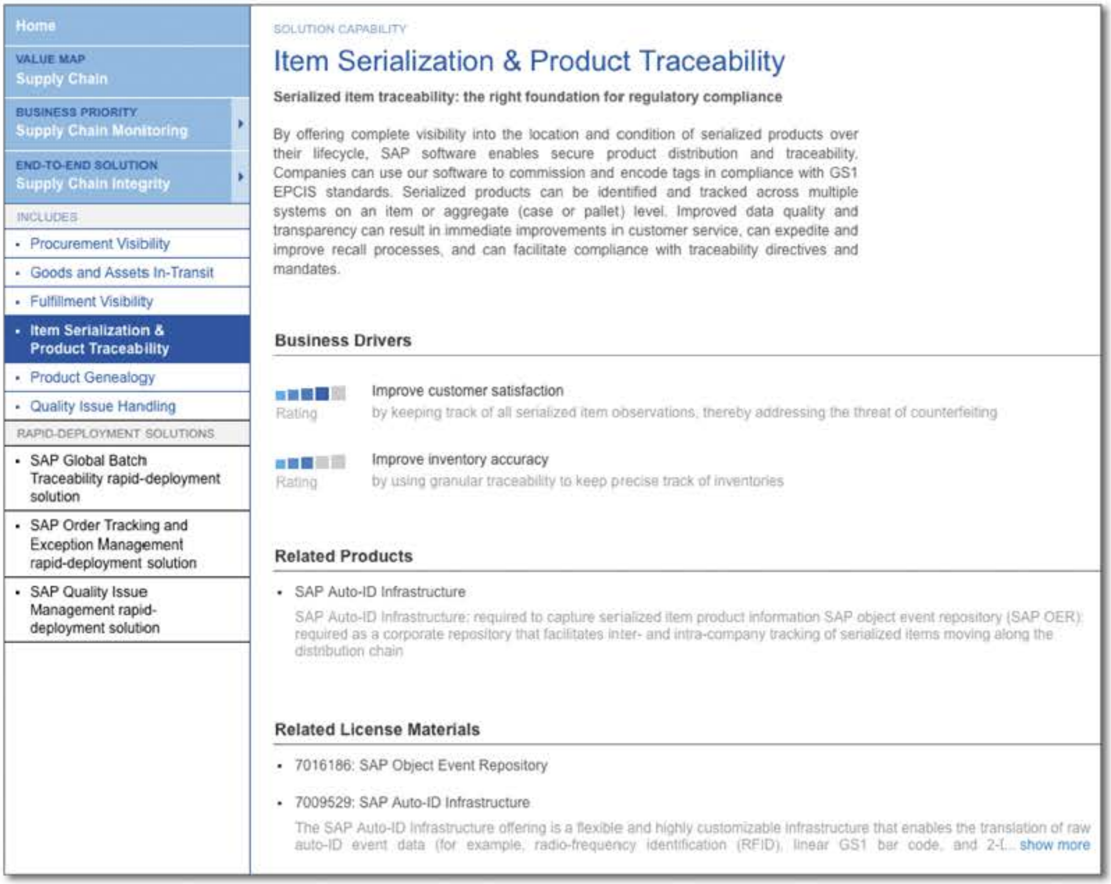
We have emphasized the advantages of an inductive approach that is data driven and free of prejudice a couple of times. Now, in this section we will address the following additional questions:

- Where do our data come from?
- What kind of metadata do we need to describe these data?
- What are the metadata going to look like?
- Where and how could MN store metadata?

8.4.1 Related Value Maps in SAP Solution Explorer

SAP Auto-ID Infrastructure

In Section 8.1, we mentioned a couple of application areas in which organizations may want to analyze sensor data. The German initiative Industry 4.0 makes it very clear that opportunities arising from large amounts of real-time sensor data are not restricted to only certain industries. Therefore, most of SAP's solutions related to sensor data can be found in cross-industry value maps—for example, the ITEM SERIALIZATION & PRODUCT TRACEABILITY solution within the Supply Chain value matrix under SUPPLY CHAIN • SUPPLY CHAIN INTEGRITY (see Figure 8.2). The product needed to implement this solution is called SAP Auto-ID Infrastructure (AII).


Home
VALUE MAP
Supply Chain
BUSINESS PRIORITY
Supply Chain Monitoring
END-TO-END SOLUTION
Supply Chain Integrity
INCLUDES
• Procurement Visibility
• Goods and Assets In-Transit
• Fulfillment Visibility
• Item Serialization & Product Traceability
• Product Genealogy
• Quality Issue Handling
RAPID-DEPLOYMENT SOLUTIONS
• SAP Global Batch Traceability rapid-deployment solution
• SAP Order Tracking and Exception Management rapid-deployment solution
• SAP Quality Issue Management rapid-deployment solution
SOLUTION CAPABILITY
Rating
Rating
show more

Item Serialization & Product Traceability

Serialized item traceability: the right foundation for regulatory compliance

By offering complete visibility into the location and condition of serialized products over their lifecycle, SAP software enables secure product distribution and traceability. Companies can use our software to commission and encode tags in compliance with GS1 EPCIS standards. Serialized products can be identified and tracked across multiple systems on an item or aggregate (case or pallet) level. Improved data quality and transparency can result in immediate improvements in customer service, can expedite and improve recall processes, and can facilitate compliance with traceability directives and mandates.

Business Drivers

Improve customer satisfaction
by keeping track of all serialized item observations, thereby addressing the threat of counterfeiting

Improve inventory accuracy
by using granular traceability to keep precise track of inventories

Related Products

- SAP Auto-ID Infrastructure

  SAP Auto-ID Infrastructure: required to capture serialized item product information SAP object event repository (SAP OER): required as a corporate repository that facilitates inter- and intra-company tracking of serialized items moving along the distribution chain

Related License Materials

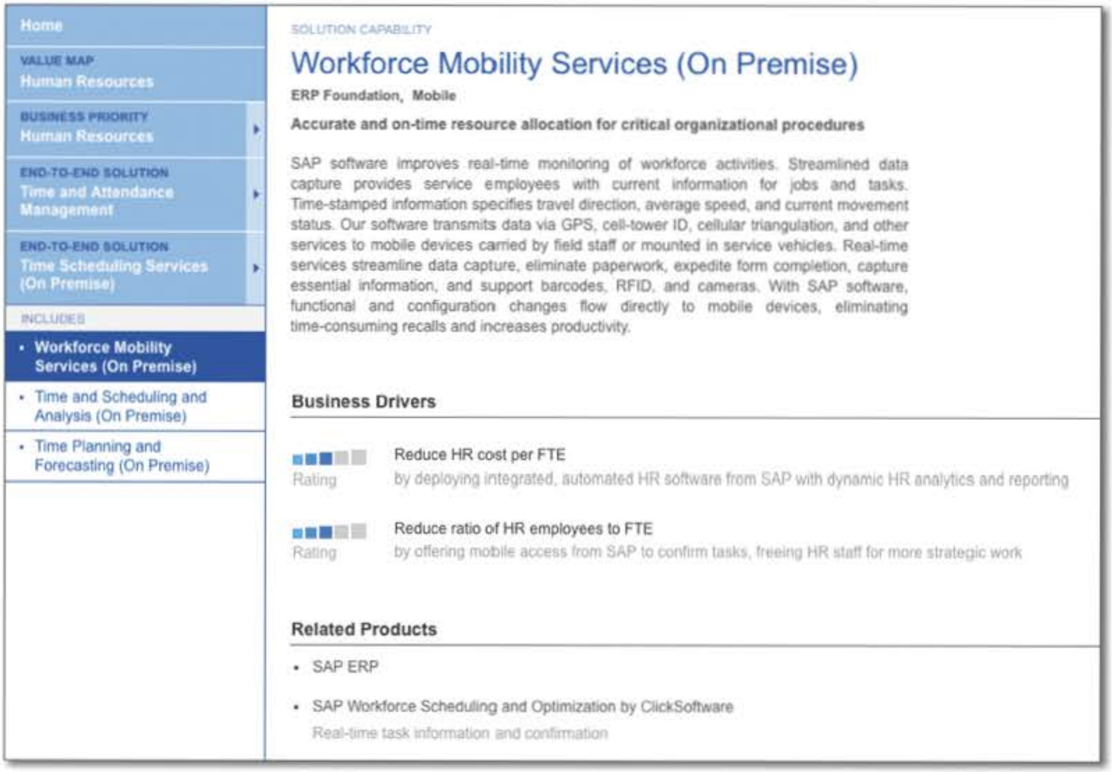
- 7016186: SAP Object Event Repository
- 7009529: SAP Auto-ID Infrastructure

  The SAP Auto-ID Infrastructure offering is a flexible and highly customizable infrastructure that enables the translation of raw auto-ID event data (for example, radio-frequency identification (RFID), linear GS1 bar code, and 2-[...

Figure 8.2 Item Serialization & Product Traceability Solution

Processing RFID data

SAP AII acts as the backend application for processing RFID (radio-frequency identification) data. The applications used to capture these data are distributed across a number of value maps and solutions. In the case of field service staff collecting RFID data, the respective functionalities can be found within the Workforce Mobility Services (On Premise) solution, part of the Human Resources cross-industry value map (via Human Resources • Time and Attendance Management • Time Scheduling Services (On Premise); see Figure 8.3). The partner product SAP Workforce Scheduling and Optimization (WSO) by ClickSoftware mentioned in this context records not only RFID events but also positioning data (via GPS, radio cell identification, mobile phone triangulation, and Assisted GPS [AGPS]). The main differences among all these positioning methods are their levels of accuracy.


Home
VALUE MAP
Human Resources
BUSINESS PRIORITY
Human Resources
END-TO-END SOLUTION
Time and Attendance Management
END-TO-END SOLUTION
Time Scheduling Services (On Premise)
INCLUDES
• Workforce Mobility Services (On Premise)
• Time and Scheduling and Analysis (On Premise)
• Time Planning and Forecasting (On Premise)
Rating
Rating

SOLUTION CAPABILITY

Workforce Mobility Services (On Premise)

ERP Foundation, Mobile

Accurate and on-time resource allocation for critical organizational procedures

SAP software improves real-time monitoring of workforce activities. Streamlined data capture provides service employees with current information for jobs and tasks. Time-stamped information specifies travel direction, average speed, and current movement status. Our software transmits data via GPS, cell-tower ID, cellular triangulation, and other services to mobile devices carried by field staff or mounted in service vehicles. Real-time services streamline data capture, eliminate paperwork, expedite form completion, capture essential information, and support barcodes, RFID, and cameras. With SAP software, functional and configuration changes flow directly to mobile devices, eliminating time-consuming recalls and increases productivity.

Business Drivers

Reduce HR cost per FTE
by deploying integrated, automated HR software from SAP with dynamic HR analytics and reporting

Reduce ratio of HR employees to FTE
by offering mobile access from SAP to confirm tasks, freeing HR staff for more strategic work

Related Products

- SAP ERP
- SAP Workforce Scheduling and Optimization by ClickSoftware
  Real-time task information and confirmation

Figure 8.3 Workforce Mobility Services (On Premise) Solution

Processing sensor data in general

RFID tags and GPS receivers are two central but nevertheless specific types of sensors. When talking about sensing elements in a more general way, one would also include sensors that measure physical, chemical, or medical data; some examples of this include temperature, humidity, pressure, and acceleration, concentration of methane or carbon monoxide, and blood glucose level. An SAP solution dealing with all of these, Sensor Integration, can be found in the Industrial Machinery and Components value map via INDUSTRIAL MACHINERY AND COMPONENTS • EMBEDDED PREDICTIVE TECHNOLOGY.

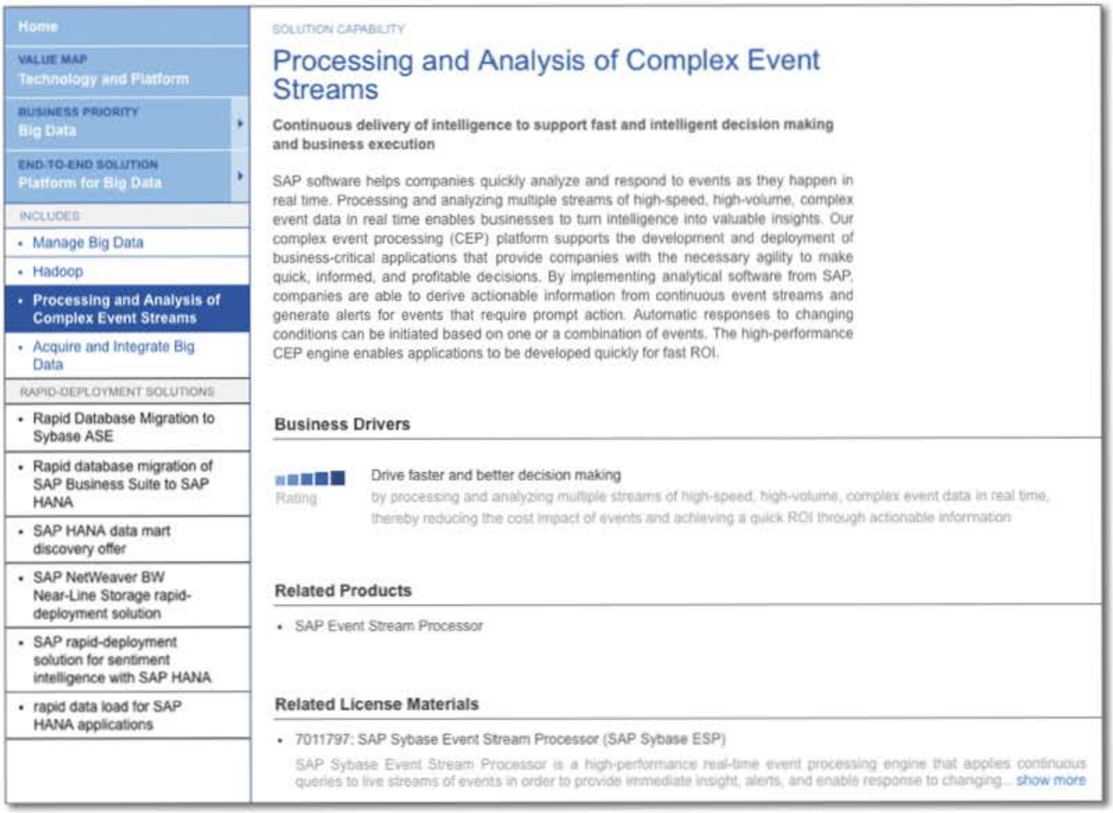
Data quality management

All solutions mentioned so far will be used primarily by MN's partners. For MN itself, the key topics at hand are extracting and storing metadata, possibly being involved with improving the data flows to their partners, and data quality—in terms of data integrity, consistency, accuracy, and so on—via the mobile phone provider. There are at least three SAP products that can be used in these areas:

- SAP Information Steward (as a product for data management in terms of managing metadata; for example, see Chapter 6, Figure 6.5)
- SAP Data Quality Management and SAP Master Data Management (as a product for data management mainly ensuring data consistency; see Chapter 6, Figure 6.5)

MN could also use two other products for data management for collecting data and data logistics:

- SAP Data Services for data logistics, found in SAP Solution Explorer under TECHNOLOGY AND PLATFORM • ENTERPRISE INFORMATION MANAGEMENT • DATA PROVISIONING • MIGRATE AND INTEGRATE DATA/ANALYZE AND IMPROVE DATA QUALITY
- SAP Event Stream Processor (ESP) for data generation (strictly speaking SAP ESP does not generate any data but picks them up from the sensors, making them available elsewhere; see Figure 8.4), found in SAP Solution Explorer via TECHNOLOGY AND PLATFORM • ENTERPRISE INFORMATION MANAGEMENT • PLATFORM FOR BIG DATA • PROCESSING AND ANALYSIS OF COMPLEX EVENT STREAMS


Home
VALUE MAP
Technology and Platform
BUSINESS PRIORITY
Big Data
END-TO-END SOLUTION
Platform for Big Data
INCLUDES
Manage Big Data
Hadoop
Processing and Analysis of Complex Event Streams
Acquire and Integrate Big Data
RAPID-DEPLOYMENT SOLUTIONS
Rapid Database Migration to Sybase ASE
Rapid database migration of SAP Business Suite to SAP HANA
SAP HANA data mart discovery offer
SAP NetWeaver BW Near-Line Storage rapid-deployment solution
SAP rapid-deployment solution for sentiment intelligence with SAP HANA
rapid data load for SAP HANA applications
Rating
show more

SOLUTION CAPABILITY

Processing and Analysis of Complex Event Streams

Continuous delivery of intelligence to support fast and intelligent decision making and business execution

SAP software helps companies quickly analyze and respond to events as they happen in real time. Processing and analyzing multiple streams of high-speed, high-volume, complex event data in real time enables businesses to turn intelligence into valuable insights. Our complex event processing (CEP) platform supports the development and deployment of business-critical applications that provide companies with the necessary agility to make quick, informed, and profitable decisions. By implementing analytical software from SAP, companies are able to derive actionable information from continuous event streams and generate alerts for events that require prompt action. Automatic responses to changing conditions can be initiated based on one or a combination of events. The high-performance CEP engine enables applications to be developed quickly for fast ROI.

Business Drivers

Drive faster and better decision making
by processing and analyzing multiple streams of high-speed, high-volume, complex event data in real time, thereby reducing the cost impact of events and achieving a quick ROI through actionable information

Related Products

- SAP Event Stream Processor

Related License Materials

- 7011797: SAP Sybase Event Stream Processor (SAP Sybase ESP)
  SAP Sybase Event Stream Processor is a high-performance real-time event processing engine that applies continuous queries to live streams of events in order to provide immediate insight, alerts, and enable response to changing...

Figure 8.4 Processing and Analysis of Complex Event Streams End-to-End Solution

Provisioning metadata

MN's main focus is to provide content-related and semantically neutral metadata to its partners. To do so, MN could use the following products:

- Metapedia within SAP Information Steward, a type of glossary in which customers can file technical or business terms together with their respective definitions, synonyms, and keywords and group these terms into categories or taxonomies.
- Metadata Management within SAP Information Steward, in which customers can create further customer-specific attributes for terms they would like to add to the Metapedia; MN could use such attributes to store the information listed in Section 8.1.1.
- Software for speech recognition/dictation and the tagging of audio and video files.

- SAP or non-SAP solutions for speech recognition, thus avoiding having to enter all information for the Metapedia manually.

8.4.2 Functional Requirements

Partners specify the requirements

The functional requirements are not primarily defined by MN but rather by its partners. First, MN has to provide master data for its vehicles, such as chassis number, year of manufacture, and optional extras fitted, and master data for sensors, such as their location within a car, types, releases, and lifespan. These master data are managed in the mobile phone provider's SAP Data Quality Management system and are distributed from there. MN may want to also store these data within its own systems; such (redundant) master data stored with MN are out of scope here.

Because MN will focus on metadata needed by their partners—regardless of whether these metadata are content-specific or semantically neutral—let's come up with an example of such data:

- All vehicles built by MN are equipped with a system that measures the distance covered during the last second at the end of each second. This number will be delivered together with a timestamp that stands for the end of the reference period. Because the reference period always equals one second, this indirectly also provides the average speed during that time in meters per second.
- At the same time, the vehicle can use the map data of the navigation system plus the images of a traffic sign camera integrated into the windscreen to determine the speed limit on the section of the road on which the car was driving during that second. Whenever the speed limit changes—for example, when the car enters a town or passes a sign in a construction zone—the respective new value is recorded together with a timestamp.

  These values are fairly exact, but not 100% reliable for a couple of reasons:

  - Map data can be old and obsolete.
  - The navigation device's GPS receiver cannot determine the car's position to an accuracy of one inch, and in any case the maps used are not that exact.

  - Traffic signs can be dirty or overgrown and might not be recognized properly by the car's camera.

- Despite these limitations, the car sends the data from these two sensors whenever it is connected to a mobile phone network. If there is no network coverage, then the data records are buffered in the car and transferred at the next possible opportunity. The structure of the data records for the distance covered/the speed driven looks like this:

  ```
  ( Sensor ID, Measurement ID, UTC Timestamp, Distance in Meters )
  ```

  The structure of the records containing the speed limit is:

  ```
  ( Sensor ID, Measurement ID, UTC Timestamp, Data Source, Speed
  Limit in mph )
  ```

Data origin

The sensor ID is a unique identification number for a certain sensor, and the measurement ID identifies an individual measurement of that sensor (you might need two fields here, because some sensors might measure more than one thing); for each sensor ID there are the respective master data within SAP Data Quality Management. The UTC timestamp is based upon the Universal Time Coordinated (UTC) that is used internationally and broadcasted automatically by a number of scientific institutes all over the world.

The distance in feet is determined by the number of turns of the car's wheels, taking into account the abrasion on the tires by continuously comparing and calibrating that measurement against data from the navigation device. The speed limit-related records also contain a data source (map or camera) and the number (speed) delivered by that data source.

The partner's view on the data

From the partners' perspective, the two record types simply contain facts (distance and speed limit) measured on a ratio scale; MN's partners are not interested in how to interpret these values but only in statistical dependencies. Typical questions they would ask include the following:

- Is there a relationship between actual speed and speed limit? If yes, then what kind of relationship?
- Do time lags play a role—that is, does the driver only start to brake a short while after passing a traffic sign that indicates a new speed limit or accelerate before passing a sign that indicates a faster speed limit than the previous one?

- Does the driver's behavior depend on the time of day, the day of the week, or the season?
- Do any of these facts have an impact on the likelihood of the driver getting involved in an accident? Are there any thresholds after which the risk of an accident increases significantly?

[»]

Interval and Ratio Scales

When defining the different levels of measurement (see Chapter 4, Section 4.4.3), we briefly mentioned interval scales. The difference between an interval scale and a ratio scale is that unlike values on an interval scale, values on a ratio scale have a meaningful, unique, and nonarbitrary zero value. For an interval scale, you can calculate only the difference but not the ratio between two values.

Content-Based Metadata (Not Semantically Neutral)

Making content-based metadata available

Coming up with information about the meaning, the purpose, and the functionality of sensors is not a major problem for MN. A team from marketing will simply have to interview the responsible engineers then follow a predefined schema and assemble a short video from the recorded replies. The video will then be processed with the computer linguistic algorithms mentioned in Section 8.3.1. All videos are then sorted by sensor ID and saved on a web server that is also accessible to MN's partners. The transcripts created for these videos are also to be analyzed using a text-mining tool (see Chapter 1, Section 1.1.2). This will extract further metadata that can be used for a fuzzy search that looks for words that are spelled incorrectly or have been used in a different grammatical flexion (e.g., "buy" and "bought"); furthermore, taxonomies can be applied to enable hierarchical searches.

Semantically Neutral Metadata

Making semantically neutral metadata available

In Section 8.1.1, we provided a couple of examples of semantically neutral metadata that could be important for our scenario. To decide which other metadata might be needed, you would have to know more about the processing mechanisms with the partners. In other words, you would have to know which attributes or facts they are planning to determine, how they are going to do so, and which assumptions have to be taken into account for these procedures. These insights result from the workshops mentioned in Section 8.3.3.

Example: metadata for distance covered

When measuring the distance covered during the last second, the metadata could look like the ones shown in Table 8.1. Remember, this table is only an example; in reality you will need a lot more metadata.

Metadata Attribute	Meaning	Capturing this (Meta) Datum for Distance Covered
Accuracy	Spread of random errors, those that—unlike systematic errors—cancel each other out on average (i.e., have an expected value of 0); assume no systematic errors here.	Standard deviation (in feet)
Level of measurement	Nominal, ordinal, interval, or ratio scale.	R (for ratio scale)
Granularity	Fineness of the values with respect to a certain reference characteristic; here, one value per second, so granularity refers to time.	*1/s*; reference characteristic: T (time)
Time difference and dilatation	Irrelevant in our case.	# or N/A (not applicable)
Type of time reference	Whether values refer to a point in time or a time interval; the distance covered at a point in time would be 0, so it must refer to a time interval.	P (period)
Point of time reference	With values that refer to a point in time, this is the point in time the value is referring to; with values referring to an interval of time, as in our case, this is the time of measurement.	UTC timestamp

Table 8.1 Semantically Neutral Metadata for Measuring Distances

Metadata Attribute	Meaning	Capturing this (Meta) Datum for Distance Covered
Time interval reference	With values that refer to a time interval, start and end time of that interval.	From *UTC Timestamp – 0 second* to *UTC Timestamp + 1 second*
Aggregateability	Can values be aggregated?	Y (yes)
Aggregation function(s)	How can values be aggregated? This field might contain more than one value—for example, for different reference characteristics; as a default, we assume that the reference characteristic is the same one associated with granularity.	SUM (summation)
Basic, calculated, or restricted value	We are talking about a basic value; the sensor is delivering the distance and not the speed.	B (basic)
Completeness	Do we have to expect missing values? If so, how many?	*Number of Missing Values / Number of Expected Values* (percentage)
Imputation	Are missing values meant to be imputed before analyzing the data?	N (no)
Algorithm for imputation	How are missing values meant to be imputed (see "Completeness" in Section 8.1.1 for details) ?	# or N/A (not applicable)
Context	Which other sensors deliver data that might help understand the measurements from this sensor? Examples in our case include rain sensor, thermometer, and traction of drive wheels.	Sensor IDs of other sensors

Table 8.1 Semantically Neutral Metadata for Measuring Distances (Cont.)

Dealing with Dynamic Environments

Choosing less granularity

Let us assume that one value for the distance covered for each and every second would be far too granular from MN's partners' point of view. The cooperative team therefore decides to aggregate these values in groups of 60, giving the distance covered over a minute. This distance will then be used to calculate the speed in miles per hour. In this specific case—unlike in the example shown in Table 8.1—imputation would suddenly be a must.

Once gaps have been filled using any of the options described in "Completeness" in Section 8.1.1, a routine in pseudocode that is based on data that uses a concatenated key (Sensor_ID and Measurement_ID) could be developed that would look like Listing 8.1.

```
program speed;
        speed_modified = 0;
        for counter = Measurement_ID to Measurement_ID + 60
        speed_modified = speed_modified +
        Strecke_in_Metern(Sensor_ID, counter);
        next;
        speed_modified = speed_modified * ( 1.875 / 50.292 )
end program speed;
```

Listing 8.1 Pseudocode to Aggregate Measured Distances

Weaknesses of classic approach

The code shown in Listing 8.1 has a number of serious weaknesses:

- The name of the program (speed) and the identifier for the variable used within it (speed_modified) make it clear that the code is tailor-made for a very specific purpose. If it was used for different facts with different meanings but identical semantically neutral metadata, then this usage would be confusing at the very least.
- The field name Strecke_in_Metern not only refers to a certain content (distance) but also to a very specific unit of measurement (meters). First, using the unit within the field name is simply unnecessary; that kind of information could also be put into the field's metadata. Second, this again restricts the reusability of the code, making it difficult to use in another country with different units of measurement.
- If after a while MN's partners come to the conclusion that looking at distances per minute is not granular enough and that they would like

to total 15 observations instead of 60, the program would have to be changed. The same applies if the code was suddenly used in central Europe, now having to deliver kilometers instead of miles per hour, in which case the formula at the end would have to be changed to *speed_modified = speed_modified * (3 / 50).*

Note that the two formulas might not be self-explanatory; if `speed_modified` contains the distance in meters driven in one minute, the car would cover 60 times that distance within an hour, meaning it would cover *60 * speed_modified* meters which is *60 / 1,000 * speed_modified = 3 / 50 kilometres*. The formula within Listing 8.1 has also been simplified by canceling a fraction (in that case *60 / 1,609.344* (1,609.344 is the number of meters within a mile).

- An array used within the code (`Strecke_in_Metern`) has been given a German name. This might be okay for programmers speaking that language, but it makes it a lot harder for others to read and understand the code. Those of you familiar with ABAP programming will be very familiar with this issue, because the majority of field names within SAP ERP are German acronyms.

  Documenting the content of a field with its name makes offshore maintenance unnecessarily complicated. Imagine that the program was written in China and the person writing it used Mandarin and didn't provide any other kind of documentation.

- Regardless of whether the preceding routine can be considered object oriented or not, it is not reusable for all other measurements with the same mathematical properties. The variables within it are linked to specific fields in specific tables that sit in a specific database.

  If the structure of the database or the underlying data changes—perhaps a new, more sophisticated sensor will provide the distance covered during the last 300 milliseconds rather than that covered during the last second—then the aggregation functions to be used will have to change as well. Adding 60 individual values no longer leads to the speed in meters per second. Are you sure you will remember to tell the programmers about that?

Not only is adapting programs an error-prone process, it is also a lot faster and less complicated to adapt centrally maintained metadata than to hunt down all routines that might be affected by this kind of change.

The information held in the videos produced by MN don't help either in that respect, regardless of sophisticated functions such as full-text or hierarchical search. Unfortunately, routines are not yet able to watch videos, decide whether they are affected by changes to the videos' contents, and implement the resulting changes autonomously without human interference.

Alternative: flexible data model

We are assuming that MN's partners have adhered to the principles laid down in Chapter 6, in which case they will not have produced routines like the one in Listing 8.1, but will instead have done the following:

- Their metadata repository contains a fact called `distance_covered`.
- For this fact, the metadata shown in Table 8.1 has been defined. These metadata have been amended in various ways—for example:
  - For which data records it would be allowable, or sensible, to calculate `distance_covered` has been defined.
  - Such a definition can be very wide or very narrow. It would be very narrow if calculating `distance_covered` would only be allowed for data records in which the `Sensor_ID` equals certain values, meaning that `distance_covered` would only be feasible for certain sensors. Whether you should be restrictive or generous with such definitions depends on the performance of your analytical solutions. The more sensors you could calculate `distance_covered` for, the wider your space of options (as defined in Chapter 6, Section 6.4.2) would become.
  - The calculation of speed or `speed_modified` from `distance_covered` would happen in a separate routine; the conversion factors to be applied would not be hard-coded but instead dynamically read from a table.
- Considering our remarks about language-specific variable names, one should obviously also reconsider identifiers such as `distance_covered`, `Sensor_ID`, and `speed_modified`.

Aggregation level to be determined by system

If your data model was sufficiently flexible, then you could also leave it to the system to find out whether 10, 20, or 60 individual values should be aggregated or whether better decisions would result if there were no aggregation. The conclusion for MN is that they will have to think about all of these factors in order to supply their partners with flexible metadata.

Cross-Referencing Metadata

Reusing metadata

Defining metadata within a metadata repository and not implicitly within routines (as done in Listing 8.1) has another crucial advantage: a set of metadata attributes can be treated as an object in its own right, which can be given an ID, thus becoming reusable itself. Instead of creating and storing metadata per sensor ID, one simply has to link a sensor ID to a metadata ID that describes the data of this sensor (plus maybe others) and their processing options.

Many sensors are going to deliver very similar types of values; think about the example of the headlights being on or off or the car door or the fridge door being open or closed. The respective metadata records still only have to be created and maintained once. This will dramatically simplify metadata maintenance and at the same time provide you with a much higher consistency in your data models without any extra effort—in fact, arguably with *less* effort.

Reference characteristics within SAP BW

SAP BW comes with a very similar option called reference characteristics. The difference from the concept presented here is that SAP BW does not connect characteristics and their metadata directly but via another reference characteristic. In SAP BW, you could, for example, define a characteristic called "amount" and then use inheritance, a concept from object-oriented programming, to pass on its metadata to other, referencing characteristics, such as amount in local currency, amount in transaction currency, or amount in group currency.

No separation between form and function

The principle "form follows function" might be great for architecture but doesn't make so much sense in metadata management. The preceding concept from SAP BW connects characteristics not only to metadata but also to other characteristics. As long as we are talking about fairly abstract fields (such as amount), this will not be a major restriction, but when it comes to characteristics that are totally different in terms of their meaning and contents, creating such a relationship isn't very wise. Within SAP BW, you could circumvent this problem by creating generic dummy characteristics, but, unfortunately, no customer we have ever encountered has thought of this option yet.

8.4.3 Building Blocks of the Solution

Within this chapter, the functional requirements for MN are centered on acquiring, managing, and providing content-specific plus semantically neutral metadata. Before inspecting potential technical solutions or products, we will as usual first clarify the functional requirements, discussing which algorithms could be used to capture, extract, explore, or make accessible the metadata.

Content-Based Metadata (Not Semantically Neutral)

Starting point: video interviews

To explain the data delivered in terms of their contents, MN is using tagged video recordings. The videos are recorded and prepared as follows:

- **Recording**
  In the course of a structured interview with an expert responsible for the respective sensor, some basic information will be collected. In doing so, MN is focusing on two topics:
  - One purpose of the interview with MN's engineers is to provide their partners' experts with knowhow about the use, purpose, and functionality of each sensor; this knowhow falls into the category of content-related, non-semantically neutral metadata. This information is only weakly structured, tailored around the needs of laymen, and only recorded on request.
  - The interview is also used to capture semantically neutral metadata as defined during the workshops with MN's partners and relevant for the data models of the partners. This part of the interview is highly structured and formalized.
- **Video editing**
  The recorded video is edited roughly to eliminate unwanted contents, such as idle times, confidential information, or slips of the tongue. In Section 8.3.1, we mentioned some exemplary software products that can be used for this purpose. The edited video will then be published as a preliminary version on MN's intranet and will be checked by the expert that was interviewed.

- **Speech recognition/text analysis**

  In the next step, the video's audio track is transferred into written text using professional dictation or speech-recognition software. This will lead to a raw version of the transcript that should again be checked by humans—for example, in the course of a crowdsourcing project. This check is only meant to make sure that spoken and written text are identical; the meaning of content must not be changed in any way.

- **Tagging**

  The written text (the transcript) is now linked to the time code of the video. Although this is theoretically possible without first transcribing the text—in which case the tagging software would do the transcription on the fly—such an approach wouldn't be accurate enough given the current state of technology. For the time being, we would not recommend proceeding without specialized dictation software or a cross-check by humans.

- **Text mining**

  In parallel with the tagging process, the transcripts can also be analyzed using text-mining tools. This analysis should be done for three reasons:

  - First, it can be used to extract key terms from the text. These key terms can be used later when performing searches.
  - Second, all key terms found in the text should be classified. This is going to make it possible to not only search for details such as "Reed sensor" but also for superordinate terms such as "proximity sensor." When using taxonomies, this will also be possible even if the respective term is not actually mentioned in the video.

    With big data applications there is now a tendency to use correlations instead of taxonomies. Creating and maintaining taxonomies is a time-consuming process, and quite often unique assignments of terms are not even possible. For example, does Douglas Adams' *The Hitchhiker's Guide to the Galaxy* belong to the category "science fiction" or to "humor" or "travel"?
  - Apart from the content-specific metadata collected in the first part of the interview and addressed under the previous two bullet points, semantically neutral metadata from the second, more structured phase are to be extracted. Text mining can be one way to

detect those data and make them available in a machine-readable format.

- **Availability**
  Once the video has been tagged and analyzed, it should be made available to partners via an extranet. With the video's first release, the extranet only offers a simple search function (full-text search within the transcript with the option to jump to the respective place in the video).
- **Metapedia**
  The keywords extracted via text mining and the semantically neutral metadata are made available within the Metapedia of MN's metadata repository.
- **Search algorithm**
  In addition to the simple full-text search, the partners are also able to search for the keywords extracted via text mining, to perform hierarchical searches, or to search for semantically neutral metadata. The search itself is going to be handled by a sophisticated tool that can also deal with fuzzy search terms or complex search requests. In this context, MN is thinking about its already implemented Google Search Appliance that has turned out to be quite useful when searching technical documents.

Knowledge transfer is being revolutionized

Ever more powerful tools for speech recognition not only provide an easy way of documenting systems or data models but also dramatically change computer-based knowledge transfer and learning. Quite a few leading universities already offer their courses as massive open online courses (MOOCs); snappy name, eh?

At the moment, many of these offerings are nothing more than simple video recordings of classic lectures; among the add-ons for this book (find them at *www.sap-press.com/3647*), however, you will find an example of a tagged training video, and a lot of the material provided by leading online training suppliers like *lynda.com* is tagged as well.

Semantically Neutral Metadata

Semantically neutral metadata to be structured

Standardizing the second part of the interviews with your experts will make it a lot easier to use text mining for extracting semantically neutral metadata from the recordings. Once extracted and structured, these metadata will be made available in a database to the experts that have been interviewed, asking them to cross-check and complete them. A workflow software supervises that process, making sure that the work gets done. When new technological solutions come along—for example, if new sensors are released—the appropriate workflows to update affected metadata are triggered automatically.

In the automotive industry with its extensive supplier networks, contractors often know more about specific components than the car manufacturer itself. It may, therefore, be a good idea to design both the metadata repository and the workflow solution as an open system (extranet) that can be accessed not only by the mobile phone company and the insurance company, but also by MN's suppliers.

Dealing with Dynamic Environments

Separating static and dynamic metadata

When defining semantically neutral metadata at the start of this chapter, we pointed out that the boundary between semantically neutral and semantically not neutral metadata is blurred. Semantically neutral metadata can be identified by two criteria:

- They don't refer to the contents or meaning of data but to their mathematical properties and to the options for processing them further.
- Semantically neutral metadata change rarely or not at all. If a sensor is delivering binary values (yes/no), such values can be counted but can't, not even in a hundred years, be added. Only if such a sensor were to be replaced by a new model that also tells you how far a door has been opened are we dealing with new data and a new metadata object.

Content-specific and semantically neutral metadata should be kept apart. Content-specific metadata are used to search data and can be considered to be characteristic values categorizing or describing the data themselves; they can also be the subject of analyses and reports. In contrast, semantically neutral metadata are used to construct data flows and to design analytical algorithms. It therefore makes an awful lot of sense to treat both databases as totally different beasts. In Section 8.5, we will explain that, sadly, metadata are not handled like this by most metadata repositories—not even by SAP's solution.

Cross-Referencing to Metadata

Defining metadata objects

Another reason that content-specific and semantically neutral metadata should be kept separate from each other is the reusability of metadata objects. If content-related and semantically neutral information are mixed when creating metadata objects, the reusability of these objects is restricted, as is the case with reference characteristics within SAP BW. Such restrictions already started with the naming of the metadata objects.

Let's look at an example: you can probably imagine that the mathematical-statistical logic and algorithms used to process cost-related facts from cost centers and profit centers could be very similar if not identical. If, however, you go ahead and define a metadata object called "cost center," it will look a bit strange if another object called "profit center," which differs a lot from a cost center in terms of business semantics, is then defined by linking it to the metadata object called "cost center."

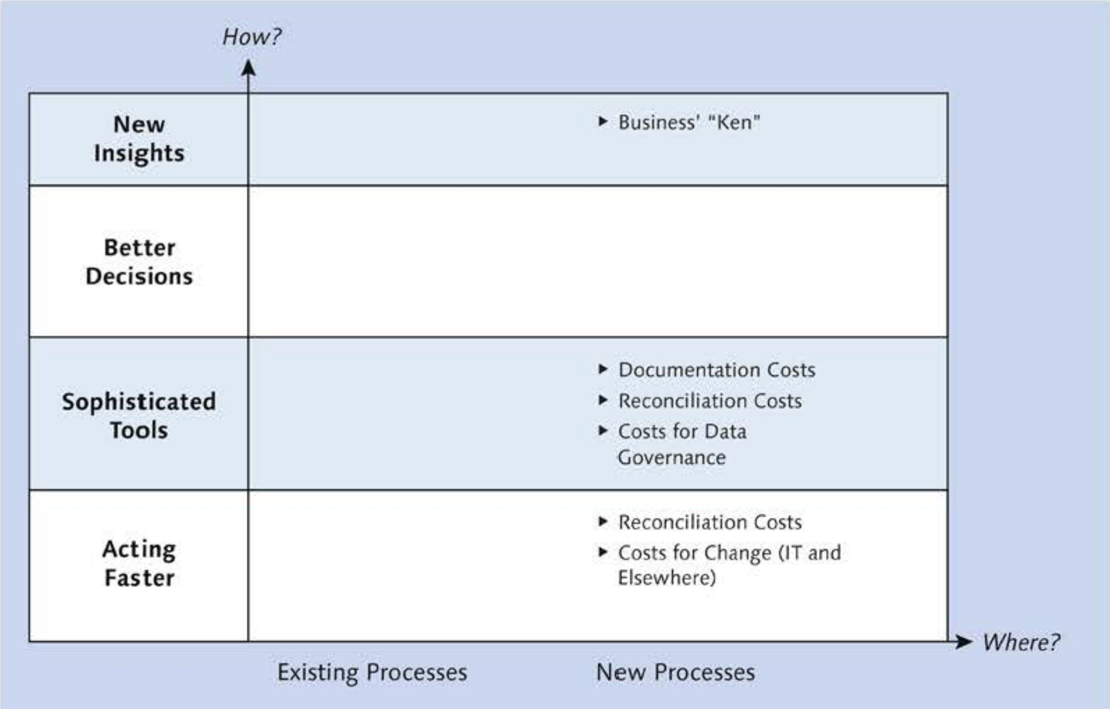
8.4.4 Potential Benefits and Value Drivers

Semantically neutral metadata and big data

In this case study, we once again need to clearly discriminate between the potential benefits of the new or existing business processes we are planning to support (for example, the additional revenues MN can generate by cooperating with the insurance company) and the additional benefits that the big data solution we are designing (extracting metadata automatically) might bring to the equation.

The idea of selling sensor data to third parties is case-specific and therefore may not be transferable to your environment even if you are working in the same industry. Our focus lies with the more generic part of this scenario—that is, with defining, extracting, and exchanging semantically neutral metadata. For this situation, there will be far more application scenarios in various industries and even with nonprofit organizations.

After talking to MN's partners, MN's CIO has run a couple of internal workshops and has identified the value drivers shown in Figure 8.5.



	Existing Processes	New Processes
New Insights		▸ Business' "Ken"
Better Decisions		
Sophisticated Tools		▸ Documentation Costs ▸ Reconciliation Costs ▸ Costs for Data Governance
Acting Faster		▸ Reconciliation Costs ▸ Costs for Change (IT and Elsewhere)

Figure 8.5 Sensor Data and Metadata Benefit–Value Driver Matrix

Value drivers for MN

The value drivers shown in Figure 8.5 might not be self-explanatory, so we'll say a few words about them:

- **Business' "ken"**

  First and foremost, MN is providing its partners with sensor data and semantically neutral metadata. MN's partners will use these metadata to analyze the delivered sensor data via complex mathematical–statistical algorithms, searching for interesting patterns and dependencies.

  Both the mobile phone provider and the insurance company are approaching the task unbiasedly, treating the data they received like the US Postal Service still (hopefully) treats our letters: they register the size of the envelope and the receiver written on it, deciding on that basis how to process the mail; meanwhile the contents of the envelope will remain untouched. Only if there are any peculiarities (if some white kind of powder is trickling from a letter addressed to the White House, for example) will they take a closer look. In the same way, MN's partners are only supposed to look at content-related metadata if and when they have detected a conspicuous pattern.

This naïve kind of approach, characteristic of big data, makes it possible to unveil relationships nobody might have thought of before. When it comes to shareholder value, it stands to reason that shareholders will be more interested in companies that are better than their competitors at looking beyond the ends of their noses.

- **Documentation costs**
  With the approach we are proposing, the effort that has to be invested in documenting technical objects such as sensors or IT systems is a lot lower than with classic Word documents or Excel tables, regardless of whether this information is to be used internally only or passed on to partners afterwards. MN's approach combines minimum effort with maximum quality and accessibility via transcription and multiple quality assurance; this will also help to keep the load off busy engineers who are supposed to invent new technologies rather than write tedious, lengthy documentation. Furthermore, if there is less effort involved in writing, finding, and understanding documents, then this should lead to lower personnel costs—and not only in IT.
- **Reconciliation costs**
  If documents are not clearly structured or—even more common—cannot be found any more, then users turn directly to the experts whenever they can. In MN's case, these would be the engineers, who would then be flooded by inquiries from the partners, having to answer the same questions over and over again. Although such direct contact can be stonewalled by setting up appropriate support structures, this will only lead to frustration for the other parties. Anybody who has ever used a hotline should be able to relate to this.

The approach chosen by MN combines the best from both worlds: the catchiness and clarity of a video with the ease of search of written text. MN's partners can use that information seven days a week, 24 hours a day. Everything is quality assured, easy to retrieve, and always up-to-date.

Easier documentation and reconciliation processes will reduce the amount of work needed to operate the new analytical solution. This is important because the calculation in Section 8.3.2 would no longer work if maintenance costs suddenly exploded.

- **Costs for data governance**
  If metadata objects only have to be created once and can then be used by various data, the metadata maintenance effort can be reduced by orders of magnitude. This affects both the initial creation of such metadata and their maintenance and updating. Furthermore, the more metadata objects you have, the more likely it is that one already exists for the new data that you're planning to add to your models.

  If your semantically neutral metadata are uniformly structured, then new metadata objects can automatically be checked for redundancies. If you are operating quite a few systems at a global scale, then you could also use periodical check runs to detect and/or eliminate such redundancies. Both effects together give you an edge over many of your competitors.

  Documenting metadata by structured interviews is another measure that helps to reduce the maintenance work, but because video documentation also needs a bit of reviewing and editing this has only a small effect on the value driver (the costs for data governance).

- **Costs for change (IT and elsewhere)**
  If the behavior of a sensor changes (for example, due to a new software version in your car's operating system), then you don't have to review and change umpteen potentially affected routines but only a single metadata object. Subsequently, you only need to check which data flows are affected (remember our thoughts about data lineage in Chapter 6, Section 6.1.1); in data governance such a check is called *impact analysis*. Then, if data flows have to be changed this can also happen (semi) automatically (see Chapter 6).

  The resulting benefits in terms of personnel costs are the same as described in Chapter 6, Section 6.4.4 for the development costs value driver.

As in Chapter 6, our value drivers are located in the right-hand column of the benefit–value driver matrix. This is because both the underlying business process (analyzing sensor data) and the one we are focusing on (extracting and exchanging semantically neutral metadata among partners or internally between business areas or associated companies) are relatively new for most organizations.

Value drivers and the underlying business process

Apart from these metadata-related value drivers, there are also some that are affected by the underlying business process. Although that process is not the main topic of this chapter, we would still like to mention the affected value drivers. On the one hand, they are the quick wins for MN and its partners; on the other hand these benefits could not be realized without the approaches explained in this chapter.

Such value drivers are as follows:

- New insights for the insurance company in terms of assessing customers and risks
- New lessons learned that will help improve algorithms in MN's driver assistance systems
- MN's revenues from data sold to partners
- MN's additional revenue from selling more cars

Value driver database (add-on)

The last two items in the above list are not really due to big data but to exploiting customer data by selling them with, and to linking insurance policies to, vehicle sales. Both have been done before and are not big data specific. If you need to find more big data-related value drivers, the value driver database that is provided as an add-on to this book and that can be downloaded from *www.sap-press.com/3647* might help you generate more ideas.

8.5 Implementation Scenario and Framework Architecture with SAP HANA

Focus: metadata

In this chapter, metadata are the center of attention, but SAP HANA is not a tool used only to file or manage metadata. At the end of Section 8.4.1, we mentioned other suitable products from SAP. Taking that into account, the case study here focuses predominantly on two questions:

- How can both content-specific and semantically neutral metadata be extracted from unstructured or loosely structured sources—for example, data from videos?
- What do semantically neutral metadata need to look like to make the respective metadata objects reusable?

8.5.1 Implementation Scenario and Framework Architecture

SAP Business Suite not involved

In our scenario, SAP Business Suite doesn't play a major role as a source of data. The same applies to other classic SAP products, such as SAP BW. Regardless of whether SAP software (as described in Section 8.4.1) is used for data acquisition or whether extracted metadata are later stored in SAP's metadata repositories, the integration with SAP's products for data generation is less important here.

For this reason, we are proposing the app scenario. The app scenario is not an integrated scenario but one of the replication scenarios. The data mart scenario—another replication scenario—would probably not be sufficient for our purposes, because it more or less relies on standard products for data exploitation and not on particularly sophisticated analytical scenarios.

Differences compared to Chapter 4

The app scenario has been used before, in Chapter 4 (see Figure 4.15). Its components in the case at hand, however, are very different from the ones suggested there. Based upon Figure 8.6, we will explain the differences in detail.

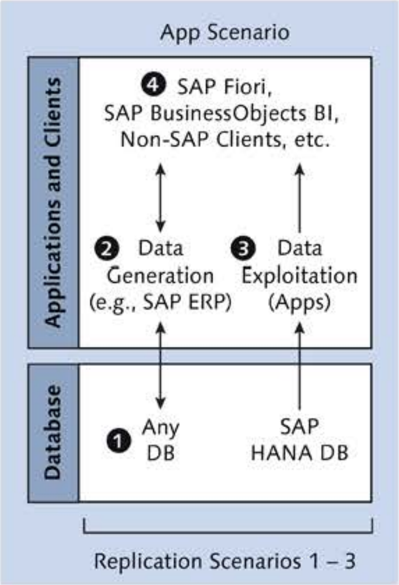

App Scenario
Applications and Clients
❹ SAP Fiori, SAP BusinessObjects BI, Non-SAP Clients, etc.
❷ Data Generation (e.g., SAP ERP)
❸ Data Exploitation (Apps)
Database
❶ Any DB
SAP HANA DB
Replication Scenarios 1 – 3

Figure 8.6 Sensor Data and Metadata Implementation Scenario

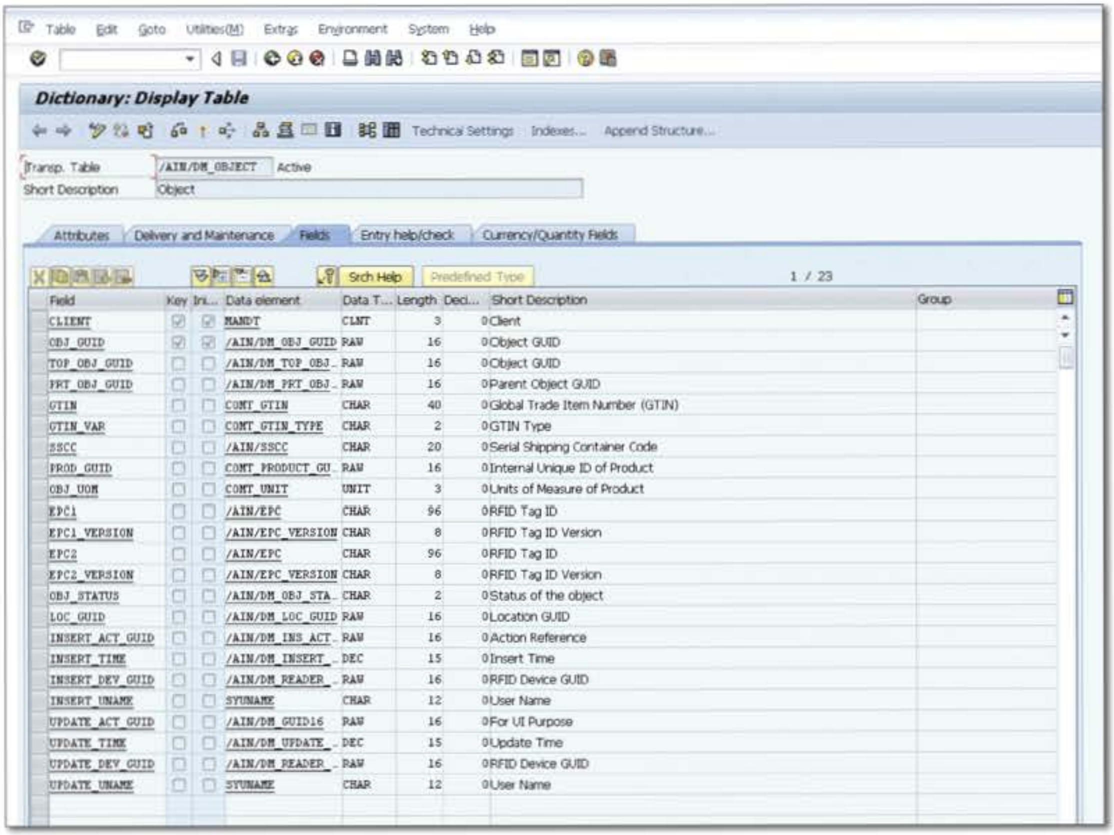
Databases ❶

Maneki-neko may want to use data from a variety of sources, such as RFID data from transponders, measurements from sensors, metadata or unstructured data.

RFID Data

Tables within SAP Business Suite

SAP AII files its data in plain vanilla relational database tables within SAP Business Suite. You can find these tables by searching for `/AIN/DM_OBJ*` in your ABAP Dictionary (Transaction SE11). Table `AIN/DM_OBJECT`, shown in Figure 8.7, plays a central role in this context. It collects the following objects and creates a link among them by putting them all into one data record.



Dictionary: Display Table

Transp. Table /AIN/DM_OBJECT Active
Short Description Object

Field	Key	Ini...	Data element	Data T...	Length	Deci...	Short Description	Group
CLIENT	☑	☑	MANDT	CLNT	3	0	Client	
OBJ_GUID	☑	☑	/AIN/DM_OBJ_GUID	RAW	16	0	Object GUID	
TOP_OBJ_GUID	□	□	/AIN/DM_TOP_OBJ..	RAW	16	0	Object GUID	
PRT_OBJ_GUID	□	□	/AIN/DM_PRT_OBJ..	RAW	16	0	Parent Object GUID	
GTIN	□	□	COMT_GTIN	CHAR	40	0	Global Trade Item Number (GTIN)	
GTIN_VAR	□	□	COMT_GTIN_TYPE	CHAR	2	0	GTIN Type	
SSCC	□	□	/AIN/SSCC	CHAR	20	0	Serial Shipping Container Code	
PROD_GUID	□	□	COMT_PRODUCT_GU..	RAW	16	0	Internal Unique ID of Product	
OBJ_UOM	□	□	COMT_UNIT	UNIT	3	0	Units of Measure of Product	
EPC1	□	□	/AIN/EPC	CHAR	96	0	RFID Tag ID	
EPC1_VERSION	□	□	/AIN/EPC_VERSION	CHAR	8	0	RFID Tag ID Version	
EPC2	□	□	/AIN/EPC	CHAR	96	0	RFID Tag ID	
EPC2_VERSION	□	□	/AIN/EPC_VERSION	CHAR	8	0	RFID Tag ID Version	
OBJ_STATUS	□	□	/AIN/DM_OBJ_STA..	CHAR	2	0	Status of the object	
LOC_GUID	□	□	/AIN/DM_LOC_GUID	RAW	16	0	Location GUID	
INSERT_ACT_GUID	□	□	/AIN/DM_INS_ACT..	RAW	16	0	Action Reference	
INSERT_TIME	□	□	/AIN/DM_INSERT_..	DEC	15	0	Insert Time	
INSERT_DEV_GUID	□	□	/AIN/DM_READER_..	RAW	16	0	RFID Device GUID	
INSERT_UNAME	□	□	SYUNAME	CHAR	12	0	User Name	
UPDATE_ACT_GUID	□	□	/AIN/DM_GUID16	RAW	16	0	For UI Purpose	
UPDATE_TIME	□	□	/AIN/DM_UPDATE_..	DEC	15	0	Update Time	
UPDATE_DEV_GUID	□	□	/AIN/DM_READER_..	RAW	16	0	RFID Device GUID	
UPDATE_UNAME	□	□	SYUNAME	CHAR	12	0	User Name	

Figure 8.7 Table AIN/DM_OBJECT

- An RFID transponder on which an electronic product code (EPC; the EPC1 field in Table AIN/DM_OBJECT) has been stored
- A product identification number or global trade item number (GTIN; the GTIN field in Table AIN/DM_OBJECT)—for example, an international article number, formerly called a European article number (EAN), that uniquely identifies a product
- A serial shipping container code (SSCC; the GTIN field in Table AIN/DM_OBJECT), sometimes called NVE (German acronym for *nummer der versandeinheit*) in SAP ERP, that represents something akin to a container's license plate
- A product ID (the PROD_GUID field in Table AIN/DM_OBJECT) that refers to a product's or an article's master data within SAP ERP

Table AIN/DM_OBJECT comes into operation if sensor data sent by RFID transponders have to be processed. In most cases, such data are nothing but location IDs that can be used to derive the respective processing steps within material flows of supply chains. Although MN uses RFID systems in supply chain management, such data are less important in the current scenario.

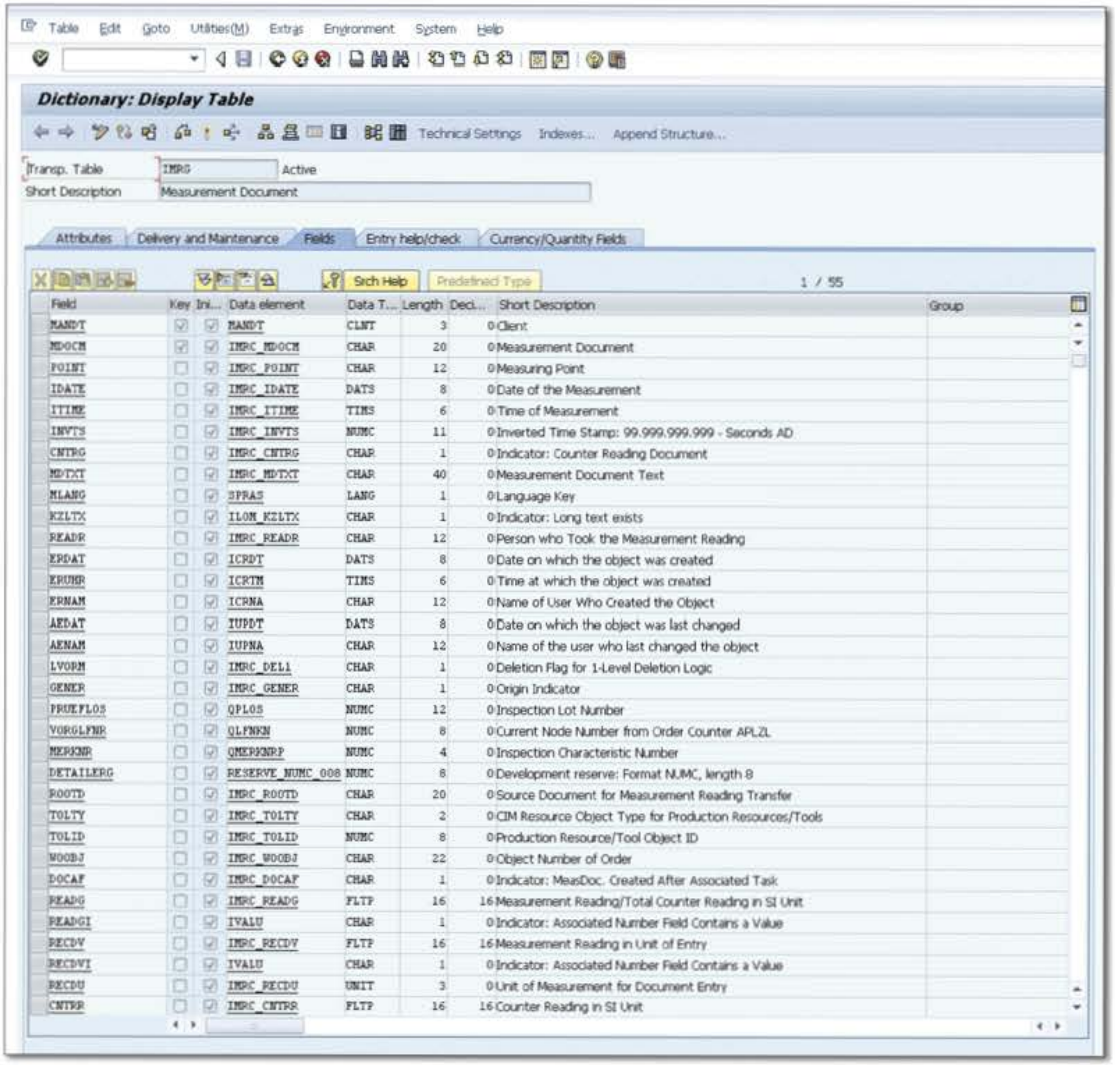
Measurement Data

Table IMRG

Sometimes sensors that capture measurement readings are modeled as measuring points within SAP ERP; in such a case—importantly, for our scenario—the data delivered by sensors are stored in measurement documents that contain the actual readings. Table IMRG (see Figure 8.8), which belongs to SAP ERP's plant maintenance (PM) module (formally known as *SAP Enterprise Asset Management*, or SAP EAM), contains such measurement documents. In PM, measurements are used to schedule preventative maintenance measures.

One of the most important master data objects within plant maintenance is the so-called equipment, representing technical systems, assets, or parts of these. On any equipment, there can be *n* measurement points (the POINT field in Table IMRG), and for each measurement point there can be an infinite number of measurement documents. Table IMRG contains some of the information we have discussed when defining sensor data–related metadata—for example:

- The time of measurement (the IDATE—Date of the Measurment, ITIME—Time of the Measurement, and INVTS—Inverted Time Stamp fields), which is different from the time the measurement document arrived in Table IMRG (the ERDAT and ERUHR fields)
- An indicator (the INTVL field) that indicates whether the measurement refers to a period of time or a point in time
- If applicable, the start of the time interval (the IDAT1 and ITIM1 fields; the end is not needed as long as intervals can never overlap and as long as the last measurement is not used)



Transp. Table: IMRG Active

Short Description: Measurement Document

Field	Key	Ini...	Data element	Data T...	Length	Deci...	Short Description	Group
MANDT	☑	☑	MANDT	CLNT	3	0	Client	
MDOCM	☑	☑	IMRC_MDOCM	CHAR	20	0	Measurement Document	
POINT	☐	☑	IMRC_POINT	CHAR	12	0	Measuring Point	
IDATE	☐	☑	IMRC_IDATE	DATS	8	0	Date of the Measurement	
ITIME	☐	☑	IMRC_ITIME	TIMS	6	0	Time of Measurement	
INVTS	☐	☑	IMRC_INVTS	NUMC	11	0	Inverted Time Stamp: 99.999.999.999 - Seconds AD	
CNTRG	☐	☑	IMRC_CNTRG	CHAR	1	0	Indicator: Counter Reading Document	
MDTXT	☐	☑	IMRC_MDTXT	CHAR	40	0	Measurement Document Text	
MLANG	☐	☑	SPRAS	LANG	1	0	Language Key	
KZLTX	☐	☑	ILOM_KZLTX	CHAR	1	0	Indicator: Long text exists	
READR	☐	☑	IMRC_READR	CHAR	12	0	Person who Took the Measurement Reading	
ERDAT	☐	☑	ICRDT	DATS	8	0	Date on which the object was created	
ERUHR	☐	☑	ICRTM	TIMS	6	0	Time at which the object was created	
ERNAM	☐	☑	ICRNA	CHAR	12	0	Name of User Who Created the Object	
AEDAT	☐	☑	IUPDT	DATS	8	0	Date on which the object was last changed	
AENAM	☐	☑	IUPNA	CHAR	12	0	Name of the user who last changed the object	
LVORM	☐	☑	IMRC_DEL1	CHAR	1	0	Deletion Flag for 1-Level Deletion Logic	
GENER	☐	☑	IMRC_GENER	CHAR	1	0	Origin Indicator	
PRUEFLOS	☐	☑	QPLOS	NUMC	12	0	Inspection Lot Number	
VORGLFNR	☐	☑	QLFNKN	NUMC	8	0	Current Node Number from Order Counter APLZL	
MERKNR	☐	☑	QMERKNRP	NUMC	4	0	Inspection Characteristic Number	
DETAILERG	☐	☑	RESERVE_NUMC_008	NUMC	8	0	Development reserve: Format NUMC, length 8	
ROOTD	☐	☑	IMRC_ROOTD	CHAR	20	0	Source Document for Measurement Reading Transfer	
TOLTY	☐	☑	IMRC_TOLTY	CHAR	2	0	CIM Resource Object Type for Production Resources/Tools	
TOLID	☐	☑	IMRC_TOLID	NUMC	8	0	Production Resource/Tool Object ID	
WOOBJ	☐	☑	IMRC_WOOBJ	CHAR	22	0	Object Number of Order	
DOCAF	☐	☑	IMRC_DOCAF	CHAR	1	0	Indicator: MeasDoc. Created After Associated Task	
READG	☐	☑	IMRC_READG	FLTP	16	16	Measurement Reading/Total Counter Reading in SI Unit	
READGI	☐	☑	IVALU	CHAR	1	0	Indicator: Associated Number Field Contains a Value	
RECDV	☐	☑	IMRC_RECDV	FLTP	16	16	Measurement Reading in Unit of Entry	
RECDVI	☐	☑	IVALU	CHAR	1	0	Indicator: Associated Number Field Contains a Value	
RECDU	☐	☑	IMRC_RECDU	UNIT	3	0	Unit of Measurement for Document Entry	
CNTRR	☐	☑	IMRC_CNTRR	FLTP	16	16	Counter Reading in SI Unit	

Figure 8.8 Table IMRG

Admittedly, we have learned that most companies do not store measurement data within Table IMRG but instead design their own structures within a data warehouse for maximum flexibility.

Accessing measurement data from SAP HANA

Regardless of where the data are stored, measurement data can originate from non-SAP or even proprietary legacy applications. But no matter where such data come from, they will probably either end up in a database or will be modeled as events, which means they are accessible to SAP HANA (via SAP ESP or SAP Data Services) either way.

Metadata

Other databases

Products for data management (such as SAP Data Services or SAP Information Steward) also store their data—that is, the metadata—in regular databases (MS SQL, Oracle, and so on). These metadata can be exchanged via the Meta Integration Model Bridge (MIMB) mentioned under "Other Products (Databases, Platforms, Technology, and Services)" in Chapter 2, Section 2.1.3. SAP Information Steward also uses MIMB to extract metadata from SAP and non-SAP sources.

Unstructured Data

When dealing with weakly structured or unstructured data, as is the case with video recordings, one can use Apache Hadoop to store them; SAP Data Services can then wrestle the data from there and pass it on to SAP HANA.

Products for Data Generation ❷

Acquire/manage sensor data/metadata

In our scenario, the products for data generation are not just ERP applications but instead all solutions that handle acquiring and managing sensor data and metadata, as listed in Section 8.4.1. For obvious reasons, the SAP Solution Explorer only shows SAP solutions, but there are numerous other products with more or less identical, similar, or even superior functionalities. From the perspective of our SAP HANA app, it doesn't matter in the slightest where sensor data or metadata come from or what exactly sensor data are measuring.

Products for Data Exploitation ❸

Central duties of our solution

Data exploitation is the central responsibility of the application we are trying to design here. Within our implementation scenario, products for data exploitation (both standard software and our proprietary app) are required to do four different jobs:

- **Acquire and manage content-specific metadata**
  "Acquiring" in this context means extracting metadata from a variety of sources; such sources could be the videos mentioned previously, documents residing on the intranet or the Internet, or even recorded video conferences. Products that could be used here include the following: storage solutions to organize video data, speech recognition software, text-mining functions (such as creating full-text indexes via the `EXTRACTION_CORE_VOICEOFCUSTOMER` function available in SAP HANA from SPS 6 Revision 60 or—going even further—via products like SAP PAL that support cluster analysis), products for time code–based video tagging, metadata repositories to store collected metadata, and search engines or appliances like GSA that help you to really tap into the value provided by your metadata.
- **Acquire and manage semantically neutral metadata**
  Apart from content-specific metadata, MN first and foremost needs semantically neutral metadata. These data can be obtained automatically (via speech recognition and text mining) or semiautomatically (via workflow solutions). Flexible, configurable workflow solutions will be a great help here. These workflows can be triggered by activities within SAP Business Suite—for example, by transactions that change the material master data or a draw for a sensor. In SAP's product portfolio, these solutions can be found by searching for "SAP process orchestration". For semantically neutral metadata, you will also need filing cabinets (metadata repositories) and a good librarian (search algorithm or appliance) to help you retrieve the ones you need.
- **Connect data and semantically neutral metadata**
  The applications used by MN's partners need to know how they can process or analyze the data delivered by the sensors. Therefore, sensor data and semantically neutral metadata need to be linked. To make that link as flexible as possible, it could be created in a relatively simple table in SAP HANA that is maintained via a user-friendly frontend application. This could contain the following fields: ID of the data source (a database or an event stream), ID of the object within the data source, and ID of the semantically neutral metadata object assigned to the object and defined in MN's metadata repository.

- **Connect semantically neutral metadata and content-specific metadata**
  We have suggested that MN's partners should only look at the meaning of data once they have discovered peculiarities, conspicuous patterns, or potential dependencies. When they do, they will need fast and easy access to content-related metadata to help them interpret their findings. This means they will want to see the video recordings as a whole or the parts of them that may be related to their discoveries.

Clients ❹

Visualize and pass on metadata

Ultimately, our application's job is to pass on metadata to MN's partners. As mentioned in Section 8.2, the sensor data are not to be sent to partners by MN but directly transmitted from the vehicles to the mobile phone provider and from there to the insurance company. Consequently, we don't need clients that can forward or visualize sensor data. Metadata, however, are handled by MN, so at a client level they are looking for metadata-related functionalities:

- **Clients to visualize metadata**
  One of the most important jobs of a metadata repository is to create transparency and to define and clarify relationships. As a consequence, almost each and every metadata-management solution comes with a user interface that is able to list metadata or graphically show links between them. With SAP Information Steward, relational metadata (such as information about data lineage) can be visualized in lineage and impact diagrams. In addition, there are a number of useful dashboards—for example, dashboards depicting data quality.
- **Clients to pass on metadata**
  One of the most important quality criteria when selecting a metadata repository is its openness—that is, the ability to import metadata from a wide variety of sources and to export them to all kinds of recipients.

  Most metadata exchange formats are XML based; SAP Information Steward uses CWM-XLM, the most common cross-system metadata format. CWM stands for *Common Warehouse Metamodel*, a standard developed by the Open Management Group (OMG) and primarily used for exchanging metadata between data warehouses.

We can therefore select our client(s) from quite a few existing, standard products for data management; there is no need to start from scratch.

8.5.2 Data Architecture

Looking at content-specific metadata

When describing the suggested data architecture for this chapter, we will first focus on acquiring/extracting content-specific metadata. There are three reasons for this:

- In terms of extracting or managing metadata, there are no major differences between content-related and semantically neutral metadata; hence most of what we are saying about content-related metadata can also be applied to semantically neutral metadata.
- When using semantically neutral metadata, remember that MN will not analyze sensor data itself. The partners are going to take care of this. Therefore, using semantically neutral metadata to properly process sensor data is not really in scope from MN's perspective.
- Chapter 9 also addresses sensor data. There we will take a closer look at how metadata can be analyzed, focusing on the client level and alerting functionalities.

Extracting content-specific metadata

Figure 8.9 shows a sample data model with possible data flows used to extract and file content-specific metadata from audio and video recordings.

Two domains/ groups of domains in data model

As in Chapter 6, we have split the architecture into two domains or domain groups, but in a different way:

- **Tasks executed by humans**
  Right now, human skills are indispensable when extracting and interpreting metadata from weakly structured/unstructured multimedia sources. In Figure 8.9, we put all the data that humans will have to create or work with on the left-hand side. Because humans cannot directly interact with systems, user interfaces are required for these five boxes. At the bottom, there is a (human) source of data (the interviews), and at the top you'll find a client used by humans (the metadata repository).

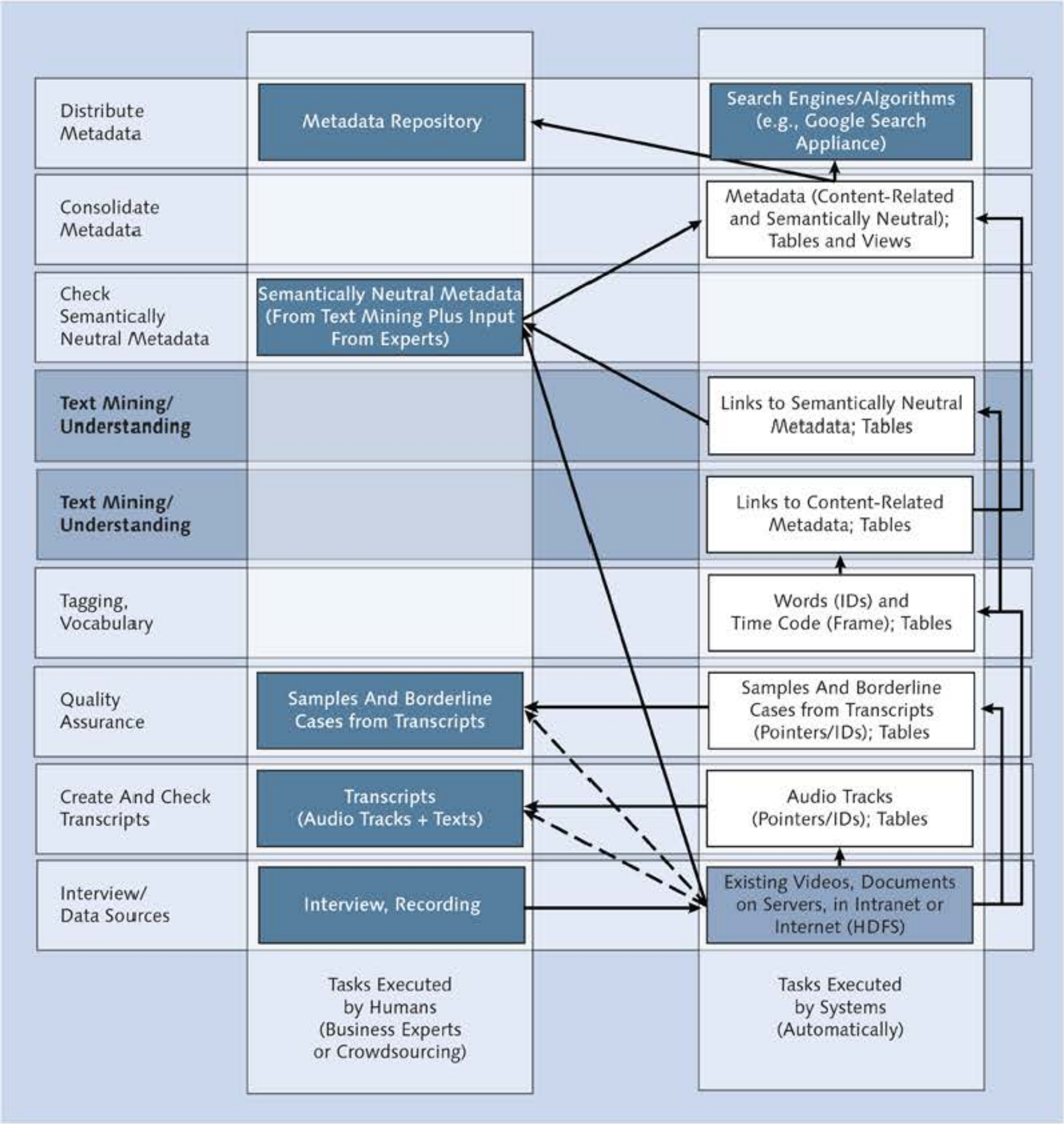

Distribute Metadata
Metadata Repository
Search Engines/Algorithms (e.g., Google Search Appliance)
Consolidate Metadata
Metadata (Content-Related and Semantically Neutral); Tables and Views
Check Semantically Neutral Metadata
Semantically Neutral Metadata (From Text Mining Plus Input From Experts)
Text Mining/ Understanding
Links to Semantically Neutral Metadata; Tables
Text Mining/ Understanding
Links to Content-Related Metadata; Tables
Tagging, Vocabulary
Words (IDs) and Time Code (Frame); Tables
Quality Assurance
Samples And Borderline Cases from Transcripts
Samples And Borderline Cases from Transcripts (Pointers/IDs); Tables
Create And Check Transcripts
Transcripts (Audio Tracks + Texts)
Audio Tracks (Pointers/IDs); Tables
Interview/ Data Sources
Interview, Recording
Existing Videos, Documents on Servers, in Intranet or Internet (HDFS)
Tasks Executed by Humans (Business Experts or Crowdsourcing)
Tasks Executed by Systems (Automatically)

Figure 8.9 Sensor Data and Metadata Data Architecture

- **Tasks executed by systems**
  Humans and systems interact when processing metadata, each receiving input from the other party and handing results back to it. All the data that SAP HANA will have to process have been put on the

right-hand side of Figure 8.9. There is also a data source (existing files) that may not need further human input and another client at the top (a search engine) that is for the humans to use.

Layers in data model

Our data model has nine layers:

- Layer 1: Interview/data sources
- Layer 2: Create and check transcripts
- Layer 3: Quality assurance
- Layer 4: Tagging and vocabulary
- Layers 5/6: Text mining and understanding (two layers)
- Layer 7: Check semantically neutral metadata
- Layer 8: Consolidate metadata
- Layer 9: Distribute metadata

Both one-way and bidirectional data flows

In the following sections, we are going to describe the individual processing steps that sit between these layers. With some of these processing steps or data flows, data can flow in both directions (the data flows are bidirectional); we have used dashed lines to indicate these.

The upper TEXT MINING layer and the CHECK SEMANTICALLY NEUTRAL METADATA layer deal mainly with semantically neutral metadata. All other layers would be the same for both metadata types (as mentioned in the introduction to this section).

Interview/Data Sources

Conduct/record interview

The first step is to conduct the interviews described in Section 8.4.3, recording them with a suitable solution (such as Adobe OnLocation, Microsoft Lync, etc.).

Together with other, existing sources of information, such as documents on MN's servers and its intranet and information downloaded from the Internet or from other multimedia sources, the interviews are stored; due to expected data volumes and structure (or lack thereof), Hadoop Distributed File System (HDFS) could be used for this.

Create and Check Transcript

Available audio information, such as the audio tracks of video recordings, recorded conference calls, and the like, are transcribed using speech-recognition/text-analysis software. If the software has been trained before via vocabularies of MN's business and technical terms, it will perform much better when recognizing text.

Security with crowdsourcing

The transcripts and the related audio files are then split into small fragments; each of these fragments will have its own ID and will be made available to a crowdsourcing community via an extranet or a virtual private network (VPN). For security reasons, people working on this project will have to sign nondisclosure agreements; furthermore, each individual will only work on small, unrelated fragments.

Purpose of crowdsourcing

The crowdsourcing community will then check the transcripts, making sure that spoken and written texts are 100% identical, and any corrections needed are stored together with the affected transcripts. Quality can be improved dramatically if one and the same transcript is checked more than once by different reviewers; statistical algorithms can help you work out how often a certain fragment needs to be presented to reach a predefined maximum number of differences between the spoken text and the transcript.

Quality Assurance

More sampling

Samples are used to check the quality of the output that results from the previous step. Our application can automatically identify the borderline cases—that is, all fragments for which different people have come up with different results. These should go into additional review rounds in crowdsourcing or should be reviewed internally. On top of that, a sample of all borderline fragments can be extracted and cross-checked (just to be on the safe side).

For sample sizing and sampling, you can once again use the algorithms discussed in Chapter 7, Section 7.4.3 and Section 7.5.1. As in Chapter 7, these data do not have to be replicated; all SAP HANA will have to do is generate a pointer or an ID for selected data sets. This pointer or ID can then be used to hand over data to the extranet for review by the crowdsourcing community.

Tagging and Vocabulary

Tag audio and video recordings

Based upon the quality-assured transcripts, the words within the written texts are connected to the time codes of their respective audio and video files. For audio files, one can use the time at which a word has been spoken (if necessary, locating the time to as close as one hundredth of a second); for video recordings, frames are a better option.

Create vocabulary

The transcripts can also be used to create a vocabulary via SAP HANA. Once created, such a vocabulary should also be cleaned via a stop vocabulary. A *stop vocabulary* is a vocabulary of filler words and phrases, such as "and," "the," "sort of," and so on; such words carry very little information. They should not be removed but should be marked for further analysis. Some text-mining algorithms can use that kind of information.

Tagging other files

Tagging should not only be used for audio and video files but also for all other kinds of information available on your Hadoop/HDFS cluster. A user looking for a certain word later not only wants to find audio and video files but also all other kinds of documents that contain his search terms or words with similar or related meanings. GSA, the search appliance used by MN, also comes with tagging functions.

At this point, there isn't much more we can do in terms of extracting metadata. We now have excellent transcripts of audio and video files, and we are able to use sophisticated search algorithms on both of them and on all other types of documents.

Text Mining and Understanding

Understanding texts

As well as searching for individual words, MN would also like to be able to search for meaning. As mentioned under "Content-Based Metadata (Not Semantically Neutral)" in Section 8.4.3, when entering "proximity sensor" as a search term MN wants to be able to find all documents containing the term "Reed sensor." Therefore, we are not yet out of the woods; MN would like to go a few steps further in terms of systems understanding the meaning of texts.

In Section 8.3.1, we explained that taxonomies are one way to get a bit closer to this vision. A Reed sensor is one special type of proximity sensor, but there are quite a few others. If all of them were assigned to a node called "proximity sensor" within a hierarchy, then a search like the one mentioned previously would find all related documents when entering the superordinate term (proximity sensor).

Association/ tagging instead of taxonomies

In practice, however, taxonomies are often unwieldy, inflexible, and costly to maintain, and assigning a term to a node is often an expression of subjective judgment rather than an objective one. This is why a lot of applications and websites featuring content-related search—such as image databases that sell photographs—choose to use tagging and associations rather than hierarchical structures. If the words "Reed" and "sensor" often occur together with "proximity" or in text segments titled "proximity sensors," then a system may just start to link these two terms using Bayesian inference (see Chapter 10, Section 10.4.3 for a full definition).

Going after the real thing: the brain's algorithms

To really tap into the value of data, there is no way around understanding text. Google's algorithms are not there yet, and text mining or toolkits such as NLTK are only a starting point. IBM's Watson has performed exceptionally well at the quiz show *Jeopardy!;* nevertheless, all of these stage wins only utilize big data's ability to crunch huge volumes of data in a short time. Imagine if that kind of speed was combined with the built-in reasoning algorithms of the human brain.

Check Semantically Neutral Metadata

Quite a few things that look like understanding from the outside are simply based upon spotting correlations. We are not saying that this tactic doesn't make sense: in fact, quite the opposite. When it comes to translation, software built on the basis of linguistic rules never got very far; in the meantime, Google's translation tools are performing a lot better by ignoring linguistics and instead looking for corresponding terms on websites that are available in different languages.

Cross-check by experts

Regardless of how advanced your text mining tools are, you will not be able to avoid a final check of the extracted semantically neutral metadata by your experts. You can, however, make them happy by providing them with a user-friendly interface, showing them all the information you have collected so far, and allowing them to correct it if necessary. One such user interface is the Metapedia within SAP Information

Steward. Alternatively, you could export the metadata stored there, making them available via your choice of web-based client, or simply give your experts or your partners' experts access to the raw data within your SAP HANA database before these data are handed over to the metadata repository.

In our model, we chose the latter alternative. The results from text mining are placed at the disposal of engineers and are revised by them before getting passed on to metadata repositories or being distributed elsewhere.

No expert check for content-related metadata

One could also think about a similar cross-check for content-related metadata, but we don't believe that this would be worthwhile. Errors are less likely there (people will notice if their searches and the videos returned don't match up), and the damage these errors would cause is probably minimal. Also, don't forget that content-specific metadata change frequently.

Consolidate Metadata

Clean up metadata

For traceability reasons, you should keep all the information that went into the preceding process; this includes the input from your experts as well as the draft metadata you provided them with. If you do that, however, you will need two additional consolidation steps:

- First, the text mining output will have to be cleaned and corrected based on whatever your experts told you before it was passed on to the next processing step.
- Second, now is the time to link the content-related metadata that have been quality assured in the respective layer (quality assurance) and your semantically neutral metadata. The connecting elements among them are your metadata objects; remember that each value or fact delivered by one of MN's sensors is assigned to such a metadata object.

Distribute Metadata

Handover to metadata repository/search engine

Now that you have the final version of your metadata, you need to hand them over to your metadata repository and make them available or accessible to your search engine (GSA in MN's case). You can use the standard interfaces discussed previously to hand over metadata to whatever kind of metadata repository you may have implemented; GSA explores metadata by using crawlers (see Chapter 4, Section 4.5.2).

8.5.3 Applying the Concept to Other Case Studies

In this chapter, we looked at two problems that can be transferred to a whole range of other industries and cases. We explained how highly structured information, such as semantically neutral metadata, can be extracted from weakly structured or unstructured data, such as audio or video recordings, and also talked about analyzing sensor data as one application area for highly flexible analytical solutions founded on semantically neutral metadata. We have also suggested a data model in which content-related metadata and semantically neutral metadata are clearly separated from each other and at the same time dynamically linked to the respective data.

Other industries and scenarios

For these concepts, we see quite a few potential fields of application:

- How about analyzing videos, audio files, or images that your customers post on Facebook or YouTube in the same way that you analyzing Twitter feeds at the moment? Like tweets, these videos or photographs are posted at irregular intervals, which means that your data source is an event stream, but as explained in Section 8.4.1 and Section 8.5.1 this could also be the case with sensor data.
- In major cities, we are surrounded by cameras. Many of these cameras supply images that are publicly available, and some of these images could be useful for your organization. A weather cam might provide you with information about an approaching rain front. If you are a main street retailer, this will certainly have an impact on the number of visitors strolling around your shops to keep dry. Maybe it will provide a good opportunity for a gentle price increase. Maybe a traffic cam could show you how many trucks are leaving your main competitor's central warehouse each day, providing you with a way to estimate its revenues.

  In these examples, we are talking about analyzing images rather than texts, but once the key information within these images has been identified (such as "heavy rain to be expected in the next 30 minutes") and converted into text the subsequent processing steps are more or less the same.

- The term *sensor data* can be expanded a bit. As well as sensing elements in cars or machines, there are also sensors for body functions (see Chapter 9) or sensors that collect data within an IT service organization (see Chapter 11). As we discussed in this chapter, in these cases it also makes sense to separate content-specific and semantically neutral metadata and to link both kinds of metadata dynamically with the data they refer to.

We are sure that these generalizations and examples will spark a few ideas and thoughts for you and your organization.

If you don't invest a little bit of time and effort in your health every day, you may one day have to spend a lot of time on your disease.

Sebastian Kneipp, hydrotherapist

9 Health Management as a Service

No network; Derek put his mobile phone away again. In the course of the last three hours, he had taken the thing out of his breast pocket dozens of times. Not even one measly bar was registering on its display.

A typical tussock-strewn terrain lay before him. Mountain streams were meandering in wide turns through an extended gravel plain that was bordered by steep green hills on either side. Behind those hills were craggy rocks, and behind those rocks even higher slopes, on which the first snow of the year had already fallen. No road or bridge, no house, no sign of human civilization for miles around.

Figure 9.1 Landscape in Arthur's Pass National Park, Canterbury, New Zealand

This was the South Island, and Derek always opted for peace and solitude on his trips and hikes, but since his favorite Aunt, Marian, had passed away, *being in such a deserted place made him nervous. A few months ago, Marian had died from a stroke after she had lain helpless on the floor of her flat for what must have been many hours.*

Marian had been living in a retirement village for seniors; she sometimes suffered from bouts of dizziness, so she was grateful for the emergency buttons in each room. Derek had also bought her a mobile phone especially designed for seniors. She only had to press a big red button on the phone to be immediately connected to a 24/7 answering service; once connected, the caller could ask for help, and if the caller didn't say anything then a paramedic would be sent round immediately.

Unfortunately, both the emergency button on the wall and the telephone must have been out of reach when Marian had fallen, and even if the phone had been attached to her belt (as it was meant to be), she might not have been able to press the button.

There are systems that consist of a mobile phone and a wristwatch with a fall detector and there are floor coverings that can detect a fall, but even these would have been useless had Marian first become a bit dizzy, sat down, and then lost consciousness sitting on her sofa.

Sensors in medicine

Sensors are not only components of vehicles and machines; they also play an important role in medical devices. Sensing elements in a patient monitor in a hospital measure values continuously; others measure on request: when analyzing blood, for example, and when capturing vital signs to be interpreted by medical staff.

Big data enables new medical applications

Thanks to the miniaturization of electronic components, mobile communications, the analytical capabilities of big data, and intelligent scoring algorithms, new, predictive applications are being developed. There are some similarities between the impending failure of a plane's engine and an imminent heart attack: both are dramatic events for which there are known warning signs, and for both their probability within a certain future period of time can be estimated on the basis of specific parameters. Furthermore, a real-time alert is useful in both cases.

Analyzing sensor data

In this chapter, we will take another look at sensor data. This time, however, our focus will not be on how to define, extract, or manage related metadata but on triggering alerts on the basis of these sensor data. In this context, and taking into account some regulatory restrictions, we are going to evaluate the current state of this technology. Remember that the question of how to build the rules that lead to alerts was addressed in Chapter 4.

A new kind of service

To illustrate our thoughts, we will use a scenario in which a new kind of service based upon medical sensor data and big data applications will be offered. For this offering, we are going to identify its specific costs, risks, and opportunities. This time, we are not going to develop a detailed numerical example; instead, we will create a generic business model.

Implementing the scenario using SAP's products

We will then review if and how such an application could be implemented using SAP's tools and products, and specifically SAP HANA. While we have focused on the analytical capabilities of SAP HANA, SAP PAL, and R in previous chapters, we are now going to have a closer look at how insights gained via an analytical or predictive application can be revealed to the world outside via alerts and/or messaging services. SAP HANA's openness and its service-oriented architecture help facilitate this.

To do this, we will once again first define functional requirements independently of a specific solution and then compare them to SAP's offerings. Finally, we will close the chapter by comparing implementation scenarios and defining a data architecture. Our implementation scenario will be similar to the one in Chapter 8; the data architecture will focus on real-time alerting solutions that sit on five different client layers.

9.1 Medical Sensor Data

Invasive and noninvasive systems

Sensors used in medicine can be divided into two main categories: invasive and noninvasive. For invasive sensors, a sensing element will be implanted in the patient's body, meaning that tissue (such as skin) will have to be cut or penetrated. Noninvasive sensors measure things from outside the body. An electrocardiogram (ECG) is taken by noninvasive measurement, and a loop recorder is its invasive alternative.

Mobile patient monitors

The rapid progress in the miniaturization of electronic components has made it possible to use both invasive and noninvasive systems on the move, eliminating the need to visit the hospital for routine testing. Today, both types of sensors can be used at home (think of blood pressure monitors—noninvasive—or blood glucose measuring—mostly invasive) or even worn permanently (such as an ECG Holter monitor or a 24-hour blood pressure Holter monitor).

9.1.1 Invasive and Noninvasive Sensors

Noninvasive body sensors

Quite a few body sensors that are not considered medical devices are freely available. They largely fall into two categories:

- **Wristwatch sensors**
  Wristwatch sensors (for example, the Apple Watch) can do things like record exercise activities, the quality of your sleep, heart rates, and more. Many of these devices can transfer their data to a web server via the Internet, either in batch mode or online via mobile phones/data connections.
- **Body composition monitors**
  Body composition monitors can determine not only your weight but also many other parameters, such as body fat weight, muscle mass, and metabolic age. Some of them are also able to immediately transfer the results to a web server using your Wi-Fi connection at home.

Invasive medical sensors

Although some of the makers of these devices serve both the home and the professional medical market, there is one big difference between these distribution channels: in most countries, medical devices need some kind of approval or certificate.

Invasive systems for medical use

Although noninvasive measurement systems are often available for both home and professional use, most invasive sensors are only used by medical experts. The reason is obvious; we're sure you haven't toyed with the idea of cutting your skin open with a scalpel and popping a sensor in for a quick test. Nevertheless, there are some invasive sensors suitable for home use, for example, subcutaneous sensors (self-applied under the skin to monitor blood glucose levels), loop recorders (implanted surgically, but then used at home to monitor electrical activity of the heart), and miniature sensors (implanted surgically, but then used at home to monitor blood values).

9.1.2 Options for Data Transfer

Communication channels

Usually, noninvasive systems don't transfer their data to web servers directly. Often, these data are first put on a device supplied by the manufacturer of the sensing unit, typically a computer or a smartphone. The transfer itself happens via a cable (Ethernet, USB, or vendor-specific variants) or wirelessly (Wi-Fi, Bluetooth, or again vendor-specific solutions). For invasive systems, cables are usually not an option for practical reasons, which is why they usually either have to buffer data until they are removed or transfer data wirelessly.

Once data have been made available on another system, such as a laptop, they can then be transferred to a centralized facility—to the family doctor, for example, or to a fitness or health portal. This transfer can also take place via mobile data connections. Usually, only very small amounts of text data have to be sent, which is why even relatively slow connections (like GPRS instead of 4G/LTE) will do.

Continuous versus periodic transmission

Data transfer to the final recipient can either take place in real time or periodically (for example, every morning when stepping on the scale or every three months during a consultation with your family doctor). Whether, how, and how often data are to be transmitted to a database or a product for data exploitation depends on the specific objective. When trying to identify long-term trends with regards to heart rhythm problems, checking the data in longer time intervals and only sending results to your cardiologist's computer will do. An early warning system meant to detect an impending heart attack or the wearer's inability to move should, if at all possible, transfer its data in real time, perhaps to more than one recipient—for example, maybe to both the patient's doctor and his or her next of kin.

In this chapter, we will focus on real-time data transfer. Although some of the data we are going to talk about don't have to be analyzed in real time, we still assume—for the sake of simplicity—that everything is transferred as soon as a network connection becomes available; this may not mean that data are transferred immediately, but at least as fast as possible.

9.1.3 Specific Challenges with Medical Applications

Hardly any integrated solutions

Unlike the solutions related to cars or machines described in Chapter 8, the development of integrated medical systems with open and standardized interfaces is still in its infancy. The fact that this industry—despite the great opportunities discussed in this chapter—seems to be lagging behind is due to three factors:

- No standardized interfaces
- Approval and liability
- Costs of false positives and false negatives

No Standardized Interfaces

Vendor-specific interfaces

As many of these markets are new, every producer is doing their own thing. One driver behind that move might be the intention to seal off one's own market against the competition: one blood pressure monitor from Beurer transfers its measurements—unlike a solution from Withings—not to a smartphone but via a proprietary transmission protocol to a small box that has to be connected to an Internet router. The data are then sent via Internet to a portal from which they can be accessed using a browser or an application called HealthManager, again developed by Beurer. You can also download the data manually as a CSV file from that portal.

The blood pressure monitors from Withings are connected to a smartphone by cable or are coupled with one via Bluetooth. Measurements can then be viewed or analyzed using an app called Health Mate, which—apart from having a similar name and serving exactly the same purpose—has nothing in common with HealthManager by Beurer. Another company, Tanita, uses yet another variant; they partner with Garmin and therefore transfer the data of their body composition monitors to Garmin's wearable pulse monitors, from which they are transferred to a fitness portal called Garmin Connect.

Such bulk-heading tendencies can be seen in most new markets. Sooner or later, an open solution like XML or one vendor's variant (like Microsoft Office) establishes itself as the de facto standard (remember Betamax and VHS?).

Approval and Liability

Legal rules and regulations

In the United States, medical products have to be approved by the US Food and Drug Administration, whereas in Germany they are subject to the *Medizinproduktegesetz* (MPG; Medical Products Act), which represents the national implementation of three EU directives. The regulations in Austria are based upon the very same directives, and Switzerland has harmonized its national regulations with European law in 2002 via the so-called *Medizinprodukteverordnung* (MepV; Medical Products Regulation). This is not a recipe for common worldwide standards.

Complex approval processes

The definition of a medical product is fairly broad. It not only embraces scalpels and pacemakers but also products that are used to monitor body functions. Medical products work in a physical or physicochemical way and are separate from pharmaceutical products that have a pharmacological, metabolic, or immunological effect. Once a device or a solution is considered a medical product, it will be subject to highly complex tests and approval procedures. This is certainly in the best interests of patients, but it does hamper new developments. Inventors or small makers of smart solutions are thwarted because they are unable to invest the effort required.

We do not plan to address judicial questions in this book, but want to emphasize the fact that some of the solutions discussed in this chapter, be they hardware or software, might be subject to such regulations.

Costs of False Positives and False Negatives

False positives/ negatives: serious consequences

In Chapter 7, Section 7.4.3, we defined false positives and false negatives. Such errors affect all *predictive systems* (that is, systems generating predictions/forecasts), but for medical systems, the consequences of mistakes can be more serious.

[Ex]

False Positives and False Negatives

Let's say we have a system whose purpose is to warn us if a heart attack is looming within the next 24 hours. In this case, our null hypothesis would be "there will be no heart attack within the next 24 hours"; rejecting the null hypothesis would mean saying that a heart attack was to be expected.

If that heart attack does not occur, this would be a false positive. In contrast, an unforecasted heart attack would be a false negative—that is, the null hypothesis had not been rejected although it should have been. Whether a false positive or a false negative would have the more momentous consequences clearly depends on the null hypothesis.

In the case of this example, the costs of false positives and false negatives could be evaluated as follows:

- With a false positive, an ambulance would be mobilized, and, if necessary, the patient's front door would be forced open, then the affected individual would be taken to the nearest emergency department. A huge number of false positives would probably lead to a collapse of the medical infrastructure.
- The costs of just one single false negative are hardly quantifiable, as this would be—in the true sense of the phrase—a question of life and death. A US-based vendor whose systems repeatedly failed to detect a medical crisis in time should probably review their insurance policies and cash assets as soon as possible and brace themselves for the flood of compensation claims.

9.2 Scenario: Premium Services for the Elderly

A company operating special-care and nursing homes

Venerable Villas (VV) runs special-care and nursing homes in central Europe and at some popular holiday destinations like Majorca, Gran Canaria, and Ko Samui. The houses of VV are categorized by service offering, and by the purchasing power of the residents, into four categories: tin, silver, gold and platinum. At the moment, only four homes are considered platinum; three of them are located on Majorca, and the fourth one is on Ko Samui.

Searching for new sources of revenue

VV generates annual sales of 300 million EUR but has been in the red for two years now, which is why a couple of optimization initiatives were triggered quite a while ago. Apart from measures for cost reduction, such as reducing the number of nurses and spending less on food, the management has also been thinking about ways to increase revenues.

In the tin and silver categories, the placement costs for residents are often covered by social welfare offices, insurance companies, or relatives. VV considers the potential for additional sales there limited, at least at first glance. The management's eye has therefore fallen on the platinum houses; the residents there are usually worth more than 1 million EUR, so trying to sell them extra services looks promising.

Pilot project: sensing devices

Currently, VV is working on a pilot project for its houses on Majorca. Seniors living there will be offered a contract to guarantee them a fast response in case of emergencies and medical support for maintaining or improving their physical and psychological health in return for a certain annual payment. As a part of the agreement, they will be provided with a special mobile phone, a tablet, and some peripheral devices.

The mobile phone has been set up to receive and forward data from different sensors; the tablet will be used for regular interviews and concentration training. The peripheral devices include a type of bracelet and a sensor mat to monitor vital signs and sleep quality, a blood pressure meter, and a heart rate sensor. For certain groups of people, such as diabetics, or for more life threatening cases, implantable, invasive sensors are available as well; such sensors will mean extra costs for the patients, and their implementation will have to be cleared and prescribed by the residents' doctors.

For fast response in the case of an emergency, VV would like to cooperate with a German emergency hotline that runs a 24-hour call center near Frankfurt airport, a network of German doctors based on Majorca, and a Majorcan private clinic. The data collected in terms of the seniors' physical and mental fitness are to be analyzed by more cost-efficient medical doctors in Eastern European countries.

From VV's point of view, this basic package can be extended step-by-step while charging ever-increasing prices. They are currently thinking about additional sensors but also more personal supervision services, such as regular video chats with German-based doctors via tablet.

9.3 Monitoring Health: Costs, Risks, and Opportunities

All of that sounds great and promising from the board's perspective. There are, however, a couple of potential issues that need to be taken into account.

9.3.1 Problem: Legal and Financial Risks

Concerns of the legal department

When discussing the new offering in VV's inner management circle, it turns out that both the legal and the finance departments at VV have valid and serious concerns about the proposal that was originally developed by the head of marketing. The legal department has raised the question of the approval process for the algorithms to be used in a couple of countries. Because the new process would be distributed across more than one legislation, different legal systems would be affected. Furthermore, one would have to ask whether all newly developed algorithms would have to go through a lengthy and costly approval process. VV's company lawyer has been instructed to examine these questions.

Concerns of the finance and IT departments

The CFO's concerns are just as grave. He points out that false positives or false negatives could lead to enormous costs for VV. In the case of compensation claims, such costs could be far beyond all potential extra revenues. On top of that, the CIO has pointed out that his department would not be able to live up to these expectations (that is, developing sound scoring approaches and algorithms), both in terms of headcount and knowledge. He is estimating that his team would have to be augmented by about 20 highly qualified full-time equivalents with medical and/or statistical backgrounds.

9.3.2 Problem: Algorithms Challenging to Develop

Decision trees and machine learning

We will briefly return to legal aspects, potential follow-up costs, and how to deal with both in Section 9.3.4, but because this book primarily deals with enterprise and data architecture we will first examine the concerns of the CIO. In Chapter 6, we presented some approaches related to decision trees and machine learning. The problem addressed in this chapter can also be dealt with by algorithms from that area, which is why we won't be exploring them any further here.

Scoring algorithms

Another tool often used in medical applications is scoring. We are going to talk about scoring in detail in Chapter 10, Section 10.4.3. Put simply, scoring means assigning points to certain situations, circumstances, or facts, multiplying these points with weighting factors, adding it all up, and drawing conclusions on the basis of the total score. A typical example of a medical scoring system is the so-called APGAR score that makes predictions about the death risk of a newborn child based upon its heart rate, muscle tone, and complexion.

Using external knowledge

As noted previously, VV would not be able to build appropriate decision trees or to select—let alone develop—medical scoring systems. The employees in their IT department not only lack the mathematical and statistical knowhow but also the required medical background. This is similar to our example from the automotive industry in Chapter 8, but VV also has the option to source this knowledge externally (for example, via crowdsourcing).

Medical outsourcing not new

The idea of outsourcing medical services is not new. For quite some time, hospitals in the United States have been turning to radiologists in India or Pakistan to analyze X-rays. Dental prostheses for German patients are made in China, based upon German specifications or 3-D models and are sent to the ordering dentists by air freight. Therefore, there is no reason why trained, certified, and accredited experts in other countries shouldn't be commissioned to develop early diagnosis solutions for VV. This is why the head of marketing claims that the knowledge-related problems can be solved.

9.3.3 Numerical Example

Long-term objectives

Some 10,000 people live in the facilities operated by VV; 400 of these are dwell at the platinum facility on Majorca. Getting them to sign the respective service contracts would lead to maximum annual revenues of 250,000 to 500,000 EUR according to estimates from marketing, a part of which would have to be passed on to partners. This doesn't seem very lucrative, but there are some interesting long-term perspectives for VV:

- Some of the new offerings would probably be paid for by health or nursing care insurance, in which case they could also be sold in less exclusive homes.
- At all of VV's residences, money is not made through basic services (accommodation and catering) but by add-ons, such as medical or therapeutic treatments, health checks, special meals, fitness, leisure time or holiday offerings, transport services, and so on. This is espe-

cially true in the more exclusive gold and platinum facilities, whose residents can afford extras.

It seems obvious that the new offerings will dramatically widen the spectrum of services that can be billed either to cost-bearers (such as insurance companies) or residents. Apart from basic early warning systems, one could also conceive elaborate and supervised programs for mental and physical fitness. In addition to service revenues, VV could also generate money as an intermediary selling the necessary software and products.

- The most interesting aspect is that by developing this kind of solution, VV could create knowledge and establish a market position in an area that has great potential due to demographic trends. The role of an intermediary between the producers of medical devices and the providers of medical services seems even more interesting to the head of marketing than just selling something to the residents. Due to its normal business, VV has excellent networks in both arenas.

9.3.4 Conclusion: Mitigating Risks

Low-risk concept for VV

Together with the company lawyer and the CIO, the head of marketing has come up with three measures and a rough concept to help VV mitigate some of the risks involved:

- Start with noninvasive sensors
- Outsource analytical services and medical advice
- Crowdsource for new algorithms

Start with Noninvasive Sensors

To start, VV would like to work with an entry scenario based upon simple, noninvasive sensors. The first participants would be fitted with the following devices:

- Apple iPhone with belt attachment (equipped with GPS, compass, three-axis gyro, acceleration, proximity, surrounding light, and fingerprint sensors); the smartphone can also be used as a device for concentration and mental fitness tests/training

- Garmin vivofit with breast belt (included heart rate sensor, pedometer, and functions to record activities and sleep)
- Withings Aura (records movement, breathing, and heart frequency during sleep and records data about the sleeping environment: noise, room temperature, and light level)
- Tanita Body Composition Analyzer (measures body weight, body fat, muscle mass, and so on)
- Netatmo Weather Station (Indoor Module; sensors for air quality, carbon dioxide, humidity, air pressure, temperature, noise)

Monitoring emergency indicators

Together with a team of doctors, some simple indicators for emergencies (such as falls or irregularities in heart and breathing frequencies) will be identified. Using straightforward rules, people in charge, nurses, or ambulances will be informed as needed. Monitoring such parameters goes well beyond many existing services but does not yet require a big data solution.

Outsourcing Analytical Services and Medical Advice

Service providers responsible for compliance

The services required, such as data analysis, operating the early warning system, and medical consulting, are not going to be supplied by VV itself but by foreign partner companies that are managed by medical doctors. This concept is based upon three considerations:

- By outsourcing such services, VV can dispose of the approval-related problems of medical algorithms or devices as well as the liability. Properly constructed contracts would ensure that the external partners, not VV, would be responsible for the service quality.
- Because VV's partners are medical doctors that are registered, accredited, and qualified in their countries of residence, VV does not have to provide any services that are subject to approval restrictions. Furthermore, a certain minimum level of quality can be guaranteed.
- The partners can be based in low-cost countries. Because the residents will be fitted with iPhones, counseling can take place via video conferences. Furthermore, such structures—especially if the respective legal entities are located outside the European Union—will make it very difficult to enforce compensation claims.

The nitty-gritty legal details are to be clarified by VV's company lawyer in close cooperation with external specialist solicitors who focus on medical topics.

Crowdsourcing for New Algorithms

Analyzing more data

Monitoring the parameters listed under "Start with Noninvasive Sensors" is only meant to be the very first step. All other data collected as well are not going to be ignored but will be analyzed in a major medical crowdsourcing project together with medical doctors and medical faculties at leading universities, focusing on two questions:

- Are there other early indicators for upcoming health crises? Maybe health-related problems can not only be detected by a sudden drop in a person's heart rate but also predicted hours or even days in advance by a specific combination of heart rate, breathing, and sleeping patterns from the previous night.
- What individual dependencies exist between certain behavioral patterns, such as active and passive times, and important health parameters, such as physical fitness or body fat percentage? Based upon these data, one could develop individual recommendations and monitor their implementation.

In the context of this project, the first of these two questions (early indicators and alerting) is to be answered by a decision tree, whereas new, adaptive scoring models are to be developed to address the second one (lifestyle). In later phases, the palette of sensors used will be extended based upon the state of technology; one could also think about cooperating with producers, developing new, possibly invasive systems that might then need approval from their respective authorities.

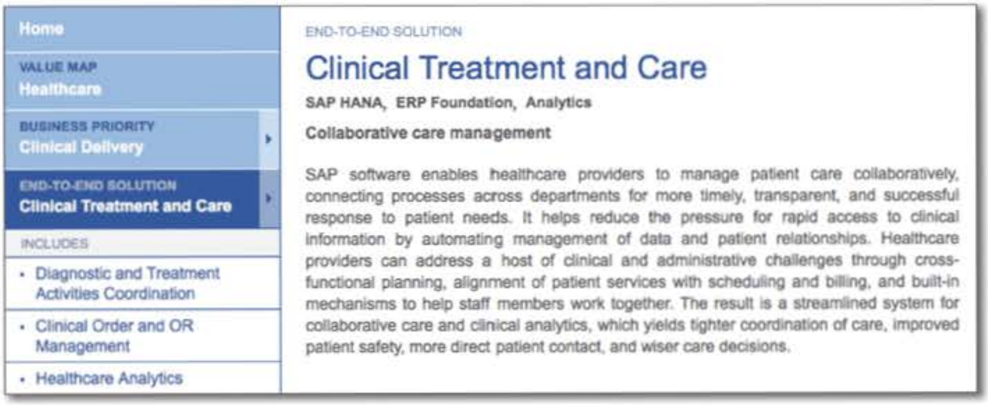
9.4 Solution: Big Data-Based Early Warning Systems

Real-time alerting process

Although we will discuss data storage briefly in Section 9.4.1, real-time alerting is our primary interest. We want to know how a real-time big data solution can alert users. The case study in Chapter 4 addressed similar questions (triggering alerts for outliers). There, we focused on detection, but not on informing affected users; this chapter will close this gap.

9.4.1 Related Value Maps in SAP Solution Explorer

SAP's industry solutions for healthcare are grouped under the umbrella of Patient Management (found in the Healthcare value map). The sensor data we are discussing here are diagnostic data from SAP's perspective; within the Healthcare value map, processing such data takes place within the Clinical Treatment and Care end-to-end solution (see Figure 9.2).


Home
VALUE MAP
Healthcare
BUSINESS PRIORITY
Clinical Delivery
END-TO-END SOLUTION
Clinical Treatment and Care
INCLUDES
• Diagnostic and Treatment Activities Coordination
• Clinical Order and OR Management
• Healthcare Analytics

END-TO-END SOLUTION

Clinical Treatment and Care

SAP HANA, ERP Foundation, Analytics

Collaborative care management

SAP software enables healthcare providers to manage patient care collaboratively, connecting processes across departments for more timely, transparent, and successful response to patient needs. It helps reduce the pressure for rapid access to clinical information by automating management of data and patient relationships. Healthcare providers can address a host of clinical and administrative challenges through cross-functional planning, alignment of patient services with scheduling and billing, and built-in mechanisms to help staff members work together. The result is a streamlined system for collaborative care and clinical analytics, which yields tighter coordination of care, improved patient safety, more direct patient contact, and wiser care decisions.

Figure 9.2 Clinical Treatment and Care End-to-End Solution

Proprietary data structures required

These solutions and their data structures will play a minor role in implementing our application. For three reasons, we assume that VV or its partners will have to develop the data structures to store their measurement data themselves:

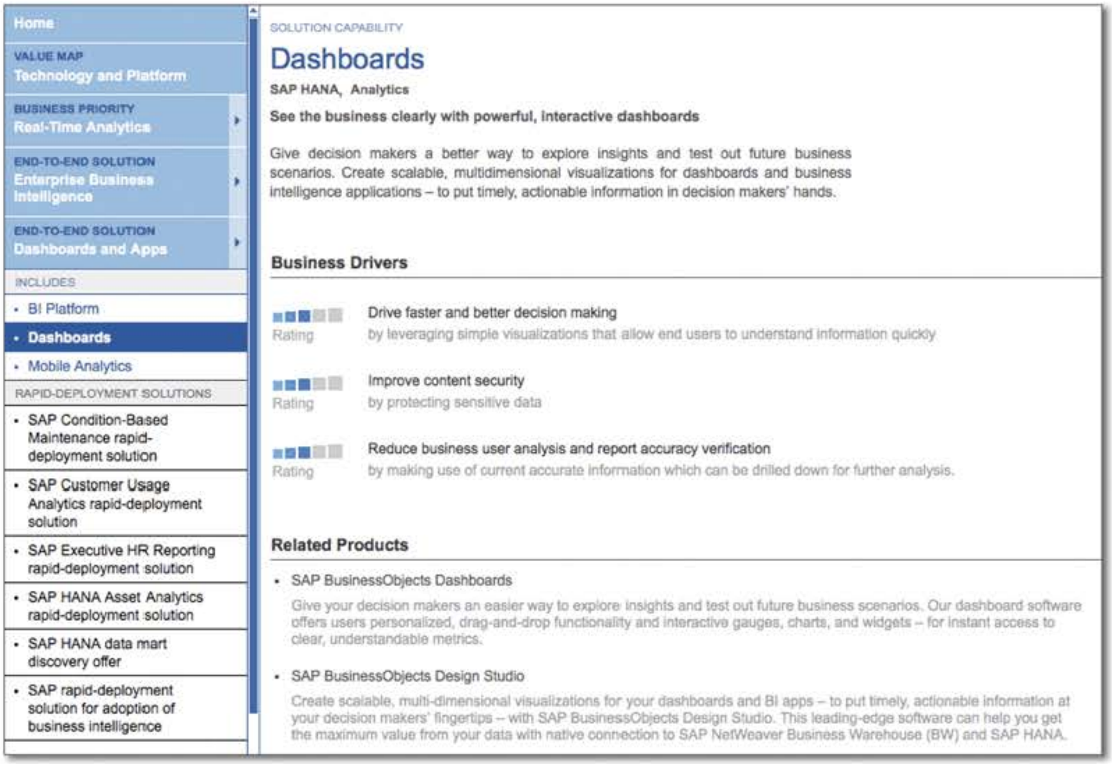
- SAP's solutions for healthcare are designed to store a limited number of diagnoses in electronic health records. When looking at our sensor data, we are not dealing with quarterly laboratory results but much higher volumes of data.
- The purpose of the solutions shown in SAP Solution Explorer is to process and store data, making them available to humans—health workers—upon request as quickly and conveniently as possible—for example, via mobile devices as with SAP's app SAP Electronic Medical Record (EMR). In contrast, our sensor data are not only to be stored

and retrieved but are to be analyzed by complex statistical algorithms. This is not what Patient Management is designed to do. At best, the results of our analyses can be filed within that industry solution.

- The structures in SAP's Clinical Treatment and Care end-to-end solution are tailored around certain processes and therefore relatively inflexible and by no means as abstract and adaptable as postulated in the previous chapters.

Generate reports and trigger alerts

This is why we are more interested in standard solutions from SAP that can be used to generate reports, trigger alerts, and forward messages. Here, solutions such as Dashboards (see Figure 9.3) or SMS/MMS for Enterprise come into the equation. We will take a closer look at the functionalities of these solutions in Section 9.5.1.


Home
VALUE MAP
Technology and Platform
BUSINESS PRIORITY
Real-Time Analytics
END-TO-END SOLUTION
Enterprise Business Intelligence
END-TO-END SOLUTION
Dashboards and Apps
INCLUDES
• BI Platform
• Dashboards
• Mobile Analytics
RAPID-DEPLOYMENT SOLUTIONS
• SAP Condition-Based Maintenance rapid-deployment solution
• SAP Customer Usage Analytics rapid-deployment solution
• SAP Executive HR Reporting rapid-deployment solution
• SAP HANA Asset Analytics rapid-deployment solution
• SAP HANA data mart discovery offer
• SAP rapid-deployment solution for adoption of business intelligence

SOLUTION CAPABILITY

Dashboards

SAP HANA, Analytics

See the business clearly with powerful, interactive dashboards

Give decision makers a better way to explore insights and test out future business scenarios. Create scalable, multidimensional visualizations for dashboards and business intelligence applications – to put timely, actionable information in decision makers' hands.

Business Drivers

■■■□□ Rating — Drive faster and better decision making
by leveraging simple visualizations that allow end users to understand information quickly

■■■□□ Rating — Improve content security
by protecting sensitive data

■■■□□ Rating — Reduce business user analysis and report accuracy verification
by making use of current accurate information which can be drilled down for further analysis.

Related Products

- SAP BusinessObjects Dashboards

  Give your decision makers an easier way to explore insights and test out future business scenarios. Our dashboard software offers users personalized, drag-and-drop functionality and interactive gauges, charts, and widgets – for instant access to clear, understandable metrics.

- SAP BusinessObjects Design Studio

  Create scalable, multi-dimensional visualizations for your dashboards and BI apps – to put timely, actionable information at your decision makers' fingertips – with SAP BusinessObjects Design Studio. This leading-edge software can help you get the maximum value from your data with native connection to SAP NetWeaver Business Warehouse (BW) and SAP HANA.

Figure 9.3 Dashboards Solution

The Dashboards solution can be found via TECHNOLOGY AND PLATFORM • REAL-TIME ANALYTICS • ENTERPRISE BUSINESS INTELLIGENCE • DASHBOARDS

AND APPS. The SMS/MMS for Enterprise solution is (among other areas) available under TECHNOLOGY AND PLATFORM • ENTERPRISE MOBILITY • MOBILE MESSAGING.

The SMS/MMS for Enterprise solution has been designed to send huge numbers of SMS/MMS messages in the context of sales activities and to analyze customer responses to these. In our scenario, we do not plan to send thousands of messages; the respective products are therefore probably overengineered for our purposes.

There is also an SAP HANA-based product for data exploitation in healthcare, the Healthcare Analytics end-to-end solution, but this solution focuses on operational and financial information and not on patterns in sensor data; it is therefore of limited use for VV's purposes.

9.4.2 Functional Requirements

Looking at the client layers

As stated earlier, we will focus on the client layers in this case study. We will therefore assume that everything needed to generate alerts or a list of alerts is already in place. We are not going to look at the algorithms for generating alerts (for this, go back to Chapter 4 or Chapter 6 or move ahead to Chapter 10); we will, however, still talk about administering these alerts, linking them to activities, and recording their statuses.

Alert List

As-is situation

Let's make the following assumptions:

- The data needed to detect health crises are available within SAP HANA.
- Together with external partners, algorithms that trigger alerts can be developed. For our deliberations here, it doesn't matter whether these alerts have been generated on the basis of simple if–then rules or based upon highly complex decision trees or statistical algorithms.
- Such an alert contains the following data:
  - Time of alert
  - Type of predicted crisis
  - Probability of predicted crisis (needed to prioritize emergencies; the keyword in this context is *triage*)

    - Timeframe for an intervention
    - Status of the alert (for example, are any activities already initiated or executed?)
- The alerts are available within a table of yours within SAP HANA.
- Certain decision tables within SAP HANA contain checklists that describe what to do in response to an alert depending on its type and details. Some examples of alert-related activities are as follows:
    - Send paramedics (plus details regarding required equipment)
    - Send emergency physician (plus details regarding required equipment)
    - Inform relatives by telephone
    - Inform family doctor by telephone
- SAP HANA also contains activity-specific data for each such measure—for example, the telephone numbers of relatives, or the email addresses of doctors.

Messaging/Process Feedbacks

Communication channel to emergency hotlines

If every item in the preceding alert list section is a given, then we are left with one single functional requirement. VV is looking for a solution that can access alerts and activity lists, trigger the required measures, receive feedback, and return status messages to SAP HANA. All of that can happen in different ways and via different communication channels:

- An alert is triggered with an employee on duty in an emergency call center. This person will initiate appropriate steps, such as sending an emergency physician or calling friends and family; he or she will also record the status of these activities within the system.
- Based upon alerts, automated calls are made that play recorded messages and request feedback—for example, by asking the person on the other end to press a certain key on their telephone.
- Short messages are automatically sent as an SMS, an MMS, or via any other kind of messaging application available. Feedback can also be requested—for example by asking the recipient to reply.

- Emails about an alert will be sent. Later, the system will follow up automatically via email, text message, or phone call, checking that required activities have been initiated on time.

A useful addendum to this process could be a cockpit or a couple of dashboards summarizing all alerts and their status in an overview and allowing users to drill down into each alert's details if needed.

9.4.3 Building Blocks of the Solution

In the end, we are talking about three different functionalities (handle alert lists and respective activities, messaging, and feedback processing). Before looking at technical options within the SAP space, we will briefly discuss the type of solutions we are looking for (always keeping an eye on things that are available for SAP HANA).

Alert List

Report showing alerts

In Section 9.4.2, we made the assumption that alerts and all the information needed to deal with them are already available within SAP HANA—for example, in the form of tables and views. To display these alerts within a user- or department-specific list, we need a reporting tool that can access data within SAP HANA via one of the common reporting interfaces, such as BICS or MDX.

Messaging

Automated or manual messaging

There are three different options for informing doctors, emergency services, relatives, or nurses via a voice call or a text message:

- A call is made or a text message is sent manually by an employee in a call center. This employee uses a normal telephone system or software for computer telephony integration (CTI). Appropriate software products have access to required data, automatically dial numbers, and record the status of a call or an alert.
- The whole process is fully automated using software that can make calls, generate spoken messages, and respond to inputs from the person who was called. The related technology is called Interactive Voice

Response (IVR) or Voice Portal. Similar products are available for automatically sending text messages.

- The whole thing happens via one or more web services. Compared to using locally installed software, web services have two advantages:
  - They are more flexible. Rather than being jacks-of-all-trades, they only fulfill one clearly defined task. As technology changes or progresses, web services are either adapted or replaced by competing ones.
  - If a web service is unavailable, one could simply use another service provider. It is obvious that this kind of redundancy of communication channels makes a lot of sense with medical emergencies.

Process Feedbacks

Processing feedback manually or automatically

If people or organizations affected by an alert are informed by a call center agent, this agent will also take the other party's feedback and update the status of executed activities based on that feedback, such as "relatives informed." In other cases in which there is an automated process—for example, via IVR—a system will have to handle feedback automatically. Such feedback could be provided by pressing a key on the telephone or by spoken word (which would require speech-recognition facilities); other options include answering an SMS, an MMS, or an email in a more or less structured format. Regardless of whether feedback is taken in and statuses are updated by humans or machines, such feedback will either be stored in a database or will be processed as events.

9.4.4 Potential Benefits and Value Drivers

VV's value drivers

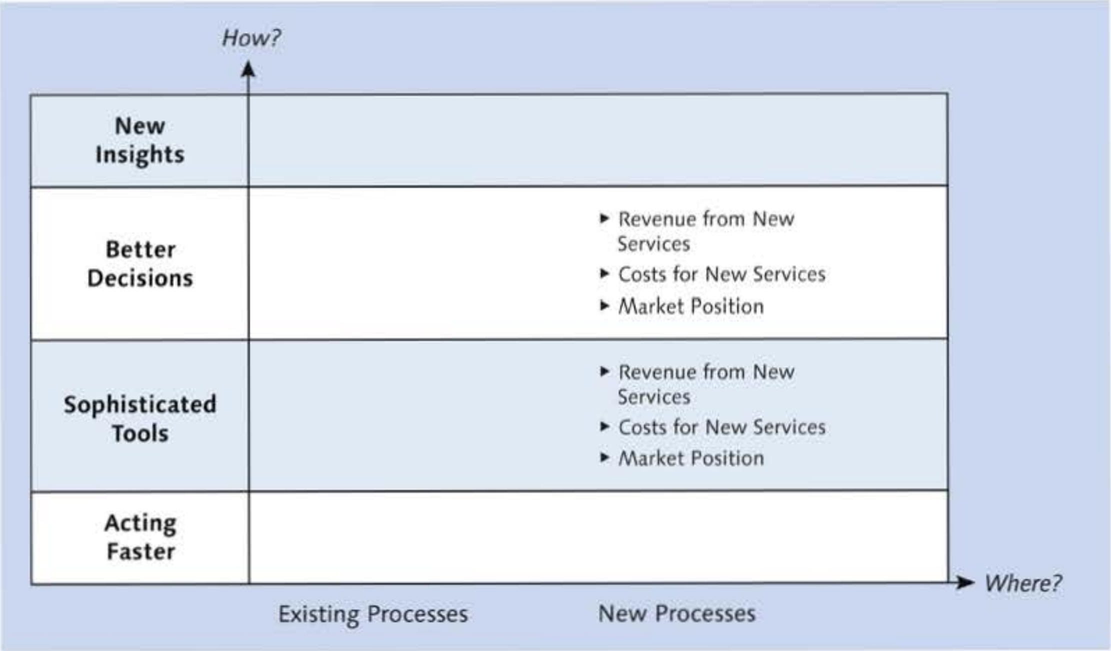
VV is developing its new solution to increase shareholder value by addressing three value drivers:

- The new service, plus any follow-up business related to it, will generate additional revenue.
- By automating the processing of emergency calls, VV can generate these revenues at lower costs than other suppliers of similar services.
- As a result, VV will be able to position itself as a pioneer and as a bearer of knowhow in a new and lucrative market; we assume that shareholders will recognize and reward this.

In the end, being a pioneer or having a strong position in a market is probably not the value driver in its own right but an advantage that rests on a few other value drivers—for example: market share, image (technological leadership), and dominance when setting de facto standards.

Doing new things or old things in a new way

Figure 9.4 shows the benefit–value driver matrix for this scenario. The value drivers mentioned previously are all related to a new business process or at least to handling an existing business process in a new and innovative way, which is why we have put all of them on the right-hand side in Figure 9.4. Fewer false negatives means making better decisions (second row in the benefit–value driver matrix); using sensor data instead of simple emergency buttons and early warning systems instead of a resident having to trigger an alert himself or herself boils down to deploying sophisticated tools (third row in the benefit–value driver matrix). Essentially, an emergency button is not so different from an old-fashioned bell used to call a maid or other house servant. In terms of revenues (income) and costs (expenses) for these new services, please refer to our explanations in Chapter 1, Section 1.4.3.



	Existing Processes	New Processes
New Insights		
Better Decisions		▸ Revenue from New Services ▸ Costs for New Services ▸ Market Position
Sophisticated Tools		▸ Revenue from New Services ▸ Costs for New Services ▸ Market Position
Acting Faster		

Figure 9.4 Health Management Benefit–Value Driver Matrix

9.5 Implementation Scenario and Framework Architecture with SAP HANA

In Chapter 8, we briefly talked about processing and analyzing sensor data, mentioning data from RFID transponders or sensors in cars or machines. In Chapter 8, Section 8.5.1, we also mentioned that many customers do not use the standard tables for RFID data and measurement documents within SAP ERP but rather rely on their own structures—for example, within SAP BW.

9.5.1 Implementation Scenario and Framework Architecture

Own structures: app scenario

With this case study, we are therefore once again more or less independent from structures within SAP's solutions, which means that we will have to design our own products for data generation. Hence, the app scenario (see Chapter 8, Figure 8.6) is once again the most sensible application architecture to start with. Theoretically, one may also have thought of the new SAP HANA apps scenario. However, you should take into account that some of the data we are dealing with—for example, the feedback linked to emergency-related interventions—are not generated within SAP HANA and therefore don't have to be written into the SAP HANA database by our application.

App scenario to process alerts

For this case study, the app scenario once again comprises databases, products for data generation, products for data exploitation, and clients.

Databases ❶

Data acquisition discussed in other chapters

Because we discussed data acquisition in other case studies, we aren't focusing on it here. Instead, we simply assume that all the data we need—that is, sensor data, data about events, and feedback from other parties—are made available within SAP HANA regardless of whether they stem from classic databases or event streams.

Products for Data Generation ❷

Software often delivered by maker of device

Under "Start with Noninvasive Sensors" in Section 9.3.4, we alluded to a couple of devices that could be handed out to the first customers using

VV's new offering. Each of these products comes with its own more or less open and more or less standardized software, so each is a product for data generation.

Products for Data Exploitation ❸

Algorithms discussed in other chapters

In Section 9.3.2, we stated that we will exclude the algorithms generating alerts from our considerations in this chapter. As mentioned there, you could choose, for example, decision trees or scoring.

Clients ❹

Focus on clients

The main subject in terms of the solution architecture for this chapter is the clients used to present alerts, to trigger adequate activities, and to select feedback and status messages. Depending on which of the three functions you are going to look at, there are different SAP and non-SAP solutions available on the market.

Alert List

SAP BusinessObjects BI

The products from the SAP BusinessObjects BI portfolio, such as SAP BusinessObjects Dashboards (see Figure 9.3), are perfect for generating flat or hierarchical lists, reports, diagrams, dashboards, or cockpits. In our example, the two big advantages of using products from SAP are that they are able to deliver all kinds of reports we need and they can easily access data within SAP HANA via standard structures, such as *universes*, the representation of a semantic layer within SAP BusinessObjects BI. The connection to SAP HANA can be made via the SAP Information Design Tool or, for example, via the SAP HANA XS Engine. Using universes between the reporting and database layers is optional.

Accessing information models within SAP HANA is a bit more complicated than reading data from simple tables. There are couple of threads on the SAP Community Network that address this topic.

Messaging

SAP offers a couple of products that could handle our messaging requirements. Apart from the previously mentioned SMS/MMS for Enterprise, the Customer Interaction Center (CIC) component within SAP ERP could be of particular interest. As mentioned previously, SMS/MMS for Enterprise is mainly designed for processing large quantities of messages, and CIC, although connectable to IVR systems, was not originally intended to work autonomously but to support call center agents.

Specialized service providers

Therefore, it is probably more appropriate to use specialized service providers, such as SMS gateways (for example, Clickatell), for your messaging requirements. Many of these service providers offer their solutions via a wide variety of interfaces. There are also offerings that can automatically inform people of an alert via a voice call (for example, Twilio or Tropo). The collective term for solutions such as these is *text-to-speech calling*. Using more than one service provider will give you the redundancy you should incorporate into an emergency call application.

Processing Feedback

SAP Event Stream Processor

Sending messages, the reaction of people or units informed, or their lack of reaction within a given time are all considered events that will lead to other activities. If a medical doctor that was called or sent an SMS five minutes ago has not yet confirmed the call or the receipt of the message, then one might have to inform a deputy or trigger an escalation. SAP Event Stream Processor (ESP) is one solution that can process these kinds of events. Due to the close integration between SAP ESP and SAP HANA (ESP Studio Plug-in for SAP HANA; see also Chapter 10, Section 10.5.2), one could even be adventurous and think about a self-learning solution. The data model shown in Section 9.5.2 includes that kind of mechanism.

9.5.2 Data Architecture

Again: Focus on clients (five client layers)

Our data architecture proposal in Figure 9.5 is focused on clients taking over very special tasks, which is why we have five client layers there (alert lists, activity lists, lists/cockpits/dashboards, messaging, processing feedback). Below these client layers there are two SAP HANA-based applications (products for data exploitation) plus the usual database(s) at the bottom.

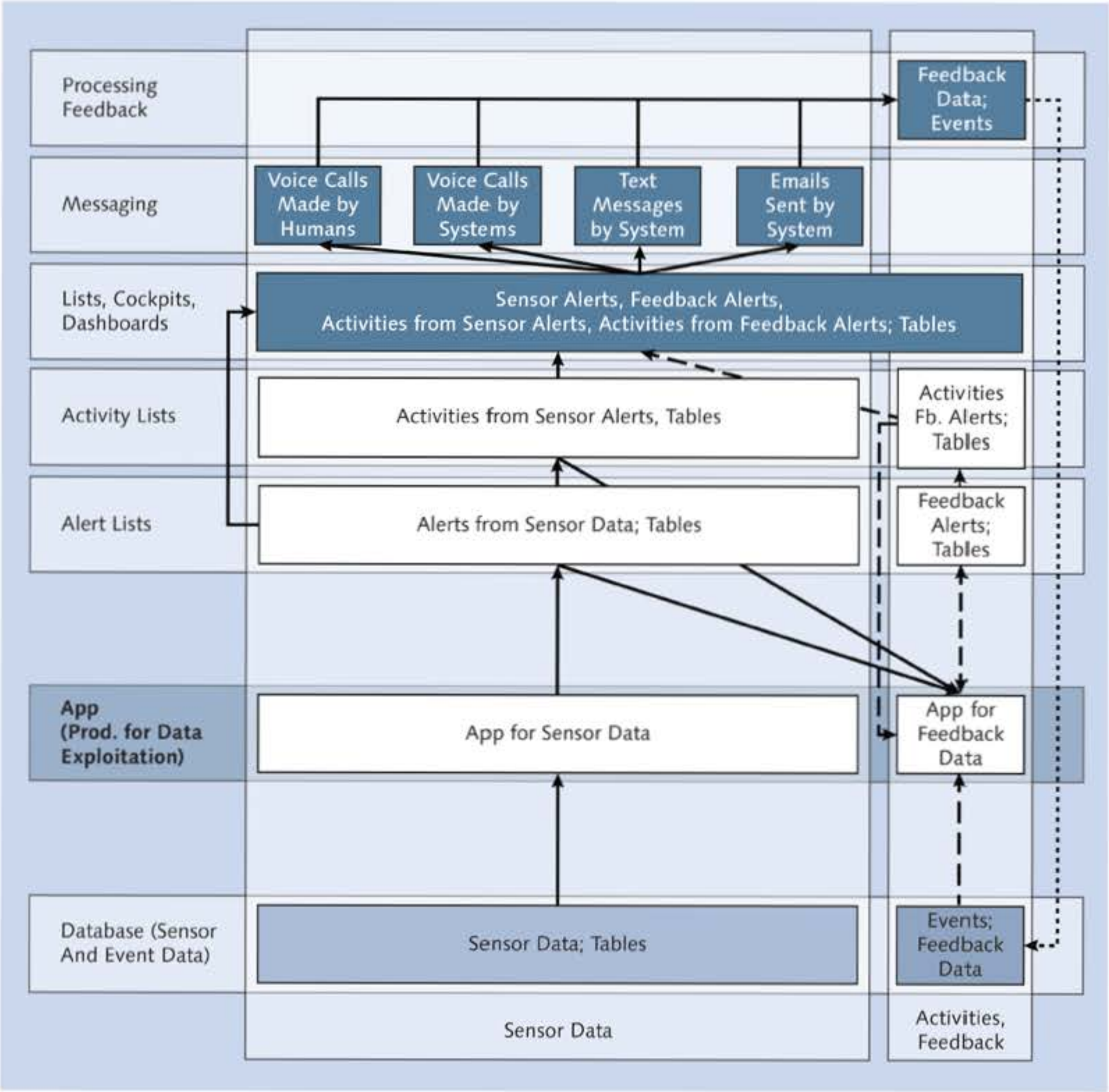

Processing Feedback
Messaging
Lists, Cockpits, Dashboards
Activity Lists
Alert Lists
App (Prod. for Data Exploitation)
Database (Sensor And Event Data)
Feedback Data; Events
Voice Calls Made by Humans
Voice Calls Made by Systems
Text Messages by System
Emails Sent by System
Sensor Alerts, Feedback Alerts, Activities from Sensor Alerts, Activities from Feedback Alerts; Tables
Activities from Sensor Alerts, Tables
Activities Fb. Alerts; Tables
Alerts from Sensor Data; Tables
Feedback Alerts; Tables
App for Sensor Data
App for Feedback Data
Sensor Data; Tables
Events; Feedback Data
Sensor Data
Activities, Feedback

Figure 9.5 Health Management Data Architecture

Database

Replicated and feedback data

Our database consists of sensor data that either arrive within SAP HANA in real time or are replicated there from other databases. It also contains information about feedback or statuses for alert-related activities. Under "Processing Feedback," we will briefly explain where these feedback and status data come from.

Application

Two applications

The application on the left-hand side generates alerts from sensor data. The one on the right-hand side handles resulting activities and feedback and generates alerts if a certain pattern consisting of alerts, activities, and feedback looks to be of concern.

Input data for applications

The app on the left-hand side needs only sensor data, which it uses to trigger alerts to which certain activities have been assigned in advance (for example, via decision tables; see Section 7.5.1 or Section 10.4.2 for details). The app on the right-hand side processes newly arriving feedback events and information about existing alerts and resulting or initiated activities, using all of that data to trigger additional alerts if there are problems, such as delays.

Alert Lists/Activity Lists

Alerts resulting from sensor and status or feedback data are collected in separate tables. The same applies to activities. There are two separate types of lists: one for activities triggered as a result of sensor alerts and one for activities triggered as a result of status-related or feedback-related alerts.

Lists, Cockpits, Dashboards

Consolidated reporting

All these alert and activity lists are consolidated on one common layer and—when necessary—aggregated for reporting. This leads to unified alert and activity lists and to summarized overviews that tell the manager of an emergency center more about the overall situation in terms of alerts, the statuses of related activities, and the respective feedback. Such overviews should be made available in the form of dashboards in which users can drill down as and when required.

Messaging

Open activities—regardless of whether they were generated by sensor or feedback/status alerts—will lead to messages being sent out by the respective communication channels.

Processing Feedback

Each of these messages will lead to a new status plus some kind of feedback in the form of an event; this feedback once again ends up in SAP HANA—for example, via SAP ESP, leading to yet another cycle starting at the bottom of our data model.

Data flows among processing steps

For the data flows in Figure 9.5, we used three different types of connections:

- Continuous lines stand for processes related to sensor data (the first processing cycle).
- The dotted line represents feedback events that are returned to the database, triggering the next (second) processing cycle.
- The dashed lines incorporate the third processing cycle, in which status or feedback data trigger new alerts, leading to further new activities, which once again end up within the consolidated layer (see "Lists, Cockpits, Dashboards").
- The dotted line from FEEDBACK DATA; EVENTS to EVENTS; FEEDBACK DATA doesn't spring into action until some feedback or statuses have been received/entered. From this time on, the application on the right-hand side will have to handle feedback-related alerts and feedback-related activities, which is why the dashed line between APP FOR FEEDBACK DATA and FEEDBACK ALERTS; TABLES is pointing in both directions.

Persistent tables for stability

Our data model includes a couple of third-party applications. Furthermore, stability is key in an emergency-related environment, which is why we are using persistent objects such as tables in a couple of places where virtual ones such as views are usually recommended.

9.5.3 Applying the Concept to Other Case Studies

Analogies in other industries

Our considerations within this chapter are not necessarily limited to healthcare; it's easy to imagine that similar applications could make sense within quite a few other industries. For Maneki-neko, our sample company in Chapter 8, alerting functionalities in a narrower sense might not generate shareholder value, but for a lot of other organizations, predicting crises in the wider sense and automatically triggering appropriate activities and following up their statuses and responses (feedback) from other parties can be worth a lot of money.

Think of the transport industry and logistics, about just-in-time production models, or forecasting delays at the level of individual containers based upon data from built-in RFID transponders, GPS receivers, or even marine weather forecasts. The latter takes us back to Chapter 5 and also proves the undeniable truth of Dettmar Cramer's statement at the beginning of Chapter 8.

It's no use shutting the stable door after the horse has bolted.

English proverb

10 Detecting Fraud Automatically

Derek had a look around. Not even the smallest cloud was in the clear blue springtime sky. The first bees of the year were refreshing themselves on the wild thyme. An idyllic, pastoral scene: only the lean, tall building with its smokestack seemed totally out of place.

Figure 10.1 Abandoned Copper Mine on the Copper Coast, Waterford County, Ireland

The lady who owned the B&B had told him that this area—now mainly inhabited by sheep—was an important copper-mining region barely 200 years ago. The hillside ruin with the grey plasterwork was the engine house of an abandoned copper mine. Due to dropping prices, copper mining in Waterford county became unattractive back in Victorian times, but maybe the operating companies should have just waited one century.

During the last couple of years, prices for copper had gone through the roof. Between 2010 and 2012, quotations had reached an all-time high. Copper had become so valuable that in 2013 German Telecom had started to mark *even subterrestrial copper cables with artificial DNA, trying to scare off thieves. Plant pots made of copper were no longer safe from theft outside the house.*

Embezzling copper and converting it into cash did not seem to be a new idea. Visiting an Australian museum, Derek once read that stealing copper was one of the crimes for which people could get deported to the red continent. Maybe the penalties were that drastic as an attempt to frighten off potential wrongdoers.

Nowadays, not only copper cables but also the data flowing through them could get stolen; in parallel, deterrents for that crime has also evolved. The consulting firm MSLE that Derek worked with for a couple of years operated some server farms in Switzerland for a number of major domestic and foreign banks. Seven months ago, one of his colleagues (the one always taking the last tea bag without telling anybody or going shopping) had sold the account details of a well-known German politician to tax-fraud investigators. Today, this same man—provided with a new identity and a decent financial cushion—resides on the Marquesas Islands.

Understandably, this episode had roused some of the rich and powerful and with them their banking partners. To rebuild its own reputation, MSLE felt the need to implement new software that would prevent such dishonorable activities ("dishonorable" referring to denunciation, not to tax evasion). The new solution analyzes certain properties (not specified) of employees, their activities in social networks, such as Xing or LinkedIn, and—insofar as the company's network was used for that—even their email communication and their surfing behavior on the Internet. These analyses have become the basis for defining risk-detection strategies that are said to have a hit rate of more than 90%.

Companies particularly at risk

Fraud detection is an ongoing challenge; a typical organization loses 5% of its revenues to fraud each year—a staggering amount. Although this is problematic for all organizations, three types of businesses are more at risk and affected than others:

- Companies with geographically dispersed operations
- Companies that manage valuable materials

- Companies that handle raw materials or finished/semifinished products that are not inventoried individually, like traditional retail products are

Such firms face particular difficulties when attempting to identify fraud in a timely manner. And the longer it takes to identify, the harder it is to stop the problem at its source.

[Ex]

Energy Theft

In developing countries, utility companies frequently deal with lots of small thieveries. People without a permanent address tap into power lines to use power with no meter, no bill, and no payment. In the past, it has taken up to nine months to identify patterns indicative of such fraud—and by that time the perpetrators have often moved on and can no longer be apprehended for criminal prosecution.

Utilities companies are just one example. Heavy industries (such as mining and steel) experience creative theft of supplies such as fuel, which is siphoned off from the tanks of transport trucks and resold on the black market.

Benefits of real-time fraud checking

SAP HANA helps detect such problems in real time. Its super-high performance creates the basis for new IT solution approaches that assist governance organizations within the company to align quickly to new fraud patterns and to adapt to changing behaviors. The benefits of such solutions are clear:

- Faster identification of genuine fraud cases
- Fewer false positive hits on innocent patterns
- More thorough documentation for law enforcement
- Improved revenues and margins

Once we have clarified some key terms, we will discuss a practical example. We'll include some numbers to give you cross-industry and industry-specific ideas about the size of the problem and the costs arising due to fraudulent activities. Finally, and independently from SAP HANA, we present a functional solution approach and explain its potential benefits and its impact on shareholder value. The last section discusses how such a functional approach could be implemented using SAP HANA.

10.1 What Does Fraud Management Mean?

In English, irregularities like the ones discussed in this chapter are often referred to as *fraud*; applications like the ones discussed here are called *fraud-management* solutions. Therefore, we would first like to clarify what exactly it is that we are considering to be fraud.

Quite often, fraud is related to cheating; the term does, however, also comprise criminal activities that might involve counterfeiting, confidence tricks, defalcation, and (to a certain extent) our examples regarding electrical energy or copper—that is, classic theft.

The legal definitions of fraud differ from country to country. Strictly speaking, fraud is related to misrepresenting facts or misleading people to them do or forebear something for one's own advantage. Fraud often includes the falsification of documents (for example, work reports, accounting documents, or contracts).

The UK Fraud Act of 2006 defines the following kinds of fraud:

- Fraud by false representation, in which a person makes "any representation as to fact or law ... express or implied" that they know to be untrue or misleading.
- Fraud by failing to disclose information, in which a person fails to disclose any information to a third party when they are under a legal duty to disclose such information.
- Fraud by abuse of position, in which a person occupies a position in which they are expected to safeguard the financial interests of another person and abuses that position; this includes cases in which the abuse consists of an omission rather than an overt act.

Embezzlement and abuse of trust

Looking at them from a criminal law perspective, many of the offenses at which fraud management is targeted are probably considered embezzlement or abuse of trust.

Taking and giving bribes

As with embezzlement, bribery is often related to misuse of power. Bribes might be requested for planning permissions, or a person's position in a business might be used to endow friends and family with orders instead of awarding them to the most suitable suppliers. If money is involved as well, then the legal term for this is usually something like "taking and giving bribes in commercial practice."

Quite often, fraud, theft, embezzlement, and bribery go hand in hand. For the purposes of this book, *fraud* is meant to cover fraud in the narrower sense, as well as forgery, unlawful appropriation, theft, embezzlement, abuse of trust, and taking and giving bribes.

Focus: detecting fraud

SAP Fraud Management, the solution discussed in this chapter, comprises fraud detection and downstream activities (for example, adequate and orderly documentation for law enforcement or triggering appropriate follow-up activities, such as blocking payments in accounting or banking-related systems). This chapter's sample scenario focuses on detecting theft.

10.1.1 Corruption in Purchasing: Caffeine Withdrawal and Exploding Coffee Machines

Caffeine shortage in the break room

Imagine the coffee break room for the purchasing team of a large organization. The productivity of the organization has recently been less than optimal; employees are complaining that this is because they are insufficiently caffeinated. Despite a new contract with a coffee vendor, there are rarely sufficient supplies, and the coffee machines often malfunction. In fact, last week one of them exploded, scalding an employee and leading to a workman's compensation claim.

Internal audit team investigating

A thirsty whistle-blower sends an anonymous report to the internal ombudsman, claiming that Terry in the purchasing organization is responsible. In particular, the whistle-blower asserts that Terry is a member of the same football club as the supplier, Carl's Caffeine, and thus might have a conflict of interest; he chose Carl's Caffeine not because of his ability to deliver but rather to help out his friend.

Safety problems as a result of dodgy deals

In addition, the tipster suspects that the exploding coffee machine might be due to Terry purchasing used equipment; he suggests that someone should investigate whether that equipment met safety specifications and was appropriately priced. Finally, he alleges that he has seen Terry pocketing coffee to take home for his family, leaving insufficient caffeine infusions for the toiling masses in purchasing.

10.1.2 Detecting Irregularities with Hindsight

Investigations are often time consuming

Prior to the advent of software-based fraud-detection systems, fraud investigation was a labor-intensive and time-consuming process. Often, corruption schemes could operate for months or years before being detected, and such detection could be dependent on the hit-or-miss strategies of luck or observant employees motivated to report suspicious behavior.

In our fictional case presented previously, the investigation of Terry's suspected fraud would require onsite interviews, poring over many paper-based invoices and other records, and observation from investigators. Such action is not only slow but also costly, and if fraud did not in fact take place, then the impact on morale for the people who were falsely accused could be significant.

Staring at the past

However, the primary problem in the past was precisely that the fraud was *in the past* by the time it was discovered; the loss had already taken place. Recovering funds after they are gone is far more difficult than preventing losses in the first place; safety issues—such as the exploding coffee machine, for example—were also discovered only after the damage had been done. Although uncovering past abuses will always be essential in a fully operating risk-management environment, most organizations strive to bring detection into a real-time system to respond in a timelier manner.

10.1.3 Detecting Irregularities in the Act

Catching perpetrators red-handed is possible

As organizations have moved to automated systems for capturing data and transactions related to multiple processes, including purchasing, it has become much simpler to research suspected fraud.

In the example of our coffee caper, with a software-based system someone in the head office can easily research the volume of business going to Carl's Caffeine and compare it with past patterns with other vendors. Alerts can be created to instantly notify a grievance officer, an investigator, an auditor, or a senior manager when a certain vendor is receiving unusual treatment, such as faster payments. This brings fraud detection more into the realm of the present; detecting issues becomes more effective if it happens quickly.

[«]

Investigator/Auditor

In this chapter, we use *investigator* and *auditor* synonymously. By *auditor* we mean specifically—regardless of local technical or legal terms—*internal* auditors.

Relevant data from various systems

However, for the purposes of our example, additional data sources beyond the purchasing system would be necessary to help enable real-time fraud detection. Integration with external sources, such as the roster of the local football club, the complaints-handling documentation for the break room, or the safety-incident reporting system, would all be valuable data points for identifying patterns that might otherwise go unnoticed. Nevertheless, the company would prefer to avoid stolen goods and exploding coffee machines altogether. Thus, they might want to invest in predicting incidents before they happen; they might want to open a window into the future.

10.1.4 Predicting Irregularities

Long-term vision: predicting the future

From an organization's perspective, it would be much better if the theft of coffee and exploding coffee machines could be foreseen and prevented. One might, therefore, consider investing in a system that could predict such events before they occur and thus open a window into the future. Much like Amazon, which patented anticipatory shipping in January 2014, or the precogs mentioned in the introduction of this book, there is value in predicting events before they occur.

Data for pattern recognition

One of the most unfortunate aspects of fraud is how common it can be; for most organizations it is a matter of when, not if. However, for each cloud there is a silver lining, and the silver lining of the large numbers of fraudulent incidents is that they provide a rich data source for modeling. Using information from past, proven cases of fraud, organizations can detect previously unknown patterns and so help prevent future losses.

[Ex]

Pattern Recognition

Data from maintenance (SAP ERP PM), from supplier master data (SAP ERP FI-AP/MM-PUR), or from site-access records might indicate a strong probability for irregularities if the following conditions are fulfilled:

- Vendors have sold equipment for which three or fewer bids existed.
- The master records for these vendors were approved between the hours of noon and 2:00 p.m. at the New York office.
- The sales personnel of these vendors visited our buyers onsite at least once a week.

The next time an invoice is presented by a vendor who meets those criteria, the system can flag the transaction for further investigation, stopping the payment until a risk-management expert decides that all is in order with both the finances and the purchased product.

If the equipment in question is safety critical, such as our coffee machine, such proactive action might not only save money but also protect life and limb.

10.2 Scenario: Theft in a Surface-Mining Operation

Mining company Iron Bug LLC

Iron Bug (IB) LLC is a large, global mining company with multiple sites. It has three surface mining locations: Alpha, Bravo, and Charlie. Each has been producing iron ore for several years. The average operating costs for supplies and materials at these locations has remained relatively steady over time, ranging from 72 cents per metric ton of ore produced to 83 cents per metric ton of production. These costs include food for the dining operations, disposable safety supplies, such as dust masks and respirators, office supplies, fuel for trucks and equipment, and even the water that has to be trucked to the site.

10.3 Traditional Investigation Methods: Costs, Risks, and Opportunities

Let's take a look at the some of the costs, risks, and opportunities associated with traditional investigation methods.

10.3.1 Problem: Inexplicable Increase of Extraction Costs

Extraction costs higher than usual

Mysteriously, the cost per ton at the Alpha location has been steadily creeping upwards. It eventually hits 89 cents per ton, far above the corporate goal of not more than 79 cents.

Internal investigations

Traditionally, such problems were approached manually at IB. Someone in a senior position orders an inquiry or audit and appoints a resource to dig into the records of the operation. This time-consuming process can involve anything from a several-days-long walkthrough of the mine site to extensive interviews with personnel to exhausting data analysis and paperwork reviews.

Manual data acquisition and analysis

In addition to on-site reviews, someone researching the cost increase in the past might import purchasing and consumption data into a legacy tool, such as Microsoft Excel, and complete steps similar to the following:

- **Check significance**
  Identify whether the difference in cost per ton between the sites is statistically significant at different points in time. When doing so, you need to take into account operational differences among facilities and locations.
- **Identify drivers**
  Which elements have changed at the site that might account for the difference? One approach might be to ascertain the proportions over time. For example, if fuel was consistently 12% of the overall costs prior to the increase and it is now 24%, then that is very informative. On the other hand, if all of the costs have gone up equally, then that might be indicative of a different root cause.
- **Uncover causes**
  For each of the elements that have increased, analyze the causes of the rise.

Disadvantages of manual process

Although the general approach outlined here is sound, following it via a manual process is fraught with problems:

- **Effort (time)**
  A large investment of time is necessary for onsite investigations, to identify the correct data sources, to import, to preprocess, and to further process data.
- **Inaccuracy**
  Inevitable human error introduced during the import or analysis process can lead to inaccurate investigations that may lead to false accu-

sations against innocent parties, ineffective oversight of true fraud cases, and a waste of investigative resources.

- **Proprietary knowledge**
  Once the investigation is completed, the insights gained during the research are typically isolated and available only on a single computer or via the professional experience of the investigator. Rarely are the patterns detected available for widespread use in finding similar future cases.

10.3.2 Break Room versus Heavy Industry

Detecting fraud in a heavy industry such as mining is considerably more complicated than detecting fraud in a company's coffee break room. Furthermore, fraud can occur not only in purchasing, but in any area of business.

Broadening the break room example

Imagine that instead of there being one small, contained environment in a single building, operations are spread out over thousands of miles and often in countries that have endemic corruption. Instead of a handful of local vendors, operators have thousands of choices from a global web of thousands of suppliers, many of whom in turn have suppliers of their own.

Safety is a key aspect

In our coffee example, the primary risk was burns from an exploding coffee machine. In our mining example, obtaining faulty equipment or supplies can lead to far greater safety risks, including collapses from faulty roof bolts, environmental damage created by unqualified contractors, or inoperable lifesaving equipment, such as personal oxygen supplies.

Fortunately, the same concepts for detecting fraud in the past, present, and future for our break room can apply to our mining example as well. Just as caffeine enthusiasts want to ensure that they're paying a fair price for quality supplies made with reliable equipment, mining operators also want to be sure of the following:

- The safety of employees and the local population are not threatened by (unsuitable or inferior) equipment and services.
- They are making purchasing decisions based on what is best for the business and supportive of long-term, sustainable growth.

- They stay in compliance with complex international regulations.
- The goods and services that are bought are the best ones from a technical and commercial perspective and are provided at reasonable prices.

Ideally, businesses also want to do all of this proactively, moving from a culture of fraud discovery to one of fraud prevention.

10.3.3 Numerical Example

What is the true value of improving the fraud-management process? Estimates vary, but it is clear that fraud engenders both financial and morale costs in organizations.

Fraud in general

According to a 2012 report from the Association of Certified Fraud Examiners (ACFE), which focused on occupational fraud, a typical organization will lose 5% of its revenues to fraud each year—equivalent to roughly $3.5 trillion if applied to the 2011 gross world product. Approximately half of those losses are never recovered at all, and fewer than 15% of victims are able to fully recover their losses.

Fraud in mining

The same report indicated that the median loss to a mining organization for a single instance of fraud is $500,000, with the majority of cases involving some form of corruption. The median loss from a case of fraud in the study was higher for mining than for any other industry (which is why we chose mining for the scenario in this chapter).

A 2011 report from Transparency International echoed this concern: according to their Bribe Payer's Index, the mining industry is one of five industries most likely to be impacted by bribery. Ernst & Young, in their 10th Global Fraud Survey, found that 47% of respondents from the mining industry indicated that corrupt practices were prevalent.

Risks affecting businesses and people

Many studies of fraud in the mining industry indicate that the remote location of mining operations, in countries where political and socioeconomic risks are high, contributes to the fraud challenge faced by those organizations. Transparency International publishes data on the perceived fraud risk in different countries; many of the highest-risk countries are precisely those in which valuable minerals are most likely to be found. This chapter focuses on fraud risks in mining because fraud risk amplifies the other risks inherent to mining—namely, health and safety risks, not only for the miners themselves but also for stakeholders in the local communities. Equipment purchased from a vendor whose products are not genuinely certified to mine-safety standards puts workers at risk. Permits and inspections issued as a result of bribes—and not as a result of proper inspection—can result in damage to the local environment and can permanently destroy the reputation of the organization and result in the loss of its license to operate.

10.3.4 Conclusion: Using New Technologies

Risks for reputation

Leaders in the mining industry are strongly motivated to improve this situation. They view ethical operations as essential not only to their financial performance but also to their ongoing sustainability as organizations. Establishing an environment of integrity is seen as a basis for attracting and retaining top talent and for enhancing the predictability of financial performance. It is also a cornerstone of responsible operations.

Detecting fraud faster

Fortunately, there are now many opportunities for improvements in fraud management beyond what was available in the stick-it-in-a-spreadsheet days. Fraud *detection* is not enough; the ACFE study showed that schemes continued for a median of 18 months before being detected. Thus, fraud *management* and fraud *prevention* are the true goals. Our next section will outline an approach to achieving those goals using SAP Fraud Management powered by SAP HANA.

10.4 Solution: Flexible Fraud Management Using a High-Performance Appliance

In this section, we will explain which new approaches IB could use in fraud management. Later on, Section 10.5.1 discusses how SAP HANA can contribute to this.

10.4.1 Related Value Maps in SAP Solution Explorer

Affected processes and industries

As mentioned before, fraud, corruption, or theft can affect all kinds of functional areas within a business and therefore also a variety of business processes. Using SAP Fraud Management is therefore not limited to applications in mining or other heavy industries. A modern system that detects suspicious cases can also be valuable for banks or insurance companies, in medical services, for manufacturers, or in public services.

You can find the solution SAP Fraud Management itself in more than just one value map of SAP Solution Explorer.

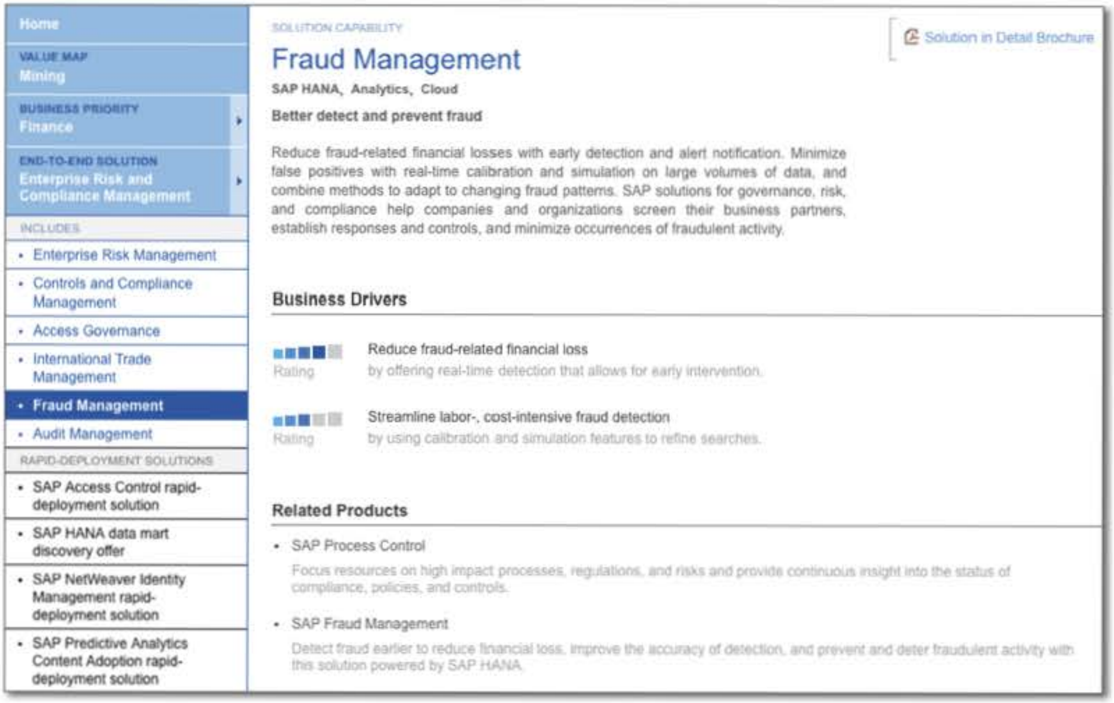
Rather than looking at a specific industry route, you can access it via the Browse by Technology section of SAP Solution Explorer:

- Browse by Technology • Big Data • Applications Using Big Data • Fraud Management

Two industry-specific paths also take you to SAP Fraud Management:

- Browse by Industry • Mining • Enterprise Risk and Compliance Management • Fraud Management
- Browse by Industry • Mining • Big Data • Applications Using Big Data • Fraud Management

As an example, Figure 10.2 shows the destination of the last path.


Home
VALUE MAP
Mining
BUSINESS PRIORITY
Finance
END-TO-END SOLUTION
Enterprise Risk and Compliance Management
INCLUDES
• Enterprise Risk Management
• Controls and Compliance Management
• Access Governance
• International Trade Management
• Fraud Management
• Audit Management
RAPID-DEPLOYMENT SOLUTIONS
• SAP Access Control rapid-deployment solution
• SAP HANA data mart discovery offer
• SAP NetWeaver Identity Management rapid-deployment solution
• SAP Predictive Analytics Content Adoption rapid-deployment solution
Solution in Detail Brochure

SOLUTION CAPABILITY

Fraud Management

SAP HANA, Analytics, Cloud

Better detect and prevent fraud

Reduce fraud-related financial losses with early detection and alert notification. Minimize false positives with real-time calibration and simulation on large volumes of data, and combine methods to adapt to changing fraud patterns. SAP solutions for governance, risk, and compliance help companies and organizations screen their business partners, establish responses and controls, and minimize occurrences of fraudulent activity.

Business Drivers

Rating — Reduce fraud-related financial loss
by offering real-time detection that allows for early intervention.

Rating — Streamline labor-, cost-intensive fraud detection
by using calibration and simulation features to refine searches.

Related Products

- SAP Process Control
  Focus resources on high impact processes, regulations, and risks and provide continuous insight into the status of compliance, policies, and controls.
- SAP Fraud Management
  Detect fraud earlier to reduce financial loss, improve the accuracy of detection, and prevent and deter fraudulent activity with this solution powered by SAP HANA.

Figure 10.2 SAP Fraud Management Solution

Value drivers

The solution-specific details on SAP Solution Explorer mention two possible value drivers:

- Reduce fraud-related financial loss
- Streamline labor-, cost-intensive fraud detection

From our perspective, there are a few other relevant factors; we will return to these in Section 10.4.4.

10.4.2 Functional Requirements

Brainstorming workshop with IB

Let us return to our scenario of the mining organization with multiple sites, in which one site has shown an increase in the cost of supplies when compared with two similar sites. After some initial research, IB has organized an internal workshop, during the course of which six key functional expectations of a new fraud-management solution that supports its detection and management of fraud cases have been defined. For each of these six requirements, a short, high-level description has been documented:

- **Replicate data**
  A key disadvantage of the current processes within IB is that data are entered and then imported into spreadsheets manually. For the future, management at IB want a system that can pick up all data relevant for fraud management from SAP ERP or other operational (OLTP) systems and replicate them in an in-memory database in real time. This not only reduces the opportunity for errors introduced by manual copying but also opens up the opportunity for much more extensive data sources to be cataloged. Furthermore, keeping the link to source documents will also make it a lot easier to produce the evidence needed for criminal charges.
- **Use existing experience**
  Investigators and auditors within IB have always learned from past experience, and they have also used their knowledge from earlier cases in current investigations. But the inventory of experience-based knowledge was usually limited to experiences from the specific investigator himself. Tapping into that knowledge was based upon intuition rather than on a certain systematic approach. Therefore,

although IB would like the new system to come with certain predefined rules they also want to go one step further in the future. They expect their new solution to learn from historical cases of fraud. In other words, the system should be seeded with dozens, possibly hundreds, of cases of proven and disproven fraud. Based on that information, the solution can compare any new cases with those same patterns and detect similarities instantly. Furthermore, it can be taught different scenarios for different business situations.

To reduce false positives to an absolute minimum and to also avoid false negatives (that is, real cases of fraud remaining undiscovered), as much data as possible should be taken into account in the course of the evaluation process. Such data could be not only transaction-related but could also refer to the environment in which a transaction might take place.

For example, a pattern of multiple purchase orders in small quantities to small businesses within 50 miles of a mine site might be indicative of proven fraud cases for a mine site in North America. However, a similar pattern might be normal business behavior for mine sites in South Africa, so investigators can tailor alerts to be sensitive to the particular circumstances of each individual data point.

- **Clearly visible status**

  Once the system is in place, risk-management executives should start the day with a simple glance at their dashboards. Such dashboards are intended to display alerts on active cases as well as on new suspicious patterns automatically detected by the system. In other words, there is no need to wait for a tip-off or for a red flag from finance or for a concern to filter down from executive management. Instead, the system is on constant lookout for concerning patterns.

  The dashboard should visually differentiate between cases initiated by a tip-off versus those identified through application of the analytical rules. It should also provide a rating and risk value for each alert, which serves as an indication of the likelihood of fraud and the amount of business risk and financial loss posed by the particular case; furthermore, the dashboard would identify the suspected *type* of fraud—for example, conflict of interest, billing, expense reimbursement, and so forth—which can be helpful when assigning investigative resources.

In our particular scenario, the increased costs at the mine site with IB, this could mean that a problem will not have to wait to be discovered in quarterly reporting; instead, the dashboard would display an alert almost immediately after the cost increase information was available within the various datasets (presuming that the rise in extraction costs matches historical fraud patterns and that data are replicated in real time). This allows for timely allocation of a skilled investigative resource.

- **Support investigation process**
  The system should support investigators in both detecting fraud and managing reported cases of suspected fraud. For the latter, IB's management is looking at the following functionality:
  - Support auditors with further case-specific analyses (for example: why has this specific alert been triggered?)
  - Monitor the investigation process (workflow management)
  - Automatically trigger events in operational systems (such as payment blocks)
  - Document cases (to support criminal and civil proceedings and to reuse this information in the system's future learning cycles)
  - Improve related internal processes
- **Evaluate cases**
  The majority of risk management organizations are engaged in a constant process of prioritization. It is not possible to investigate every suspicious activity or delve deeply into every tip-off. Nevertheless, uncovering fraud that has already taken place is far more costly than identifying, and stopping, fraud in real time. It is far more difficult to recover stolen goods from a thief than to prevent the thief from receiving the goods in the first place.

  Therefore, it is desirable to set up a system of alerts that allow for customizable weighting factors and thresholds and that can prioritize cases for investigation to help allocate resources efficiently to where they are most urgently needed.

- **Predict fraud**

  The icing on the cake would be a function via which the system is not only able to detect embezzlement when it occurs but can even be in a position to predict it, very much like crimes in the movie Minory Report are predicted and prevented.

  If that were possible, risk managers would be in the position to recognize suspicious behavioral patterns *before* any kind of fraud could take place and cause harm.

  With that ability, IB could prevent financial loss, and even dangers to people's lives and limbs, in advance. Thus, IB, its employees, and other affected stakeholders would remain untroubled by the (potentially life-threatening) impacts of criminal activities. Such considerations will, however, also lead us to complicated ethical, political, and legal issues. Whenever we talk about big data, encountering these issues becomes almost unavoidable; resolving these issues, however, goes far beyond the scope of this chapter and even this book, and so we are not going to discuss fraud prediction any further.

10.4.3 Building Blocks of the Solution

The building blocks of our solution correspond to the functional requirements, as defined in the previous section.

Replicate Data

Easy and fast data acquisition

Static rules have clearly defined input parameters that do not change over time, so for these rules only these input parameters need to be replicated into your fraud-management solution. However, when identifying new patterns you will (by definition) not know which parameters are suitable as input for early indicators, which means that you may want to obtain more rather than less data from products for data generation (such as your ERP system).

Ultimately, all of the data that exists in all your systems might be relevant for detecting fraud, and how can you select the right values at such an early stage? Consequently, you will need a relatively powerful solution for data logistics. Whether or not this solution is also meant to clean or transform data (instead of just fetching or receiving them) depends on the homogeneity and the quality of the data within your SAP or non-SAP source systems.

Use Existing Experience

Develop new models

If you are using empirical data to identify new patterns, then you are trying to detect, define, and monitor dependencies, much like in Chapter 5. In other words, you are trying to find links, develop models that are based upon these links, and continuously monitor/improve/refine these models.

Quantify dependencies

As an example, by analyzing records related to known incidents of fraud compared with clean transactions, you may be able to discover that there is a strong correlation between fraud and certain patterns. You may find that a pattern of abnormal fuel consumption at an IB site is typically related to theft of diesel. Although onsite personnel had assumed that the excessive consumption was due to equipment being in need of maintenance, which is itself a problem in need of rectification, in fact the fuel was being siphoned off for sale on the black market. As a result, when it comes to serving this specific business requirement, more or less the same algorithms as discussed under "Modeling Dependencies" in Chapter 5, Section 5.4.3 of our travel costs scenario can be applied, which is why we will not take you through these algorithms for a second time.

Monitoring models

In addition to developing models, the models themselves also need to be monitored. In Chapter 4, we looked at monitoring models. If you would like to go beyond simply monitoring them and also improve your assumptions while monitoring, then you might want to consider approaches from Bayesian inference.

[»]

Bayesian Inference

Bayesian inference is a term encompassing tools and algorithms used to improve estimates about a population in a step-by-step process. Starting with reasonable assumptions about that population (for example, about the age structure of your customers), you revise and thus improve those estimates on the basis of new evidence from samples or observations (for example, with each new customer who enters your shop).

Clearly Visualize Status

Preparing data for different user groups

Reports from fraud management will probably have a number of recipients:

- **Managers**
  Mangers (in risk management as well as in other areas) will need an overview in terms of fraud. They might be anxious about answers to the following questions:
  - How many cases of suspected fraud are there or were there (status, course, trends, etc.)?
  - How much money is or was at stake (status, course, trends, etc.)?
  - Which areas of the business were affected and how intensely (status, course, trends, etc.)?
  - What kind of difference did possible measures make?
  - Which new patterns were discovered by the system?
  - What is the performance of the solution like? How long does it take to detect fraud, and how many false positives are there (status, course, trends, etc.)?
- **Investigators/auditors**
  Although investigators might also be interested in the questions listed for managers, they would probably want to see more detailed information:
  - How (that is, based upon which predefined or automatically generated rule) has a case of suspected fraud been detected?
  - How likely is fraud in a specific case (fraud probability, fraud rating, or fraud score)?
  - Are there any comparable historical cases? In what way were they similar? What can we learn from them?
  - How much money is at stake in each case?
  - Which kind of fraud (conflict of interest, overinvoicing, irregularities with travel expenses, bribery, etc.) might have been detected in a specific case? This question becomes important when considering further investigations or getting the authorities involved.

- How much progress has been made in terms of researching each individual case?
- Which immediate measures (such as payment blocks in an ERP system or other measures in SAP Process Control) have been taken?
- Which documents are available in other systems?

- **Other parties**
  A couple of other parties within the organization might be interested in certain sections of the reports mentioned previously:
  - Subordinates of the CFO (mainly in controlling) need to know more about the financial impact of fraud in the past, present, or future.
  - Managers of certain departments or functional areas (for example, in purchasing) need information about the volume of fraudulent activities in their specific areas and about patterns they should keep a closer eye on.
  - Experts within the legal department would like to ensure that the company has taken timely and adequate action against irregularities that may later trigger high-compensation claims (exploding coffee machines and collapsing mine entrances, for example).

Report hierarchy or report network

Therefore, reporting in fraud management will not have to deal with only one report but with a portfolio of reports. Ideally, all of these dashboards need to be mutually consistent; there should also be functions that allow users to jump from one dashboard into another one or from summary reports into detailed lists and vice versa. All reports should be simple and easy to use; the portfolio of reports could be presented hierarchically (in the form of a reporting tree) or as a web of equivalent reports (that is, as a network of reports).

Mobile user interfaces

Managers and investigators often travel; hence, all relevant reports should also be accessible from mobile devices. Considering the diversity of mobile user interfaces today, it seems sensible to use open standards that make sense for desktop and mobile devices (HTML5).

Support Investigation Process

Integration with workflow management

Dashboards are not only meant to list or visualize the status of individual cases; they should also be fully integrated with workflow solutions (such as SAP Process Control, SAP Audit Management, or other products in the realm of SAP GRC). To trigger events in other (operational) systems, suitable interfaces are required.

Evaluate Cases

Scoring systems used for evaluation

Evaluating cases deals with determining fraud scores (which we mentioned earlier; see "Clearly Visualize Status"). Fraud scores might reflect probabilities, potential amounts of loss (money at stake), threshold values, nominal factors (for example, is injury to persons possible, yes or no), or combinations of all these parameters. In general, there are many so-called scoring algorithms in a variety of subject areas (for example, credit scoring or medical scoring).

[«]

Scoring

Scoring is the summarizing of a number of cardinal, ordinal, and/or nominal values, using algorithms to generate one single figure, the *score*. Such scores are then used to make decisions (for example, should a company grant a loan, yes or no). Because scoring is very similar to modeling, the quality and performance of scoring algorithms can be monitored by using the approaches described in Chapter 4.

With scoring, you often run into three problems:

- You are summarizing individual values that—strictly speaking—cannot be summarized at all; aggregating, for example, cardinal and nominal values is like comparing apples and oranges. Clearly, such an approach has its drawbacks.
- When aggregating values, you (by definition) lose detail. Furthermore, you are either forcefully quantifying or ignoring qualitative, nonquantifiable facts, which means that your decisions are (at best) based upon a fraction of the information you were provided with.
- A scoring algorithm that has produced excellent results at a certain time or place might utterly fail elsewhere or even on the very next day. In military healthcare, for example, the chances of recovery from a bullet wound in a warm and humid tropical climate are very different from the chances of recovery in the Antarctic. Scoring systems are used in these situations to determine which patient to treat first.

Predict Fraud

Time-shifted dependencies

Looking into the future also means using models, but instead of trying to quantify the dependency of simultaneous events, you are working with time offsets. At a certain point in time (let's call it t), you would like to know which effect observations you are making now (at t) or have made in the past (at t – x) might have on certain parameters in the future (at t + y). The higher x and y become, the more difficult and unsafe such predictions tend to be (just think of the weather forecast).

Building time-shifted models is not so different from building ones without time shifts, which is why you may want to return to the methods and algorithms discussed in Chapter 5. In addition, there are some statistical approaches that focus on analyzing time-dependent observations (called *time-series analysis* or *time-series forecasting*; we briefly mentioned time-series analysis in Chapter 1, Section 1.1.2).

10.4.4 Potential Benefits and Value Drivers

Effort and gain

Some of the classic and generic value drivers we discussed in Chapter **1**, Section 1.4.3—that is, expenses/income and uncertainty—also loom large with fraudulent activities. Expenditures can be related to the value of stolen goods or to excessive purchase prices but also to costs caused by detecting, investigating, documenting, or prosecuting cases of suspected fraud. Income could be affected if, for example, losses occur because products or used equipment are being sold well below their market value.

Personal injury and environmental damage

In cases of fraud, however, there are two further substantial risks:

- **Personal injury and insurance premiums**
  We have already pointed out that embezzlement, theft, or corruption can lead to personal injury. In turn, this leads to potential compensation claims that, if they can be mitigated at all, can only be handled by paying high insurance premiums.
- **Environmental damage and insurance premiums**
  Environmental damage is something that may always lurk—especially in heavy industries. Such environmental damage can easily reach levels that threaten a company's very existence; even worse, employees at all hierarchical levels might be exposed to the risk of criminal investigations in countries whose legal standards are, at best, questionable.

On top of expenses/income and uncertainty—for example, in mining—other generic value drivers related to the perceptions, expectations, and preferences of shareholders and stakeholders might come into the play:

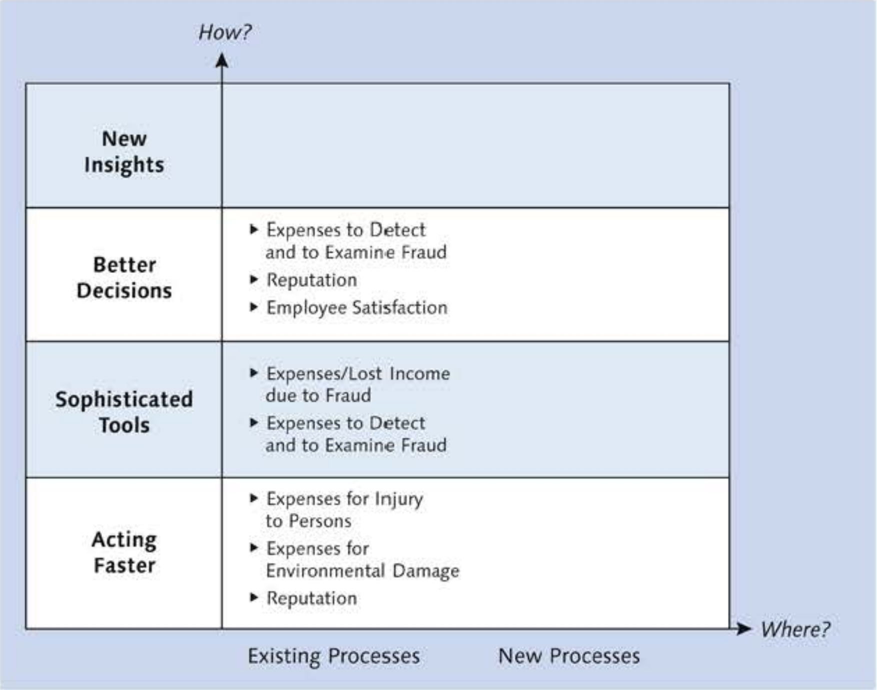
Perception of shareholders and stakeholders

- **Reputation**
  Many companies in heavy industries are concerned about their reputations. Time and time again, raw materials or mining conglomerates come under criticism due to degrading labor conditions, adverse effects on the environment, or the provenance of their products. Think of all the oil spill disasters, such as the one in the Gulf, or of nuclear disasters, such as the ones at Chernobyl or Fukushima, or of the term *blood diamonds* and the related movie starring Leonardo DiCaprio.

  As we explained in Chapter 1, Section 1.4.3, under "The Shareholders' Perceptions, Expectations, and Preferences," public perception of a business could well have an impact on its shareholder value. On top of that, major buyers are trying to protect their own reputations and are therefore starting to pay more and more attention to certain minimum standards within their supply chains. The standing of a company like IB (which delivers iron ore to steel mills and therefore indirectly to the automotive industry) can therefore come under pressure if customers in downstream markets raise their standards. In the age of the Internet and Twitter, the global public might learn about misdoings within minutes (forcing you or your customers to spend a lot of money to fix problems that could have been prevented with much less effort).

- **Employee satisfaction**
  It is obvious that the morale of your employees will suffer from false accusations related to fraud. By triggering fewer false positives, you are also reducing this risk and the resulting negative effects on value creation.

Figure 10.3 provides you with an overview of all of these value drivers. Neither the business processes in which cases of fraud might occur nor detecting or investigating fraud is entirely new, which is why we have put all value drivers into the matrix's left-hand column. The types of insights big data applications can come up with regarding fraud are also not revolutionary (which is why we have not put any of the value drivers into the first line of the matrix). However, with SAP Fraud Management and/or SAP HANA, you are able to gain these insights faster, with higher quality and fewer false positives and false negatives, and while investing less time and effort.



	Existing Processes	New Processes
New Insights		
Better Decisions	▸ Expenses to Detect and to Examine Fraud ▸ Reputation ▸ Employee Satisfaction	
Sophisticated Tools	▸ Expenses/Lost Income due to Fraud ▸ Expenses to Detect and to Examine Fraud	
Acting Faster	▸ Expenses for Injury to Persons ▸ Expenses for Environmental Damage ▸ Reputation	

Figure 10.3 Fraud Management Benefit–Value Driver Matrix

10.5 Implementation Scenario and Data Architecture with SAP HANA

Heterogeneous data sources

Apart from data that are *sent* to workflow solutions or ERP systems (for example, to block payments), fraud management mainly deals with analyzing data that were *received* (that is, replicated) from somewhere else. If the majority of data processed in fraud management originates from applications that are part of SAP Business Suite (and if SAP Business Suite is running on SAP HANA as well), such data do not have to be replicated, but even then analysis in fraud management usually also relies on complementary data from non-SAP sources or local files.

10.5.1 Implementation Scenario and Framework Architecture

App scenario fits best

The app scenario is therefore the most common architecture variant for SAP Fraud Management (see Chapter 2, Section 2.2.1). If SAP Business Suite on SAP HANA is being used, then some replication processes will not apply, leading to a hybrid scenario that is a combination of the app scenario and the SAP Business Suite on SAP HANA scenario. Apart from the omission of replication for some data, there are no substantial differences between the app scenario and a hybrid one, which is why in this chapter we are focusing on the app scenario (refer back to Chapter 8, Figure 8.6).

Databases

There is value in detecting fraudulent activities as soon as possible. We have emphasized this a couple of times. Nevertheless, in fraud management we are normally not talking about an event-oriented real-time process (except perhaps for financial transactions, such as credit card fraud). Hence, real-time data acquisition is of minor importance for classic fraud-management processes (such as the one for IB that we are using as an example in this chapter).

Relational databases

In most cases, the data needed for analysis will come from classic relational databases. These databases may belong to SAP solutions or non-SAP applications, or they may represent local data that were collected by investigators. In the future, unstructured data will become more important. If relationships between purchasing managers and suppliers are to be checked, then data from social networks could be useful (we will not examine any legal aspects in this context). Hence, *graph databases* (special databases that are used to store relationships such as links between people in social networks) might come into play. Analyzing in detail what relevant data might look like is not part of our solution, however. We simply assume that all data required for IB's fraud management system are available in a form that allows for replication into SAP HANA.

Data acquisition

Depending on where required data come from and whether and how they have to be preprocessed, different ETL tools could be used. We have provided you with examples of such tools in Chapter 2, Section 2.2.1, including the following:

- SAP Landscape Transformation Replication Server
- SAP Replication Server
- SAP Data Services

Note that data that are already stored in SAP HANA (either because they belong to an SAP HANA-based SAP Business Suite or SAP BW system or for any other reason) do not have to be replicated. To properly separate the corresponding source objects from those required for SAP Fraud Management, you might want to use views. Data that are not in SAP HANA could be accessed in a similar way via virtual tables.

Products for Data Generation

How data are being generated determines whether we require a pure app scenario or a hybrid scenario (app and SAP Business Suite on SAP HANA). Apart from that, data generation is irrelevant for the (fraud-management) solution that is in the limelight in this chapter.

Products for Data Exploitation

SAP Fraud Management

When it comes to exploiting/analyzing data, SAP Fraud Management (running on SAP NetWeaver AS 7.4 as a platform) might be the product of choice.

In addition, you may want to use some supplementary applications:

- SAP Predictive Analysis (PAL)
- Certain products developed by SAP InfiniteInsight, a company purchased by SAP in 2013 (InfiniteInsight Modeler, InfiniteInsight Scorer, InfiniteInsight Factory, and so on)

As of the time of writing, SAP Fraud Management was shipped with some 50 preconfigured rules that could be used to detect irregularities straight away. This supply of rules covers different industries and processes and is continuously being extended; all of these rules are static, however. Up to this point, SAP Fraud Management is not so different from other products for fraud detection or prevention (such as, for example, FirstStrike by APEX Analytics).

Detection method and detection strategy

In SAP Fraud Management, however, your portfolio of rules can be widened flexibly and (to a certain extent) even automatically. The latter is made possible by using dependency-detection algorithms from SAP PAL or InfiniteInsight Modeler to identify new patterns that might indicate fraud.

In addition, SAP PAL (for example, via its function `SUBSTITUTE_MISSING_VALUES`) or another SAP InfiniteInsight product (InfiniteInsight Explorer, for example) can support you in preparing your data for analysis.

Both the rules that were shipped and the new ones have the same value and can be integrated into and processed by one single detection strategy. In SAP Fraud Management, detection strategies are used to pool so-called detection methods, which is SAP's technical term for what we casually called a *rule* in fraud management so far. Technically speaking, detection methods are procedures that were implemented in SAP HANA's language, SQLScript, which means that those rules preconfigured by SAP are also detection methods with corresponding procedures.

Detection objects

Detection methods process individual detection objects and calculate a score for each of them. Strictly between us, detection objects are nothing but datasets that are meant to be checked in SAP Fraud Management—for example, a subclaim of an insurance claim. In addition to the tools explained in Chapter 4, scoring models and algorithms can be implemented, monitored, and enhanced by using InfiniteInsight Scorer or InfiniteInsight Factory.

Supplementary functionalities can be implemented in R

In case the functionalities of SAP PAL or InfiniteInsight Modeler don't fulfill all your needs, you could still use R (such as the packages `arm` or `bayesSurv` for Bayesian inference). Even scoring algorithms can be built using R; there are loads of detailed instructions for this on the Internet: simply search for "R" and "scoring" for further information.

When it comes to passing on information (for example, regarding required payment blocks) to ERP applications, SAP Fraud Management provides you with a couple of SOA components. As of today, the corresponding functionality still has to be implemented by the customer. Other interesting options in this setting (that is, the collaboration of SAP Fraud Management and other SAP or non-SAP solutions) are decision tables in SAP HANA (which can be used to control processes in an external application) or SAP Process Orchestration.

Clients

Visualization using SAP BusinessObjects BI

In Section 10.4.3 (under "Clearly Visualize Status"), we emphasized the importance of user interfaces in detecting, managing, and preventing fraud. All standard reports in SAP Fraud Management are exclusively based upon SAPUI5 and can therefore run without any components from SAP BusinessObjects BI. Nevertheless, deploying SAP BusinessObjects BI may still make sense for you because:

- You should expect quite a few horizontal (content-related) and vertical (level of detail) overlaps between the reporting requirements of your various target groups. This suggests not looking at each report individually but instead feeding all reports from clearly defined, common sources (such as universes created in the Information Design Tool).
- Remember that your (SAP or non-SAP) fraud management solution needs to be embedded into a company-wide data architecture. Such data architectures can be designed and managed using enterprise information management tools from the SAP BusinessObjects BI suite of products.
- At the very least, investigators should be in a position to quickly create ad hoc reports that satisfy special information requirements. SAP BusinessObjects Web Intelligence or SAP Lumira are easy to use and tailor-made for ad hoc reporting.

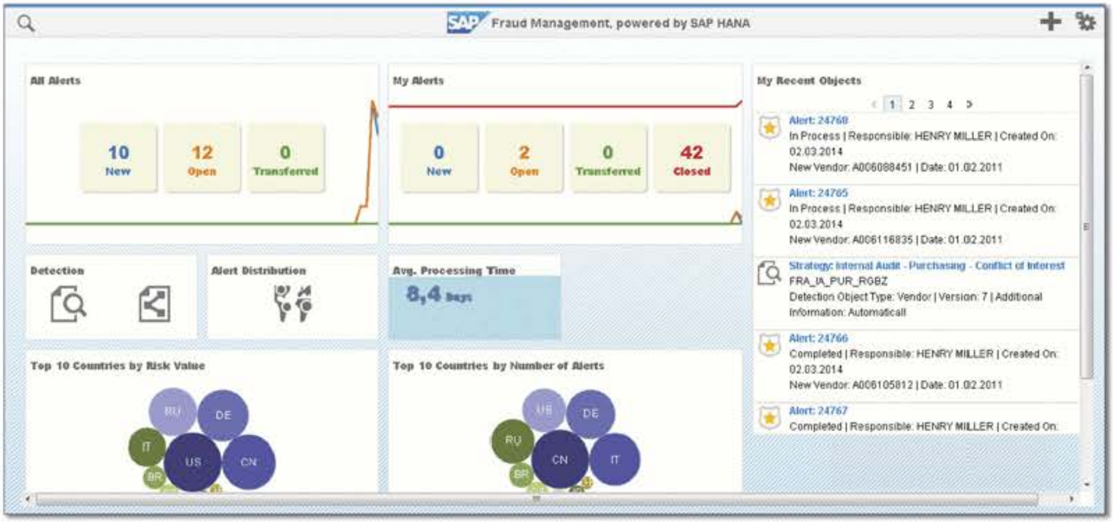
Developing frontends using SAPUI5

Customers also are free to extend the portfolio provided by SAP by developing their own SAPUI5/HTML5 reports.

Figure 10.4 and Figure 10.5 show two sample (standard) dashboards from SAP Fraud Management.

Dashboards in SAP Fraud Management

In our example, the home screen shows the total number of all new, open, and transferred alerts (ALL ALERTS). Clicking on this tile takes you to the ALERT WORKLIST, the total number of all the new, open, transferred, and closed alerts that have been or are currently assigned to you (MY ALERTS). Clicking here again takes you to the ALERT WORKLIST, the average time it took to process an alert, the top 10 countries by risk value (TOP 10 COUNTRIES BY RISK VALUE), and the top 10 countries by number of alerts (TOP 10 COUNTRIES BY NUMBER OF ALERTS). Clicking on either one of the top 10 lists will take you to the EXECUTIVE DASHBOARD (see Figure 10.5).



Alert: 24760
In Process | Responsible: HENRY MILLER | Created On: 02.03.2014
New Vendor: A006088451 | Date: 01.02.2011

Alert: 24765
In Process | Responsible: HENRY MILLER | Created On: 02.03.2014
New Vendor: A006116835 | Date: 01.02.2011

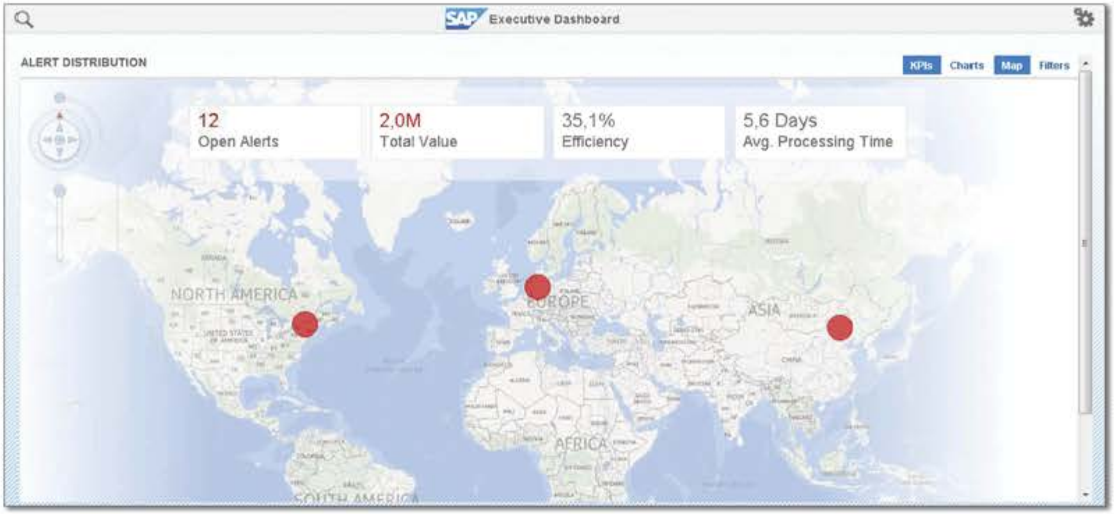
Strategy: Internal Audit - Purchasing - Conflict of Interest
FRA_IA_PUR_RGBZ
Detection Object Type: Vendor | Version: 7 | Additional Information: Automaticall

Alert: 24766
Completed | Responsible: HENRY MILLER | Created On: 02.03.2014
New Vendor: A006105812 | Date: 01.02.2011

Alert: 24767
Completed | Responsible: HENRY MILLER | Created On:

Figure 10.4 SAP Fraud Management: Home Screen

From the home screen, you could also jump to maintaining DETECTION STRATEGIES (left icon in the DETECTION group), maintaining DETECTION METHODS (right icon in the DETECTION group), or to a more detailed map view, as shown in Figure 10.5 (via ALERT DISTRIBUTION).



12
Open Alerts

2,0M
Total Value

35,1%
Efficiency

5,6 Days
Avg. Processing Time

Figure 10.5 SAP Fraud Management: Executive Dashboard

10.5.2 Data Architecture

Adaptive fraud management

Figure 10.6 shows one possible data architecture framework for an adaptive fraud management system. The model shown will give you some food for thought and help you develop ideas that are going to work in your own business environment. Numerous videos and documents provided by SAP on the Internet explain how to set up, configure, and calibrate a system to detect suspicious transactions using SAP Fraud Management.

A step-by-step guide in the form of screen recordings can, for example, be found on the SAP Community Network (SCN) (see the additional resources in the book's online appendix for the link). The videos show how to build a solution that classifies purchase orders as suspicious if there are major discrepancies among ordered, delivered, and invoiced quantities. We are also going to use these examples to explain some distinctive features of our own data architecture proposal in Figure 10.6.

Features of our data model proposal

There are two key differences between our approach and SAP's example (or many other similar examples on the Internet):

- **Flexibility**
  Chapter 6 and Chapter 8 go into considerable depth about what an adaptive and flexible architecture for a dynamic environment should look like. When discussing these matters, we emphasized two properties:
  - Automatically generated data flows
  - Separation of content-related and semantically neutral metadata

  By contrast, the example from SCN is designed as follows:
  - The solution is based upon one static rule (detection method). The only dynamic aspect of the approach lies in the fact that this rule is parameterized: the percentage threshold, above which deviations among ordered, delivered, and invoiced quantities are considered suspicious, is defined as an input parameter and can therefore be adapted if required or even automatically optimized (*calibrated*) by the system.
  - The whole data flow is developed and configured manually.

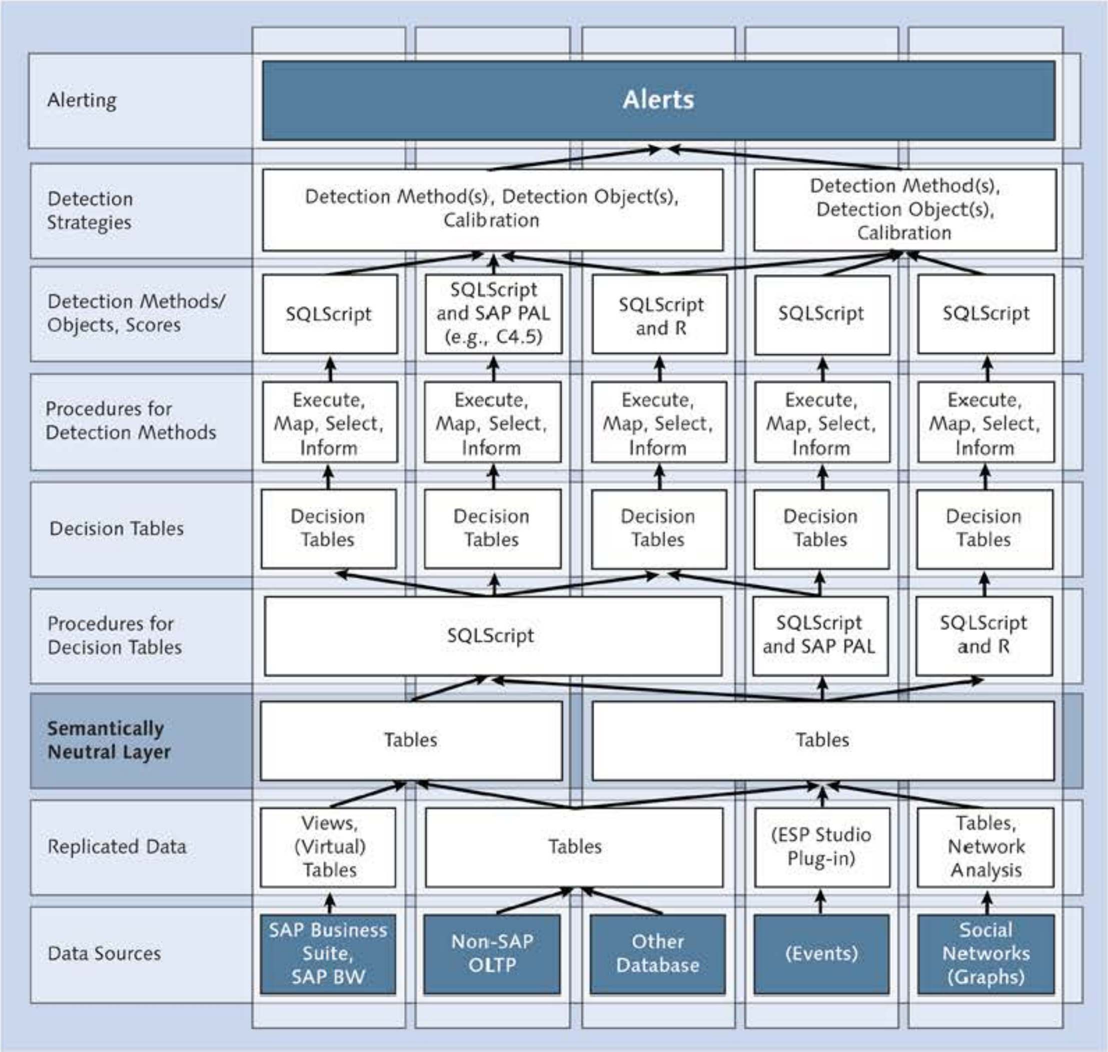

Alerting
Alerts
Detection Strategies
Detection Method(s), Detection Object(s), Calibration
Detection Method(s), Detection Object(s), Calibration
Detection Methods/ Objects, Scores
SQLScript
SQLScript and SAP PAL (e.g., C4.5)
SQLScript and R
SQLScript
SQLScript
Procedures for Detection Methods
Execute, Map, Select, Inform
Execute, Map, Select, Inform
Execute, Map, Select, Inform
Execute, Map, Select, Inform
Execute, Map, Select, Inform
Decision Tables
Decision Tables
Decision Tables
Decision Tables
Decision Tables
Decision Tables
Procedures for Decision Tables
SQLScript
SQLScript and SAP PAL
SQLScript and R
Semantically Neutral Layer
Tables
Tables
Replicated Data
Views, (Virtual) Tables
Tables
(ESP Studio Plug-in)
Tables, Network Analysis
Data Sources
SAP Business Suite, SAP BW
Non-SAP OLTP
Other Database
(Events)
Social Networks (Graphs)

Figure 10.6 Fraud Management Framework Data Architecture

- There is no separation of content-related and semantically neutral metadata. Although ordered, delivered, and invoiced quantities are probably identical in terms of their semantically neutral metadata, they are treated as three totally different key figures (which means that all three would have to be tackled in case of modifications—for example, if changing the units of measurement).

In the long run and particularly if you have a large number of detection methods and detection strategies, such an approach produces nontransparent and cumbersome silo structures that look a lot like the architectures that produce headaches for IT managers in data warehousing and business intelligence today. In Figure 10.6, we propose an alternative extended model that contains a semantically neutral layer and uses decision tables in SAP HANA.

- **Induction instead of deduction**

  In Chapter 5, Section 5.4, we laid out the disadvantages of a deductive approach in data analysis. Furthermore, as an alternative, we have provided options for implementing inductive techniques. In Chapter 7, we also stressed that assumptions about real facts (in that particular case, customer behavior) must not simply be implied as given facts and then modeled statically but rather should be determined empirically and implemented as rules that can change dynamically, automatically, and in real time.

  In contrast to this, the example on SCN explains a rule for identifying fishy orders on the basis of data but does not explain why the rule is valid nor does it tell you how the algorithm behind the rule has been developed.

  How do you know that you should keep an eye on deviations among ordered, delivered, and invoiced quantities? Perhaps order size is a far better indicator for fraud? Perhaps there is a dependency among all three parameters that can be put down in mathematical terms, and perhaps each violation of that model (that is, each outlier) is superior to percentage deviations when it comes to detecting fraudulent transactions.

Flexible data model

Theoretically, one could create an unlimited number of detection methods and associate these with detection strategies, but that would cause quite a bit of work for those who have to configure the system, and it would lead to nothing but (manually!) taking stabs in the dark, hoping to find something useful.

If you act in this way, then you will, at best, only tap into a subset of the potential insights that big data or SAP HANA could provide you with. Instead, we believe it makes more sense to design a flexible model right from the beginning. By applying a higher degree of abstraction, you can cover a far greater search area when scanning your data for indicators of fraud.

Nine separate layers in the data model

Our proposed architecture (as shown in Figure 10.6) consists of nine sample layers, of which seven (all except the ones at the bottom and at the top) are to be implemented in SAP HANA or SAP Fraud Management. In the following sections, we will provide you with a brief description of each layer.

Data Sources

Data from various systems

We are assuming that IB's SAP HANA environment is being fed with data from SAP Business Suite, SAP BW, other ERP or OLTP solutions, any other databases, local files, or social networks. Data from social networks can be useful when it comes to revealing relationships among purchasers, among suppliers, or among purchasers and suppliers.

Just in case it's needed, we have also added an event-related data flow (for example, for tweets); this data flow, however, has been put into brackets, because we are not focusing on real-time solutions in this chapter. Although SAP Fraud Management does offer such functions (called online fraud detection), we assume that batch detection will suffice for our scenario. Online fraud detection with real-time event processing might, however, be required in other industries (such as financial services).

Replicated Data

Data acquisition using views

Records from various data sources are replicated into SAP HANA using the mechanisms described in Section 10.5.1. If SAP Fraud Management requires data from SAP Business Suite or SAP BW running on SAP HANA, then often such data do not have to be replicated. Instead, you may just mirror them in views for fraud-management purposes. These views might be structured in a slightly different way from your source tables (using, for example, joins, as defined in Chapter 4, Section 4.5.2) and might also only contain a selection subset of the records available there.

When accessing event-based data from SAP Event Stream Processor, the ESP Studio Plug-in for SAP HANA will probably be your tool of choice. This plug-in makes it possible to directly access data in SAP ESP from SAP HANA Studio's modeling environment. If data from social networks are to be used, these data could be prepared, analyzed, or enriched immediately before or after replicating them into SAP HANA. There are tools for network data analysis in SAP Fraud Management (Network Analysis) as well as in SAP PAL (the `LINKPREDICTION` function). As usual, there are even more powerful (but also more demanding) tools in R (the `igraph` package, for example).

Semantically Neutral Layer

Generating and splitting off metadata

We strive to look at data without prejudice and to split corresponding metadata into content-related and semantically neutral categories, which is why we have inserted a semantically neutral layer just above the one containing replicated data. Our design is based upon the assumption that IB possesses tools (such as metadata repositories) that allow the company to store both content-related and semantically neutral data.

When entering the semantically neutral layer, data for key figures, such as ordered quantity, delivered quantity, and invoiced quantity, are stored in three identically structured tables or even within one and the same table. Using an ID that is part of this table's key, one could easily determine which records belong to which key figure.

Destination: metadata repository

IDs in data records within the semantically neutral layer point to two metadata-related destinations: three different records in the metadata repository's content-related section (describing the contents of the key figures) and one single record in the metadata repository's semantically neutral section (representing the metadata object that describes the [identical] mathematical properties—such as the scale levels—of these key figures).

The semantically neutral layer therefore serves two purposes:

- It enriches raw data coming in from the layer underneath (replicated data) by adding content-related and semantically neutral information.

- It resolutely segregates data from metadata, keeping only semantically neutral data and storing content-related and semantically neutral metadata in two separate areas of IB's metadata repositories.

Procedures for Decision Tables

Combing through the space of options

All of this leads once again to an extensive space of options (see Section 10.4.2). A big data solution can comb through this space in search of real and meaningful dependencies.

Once ETL processes have finished, the semantically neutral layer will only contain semantically neutral data and will have swapped out content-related and semantically neutral metadata to IB's metadata repositories. It is now time for a giant submersible mixer called *inferential statistics* to dip into the semantically neutral layer, creating confusion, chaos, and panic among the data and thereby perhaps triggering a Darwinian selection process that will breed the odd viable organism (that is, a detection method based on newly discovered dependencies).

Although this mixer can use the whole arsenal of algorithms (SQLScript, SAP PAL, R, and any other third-party tools, which can be used in parallel or in combination) as discussed in Chapter 5 through Chapter 7 to find dependencies, distinctive features, patterns, outliers, or relationships, it will have to stick to the restrictions defined by what is mathematically possible (that is, the restrictions specified by semantically neutral metadata in the metadata repositories). Gems found by stirring and sifting this mess will enter the next layer (decision tables) and become the basis for future decisions (in terms of detecting fraud).

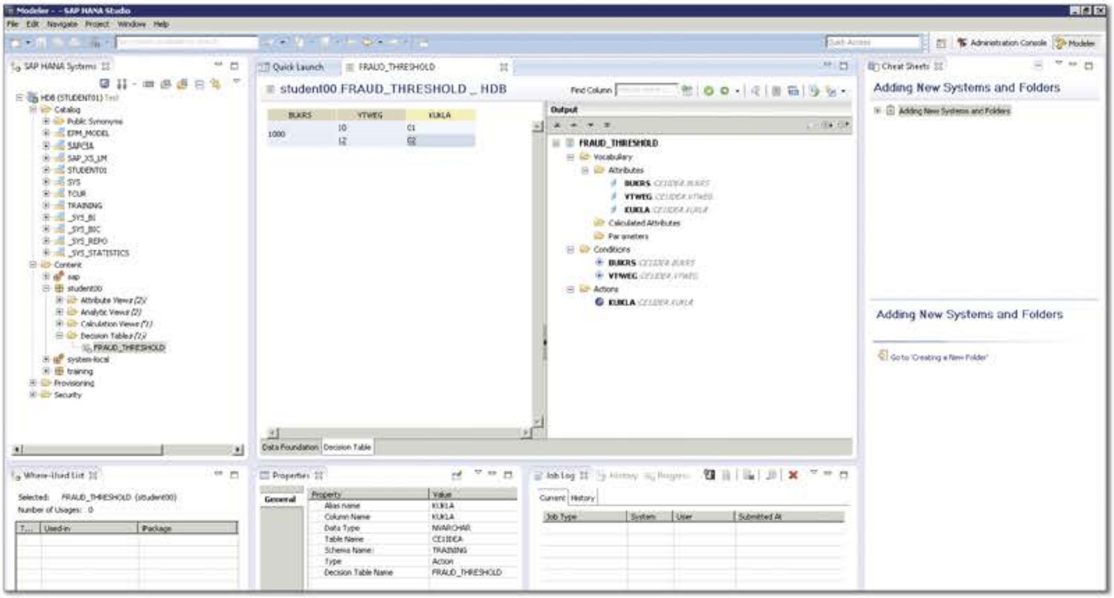
Generating database objects

To generate both the space of options and the decision tables within the following layer, RDL might be used. SAP's brand-new invention serves as a metalanguage that can generate database objects from entity relationship models. RDL can be used for two purposes:

- It can automatically create the data structures and objects in SAP HANA on which the arsenal of algorithms can run.
- Once something has been detected, RDL can generate (and fill) resulting decision tables.

Decision Tables

In Chapter 2, Section 2.2.2 and Chapter 7, Section 7.5 we touched on *decision tables*, referring to them as one option for moving decision logic from a product for data generation into SAP HANA. At the same time, modeling decision logic *within* SAP HANA via decision tables (instead of using relatively static and nontransparent procedures) is also an option that helps increase flexibility.



BUKRS	VTWEG	KUKLA
1000	10	01
	12	02

Property	Value
Alias name	KUKLA
Column Name	KUKLA
Data Type	NVARCHAR
Table Name	CE1IDEA
Schema Name	TRAINING
Type	Action
Decision Table Name	FRAUD_THRESHOLD

Figure 10.7 Decision Table in SAP HANA

Determine customer classes/ groups

Decision table FRAUD_THRESHOLD in Figure 10.7 uses a company code (BUKRS) and a distribution channel (VTWEG) from table CE1IDEA in Profitability Analysis (a component of SAP ERP) to determine customer class KUKLA. This customer class might later be used as an input parameter of a rule to detect suspicious transactions. Decision table FRAUD_THRESHOLD could be filled/maintained manually or via procedures.

Decision rules generated by procedures

A decision table assigns ACTIONS to a set of CONDITIONS. Such conditions might be values contained in another table's fields or values that are calculated from the contents of such fields (CALCULATED ATTRIBUTES). A decision table's structure is defined under OUTPUT; its contents are maintained in the area on the left-hand side of the screen. Once the table is activated, SAP HANA automatically generates a procedure to implement its decision logic. In our data model, we are assuming that the contents of decision tables are not maintained manually. Instead, the procedures described in "Procedures for Decision Tables" are intended to provide us with rules with which potential cases of fraud can be identified.

In return, this means that these procedures are expected to store the results of their considerations in decision tables or—alternatively—in statistical models, such as decision trees. In the first case, we are only dealing with decision tables; in the second case, decision tables contain links to models that are meant to be used.

If there is reason to believe that the behavior of suspects has changed, all you need to do in our data model is restart the algorithms listed in "Procedures for Decision Tables" (you may as well let them run continuously in the background) and thus update the contents of your decision tables. All layers above the decision tables are not affected; in the worst case, you may have to recalibrate your detection strategies (which can also be automated). This makes the proposed data model extremely adaptive while at the same time reducing the total costs of ownership (TCO).

How do you know whether the rules of the game have changed? Review Chapter 4 to refresh your memory.

Procedures for Detection Methods

Online or batch detection

Detection methods in SAP Fraud Management are based upon procedures written in SQLScript. Depending on whether you would like to use online or batch detection, different procedures are required:

- For *online detection* (detecting potential cases of fraud while processing transactions in your ERP solution), you need two procedures:
  - Execution procedure
    The execution procedure contains the core logic for detecting cases of fraud. With the SCN example (related to purchasing), the execution procedure would check whether deviations among ordered, delivered, and invoiced quantity are higher than the defined threshold.

  - Mapping procedure
    The mapping procedure links the database (such as the transaction data in product for data generation data) with the data required by the execution procedure. It also makes sure that data needed by the execution procedure are made available in the required format.
- With *mass detection* (detecting potential cases of fraud in batch), you also need two procedures:
  - Execution procedure
    See online detection.
  - Selection procedure
    With online detection, you are always looking at one single detection object at a time, whereas with mass detection you need a selection procedure to select the detection objects that need to be checked by the execution procedure.
- If a detection method is meant to be used in both online and mass detection, you will have to define all three procedures:
  - An execution procedure
  - A mapping procedure
  - A selection procedure
- In addition to these, you can also (optionally) create an additional *information procedure* and integrate that procedure into your detection method. Information procedures are used to determine additional values, such as the money at stake (what is your financial risk if a certain transaction is fraudulent?).

The preceding considerations have shown that detection methods in SAP Fraud Management are based upon two to four procedures written in SQLScript. The special feature of our data model arises from the fact that these procedures do not contain all the logic but instead use decision tables maintained by other (pattern-detection) procedures. This leads to a model that has the ability to learn (without human intervention), which implies that we are also going a bit beyond the classic idea of the learning organization, in which learning processes still require lengthy changes to business processes and (potentially) endless discussions. What we are after is learning systems instead. We all know that business is often about the survival of the fittest (which in turn frequently means the survival of the fastest learners).

[+]

(Partially) Dynamic Detection Rules in SAP Standard

In SAP's online help for SAP Fraud Management, you will find an example of how detection methods can be made dynamic with very little effort. Just search for the section titled "Predictive Models using the Predictive Analysis Library (PAL)." The approach described there is based upon the option to connect a detection method with a decision tree defined in SAP PAL. This decision tree model can be retrained whenever required.

You may, however, find that our approach (which is not only based upon SAP PAL decision tree algorithms but can theoretically use every statistical model one can think of in SAP PAL, R, or other systems) is far more flexible. In SAP's standard approach, detection procedures might be parameterized; in our approach, they are based upon decision tables that can contain dynamic links to every kind of statistical model.

Detection Methods, Detection Objects, and Scores

One score per detection object

As described in the previous section, detection methods largely consist of a couple of procedures plus (procedure-related) input parameters that determine the procedure's behavior. A detection method reads the data of detection objects (transactions to be checked) and returns a score—that is, a figure that describes the probability of fraud for that specific detection object. In that respect, our model does not differ from common standard approaches in SAP Fraud Management.

Detection Strategies

Generating an overall score

A detection strategy comprises several detection methods and therefore needs to aggregate the scores delivered by every single one of them using weighting factors. This leads to one overall score per detection object; if this overall score exceeds a certain threshold it will trigger an alert.

Calibration

Detection strategies can be calibrated manually or automatically. Calibration serves three purposes:

- Optimizing the input parameters of detection methods
- Fine-tuning weighting factors within detection strategies
- Determining threshold values

To do this, a specific combination of input parameters, weighting factors, and threshold values is applied to historical data (for which cases of fraud have been properly identified). On this basis, the system calculates how many real cases of fraud would have been detected with these settings and how many false positives would have occurred. The objective of calibration is to find those values that would have detected as many real cases of fraud as possible while at the same time producing no more than a few false alarms.

Maximum versus desired detection rates

Sometimes, instead of maximizing detection performance it might also make sense to try to adjust the number of detected or suspected cases to a certain level (and then use remaining optimization margins to reduce the percentage of false positives); in the end, you only have a limited number of investigators, and adapting the system's sensitivity to your needs might make more sense than being inundated with potential cases of fraud.

Calibrating detection strategies is standard functionality in SAP Fraud Management. Our data model does not differ from other approaches in that respect. Even automated calibration (working a bit like the Solver add-in in Microsoft Excel) is available in standard SAP.

Alerting

Investigators, managers, or auditors can be alerted via built-in reports within SAP Fraud Management. Suspected cases of fraud can be checked via the dashboards shown in Figure 10.4 and Figure 10.5. In addition, there are other paths for the fraud team, such as the Alert Worklist.

Trigger activities

Fraud management becomes even more efficient if—as a result of alerts in SAP Fraud Management—activities can be triggered in other systems. We previously mentioned payment blocks in products for data generation (ERP solutions). In online detection, SAP Fraud Management communicates with other applications via XML. Hence, products for data generation (whether they come from SAP or not) can send XML requests to SAP Fraud Management and will also get a response via XML. This makes it possible to not only generate reports but also integrate systems of whatever shape or form as clients. The top level in our data model (called ALERTING) not only consists of reports but can also represent communication channels via which messages are sent to other applications.

Conclusion: flexible layered model

In this chapter, we have tried to present a data model for fraud management. In comparison to common standard approaches, the model has been extended to add extra flexibility and to use the new potentials of an in-memory database like SAP HANA effectively.

Even without our attempt to broaden your horizons, SAP Fraud Management on SAP HANA will beat a lot of other products (using static rules) in terms of flexibility and performance. Furthermore, even if you are not trying to build a self-learning solution the simple fact that all relevant data sit in one consistent SAP HANA database will open the door to analyzing your transactions (that is, your detection objects) using a variety of instruments, such as SAP PAL or R, to (manually) develop new detection methods and detection strategies.

What we tried to convey, however, is that standard features, such as automatically calibrating detection strategies, are only the tip of the iceberg. Once you are able to not only detect fraud in real time but also to automatically have your rules adapted as you go along, fraud management becomes really tantalizing.

Sticking with the iceberg metaphor, if you do not want to get into maritime distress, then it might make sense to think about what might be lurking below the surface. If you don't, then your competitors might do so before you.

There are many intelligent species in the universe. They are all owned by cats.

Anonymous

11 Automating Service-Level Management

Derek thoughtfully gazed into the pair of deep blue eyes at the foot of his lounge chair. He had briefly looked up from his e-reader, reached for the wine glass, and noticed that Gina's cat eyes, though almost closed, were still fixated on him; could she be reading his innermost thoughts?

The feline psyche had always been a riddle to him. With the networks and computing clusters that occupied the majority of his waking hours, everything had its logical explanation. One pressed a button or fired off a command, and the intended result occurred. Well, usually. If not, there was always a logical reason. With Gina, things were different. Her behavior seemed to contradict all cause–effect relationships; it could be observed but never really explained.

The text on his display was exploring the brave new world of big data, discussing the paradigm shift brought about by new analytical possibilities. Why *had become meaningless; only* what *counted now in many areas. Understanding why A caused B was interesting, even reassuring, but most organizations had neither the time nor the budget to think about such things anymore.*

Well, fair enough, human knowledge has never been that impressive, has it? Despite the fact that Adam and Eve had nibbled at a fruit from the forbidden tree of knowledge, godly wisdom was still denied to their offspring. Quite a while later Isaac Newton discovered the law of gravity—once again in the shade of an apple tree—but even now, no physicist on the planet really knows the cause of gravitation. We know what it does in great detail, and how to *apply it to our advantage, but why it works? That's still the subject of great debate among great minds. Despite this, we all rely on Newton's law in our everyday lives.*

Derek grabbed the MaltSoft tube that was lying next to his glass, put a little bit of the brown paste onto his right index finger, and held it invitingly in front of Gina's nose. Normally, she went crazy for it, but today it didn't seem to be worth the effort of lifting her head by even a couple of millimeters. Sighing, Derek wiped the greasy substance from his finger. Empirical research clearly showed that he had a 73% chance of being purred to sleep by her later on, but he was still completely in the dark about why, on 27% of nights, he had to fall asleep alone. He had absolutely no idea what was driving Gina's decisions in that respect.

Figure 11.1 Gina

Dependencies among data

In most of the previous case studies, we have been working with dependencies among data. We have looked at which weather phenomena might have an impact on travel times, how customers might react to price changes, or what patterns of body functions or vital signs could help us forecast a health crisis. In all of these examples, we have underlined that by *dependency* we mean correlations rather than cause–effect relationships.

Assumptions are hard to shake off

Despite the fact that we have been trying to work inductively, all the algorithms we have presented are still constrained by certain prerequisites or assumptions. Although many of them—like multiple regression—are able to link more than just two parameters and can even detect nonlinear relationships, top-notch big data applications are trying to go one step further. They are not only trying to supply parameters for a model but also figure out which type of model should be used in the first place.

Improve performance in IT outsourcing

In this chapter, we will talk about an area you might be familiar with, either from a customer's perspective or by simply looking at your PC at home: What are the parameters driving the performance of complex IT systems? How can the providers of outsourcing services ensure that they will get certain tasks done at clearly defined levels of service? Where do they have to put in the most effort to get the best performance improvement for their dollar? At home, this question may boil down to buying either a bigger hard drive or a new graphics card; in a professional environment, this is called *service level-management*—that is, determining which resources (machinery and humans) are required to process certain amounts of data in order to guarantee specific response times.

[«]

Service Level

A *service level* is a quantifiable parameter used to measure the quality of a service. Service levels not only play a role in IT but also can be defined for each and every type of service. They are, however, especially important in IT support and IT and business process outsourcing (e.g., payroll, accounting, facility management, or maintenance).

One key challenge with service levels—as with value drivers or scores—is measuring and aggregating values.

When outsourcing services, the expected service level is usually defined in service level agreements (SLAs), which have been key to standardizing services and have therefore driven globalization within service industries.

Service levels and global competition

Service levels have risen massively in importance in the last couple of decades. Due to technical progress and the standardization of IT services and process-related services, both manufacturers of physical products and service providers are finding themselves in a global rat race, directly competing against opponents anywhere in the world. Being cost efficient has become the game-winning factor for service providers in IT and elsewhere.

Unfortunately, although sizing and costing SLAs for IT systems and related organizations is often no more complicated than doing the same for manufacturing plants and production systems, estimating the types and number of servers, networks, databases, applications, and staff needed to run them can become a massive task. This is because the challenge is newer, so there are fewer tools and algorithms to deal with it, and the underlying technology keeps changing, so every heuristically approach is out of date after a short while.

Hardware sizing: deductive approach not suitable

We are going to explain the problem by using a business process outsourcing example, first talking about hardware sizing and explaining why a purely deductive approach will not get you very far. After a couple of business-related and financial considerations, we will discuss what a suitable big data solution might look like, one that helps you pick the best of a number of alternative modeling approaches.

Then, for one last time, we will look at appropriate products from the SAP world, potential benefits, and suitable implementation scenarios and architectures. As in other case studies, we will look beyond this particular scenario; ultimately, we are trying to detect dependencies without knowing what kind or type of model to apply. This means extending the brute-force approach from Chapter 6. In Chapter 6, we used brute force to build and select decision trees; here we will use the same method to decide whether decision trees or something else should be used. This chapter is not about selecting one of many outcomes of an algorithm or about fine-tuning the algorithm's control parameters but about deciding which class of algorithms to use.

11.1 IT-Related Services as a Commodity

Remote services

Once-lagging emerging economies, such as India and Malaysia, owe their entry into the premier league of technology nations to recent leaps in information-processing and communication technologies. IT-related services such as programming, project execution, or first-, second-, and third-level support can be delivered from almost any location in the world (as long as that location has fast Internet access). When solving a technical problem, it makes little difference to a user whether a colleague from next door is looking over his shoulder or whether somebody based in Bangalore is accessing his PC or laptop via TeamViewer.

Standardized customer requirements

The Internet, fast networks, and new software have clearly contributed to these developments, but the fact that another kind of revolution took place in parallel often goes unnoticed. This is the standardization and industrialization of services in IT. Globally accepted standards and best practices such as ITIL (a collection of recommendations for IT service management) or PRINCE2 (a project-management method) have been fundamental in creating the foundation from which emerging economies have been able to exploit their cost advantage. One of the most important processes in ITIL is service-level management.

11.1.1 IT Services and Business Process Outsourcing

Standardized evaluation systems

Anybody specifying or purchasing services today is doing so on the basis of clearly defined service levels. Operational parameters are defined in contracts and then continuously measured in production later. Using these service levels, buyers can compare the offerings of various domestic and foreign suppliers and specify penalties to be applied if there are any major deviations. This is all based on the assumption that service levels measure the right things; unfortunately, this is not always the case. You will sometimes end up comparing the prices of apples with oranges, ending up with cheaper, but maybe less tasty, fruit.

Soft factors

Prices can easily be measured, but for many customers soft factors, such as a certain partner's reputation, are still important. Penalties become meaningless if claims cannot be enforced in foreign courts or if the other party hides behind some dodgy local Chapter 11 escape clause on the other side of the world. Furthermore, Swiss banks or their customers might feel uncomfortable about the idea that their customer's data or their management's emails are stored on Chinese, German, or—possibly the worst case—US servers.

Offshore's reputation has improved

Outside of these rather sensitive business sectors, the reputation of offshore service providers has improved quite a bit. Indian newcomers such as Cognizant, Infosys, Tata Consultancy Services, or HCL Technologies are on a level playing field today with established providers such as Accenture, IBM, or TDS (who, in turn, also employ offshore resources to service their clients). As a result, many services have turned into commodities. When deciding to whom to outsource processes, the cheapest supplier often beats the rest of the field. Purchasing IT-related services is therefore no longer much different from purchasing bulk goods, such as steel screws or plastic pellets.

11.1.2 Customers and IT Service Providers Speak Different Languages

Problem: defining requirements

Despite all standardization efforts, there are still significant differences between buying raw materials and outsourcing business processes. When purchasing screws, customers don't usually struggle to translate their requirements into technical specifications. They will have a lot of experience with different materials, and they will know which alloys with which strengths can be used for this particular function or product. In IT, however, most customers find it a lot harder to convert their process-specific demands into technical details.

Customers: business performance indicators

A customer planning to outsource the operation of a customer relationship management (CRM) application, his complete SAP Business Suite, or all processes in material management to an external partner is probably interested in the following parameters:

- How long is it going to take for a caller's master data or his track record to pop up on a call center agent's screen?
- How many customer master data records can be cleansed per month?
- How much time will pass between creating a ticket and the responsible support person responding to the call?
- On average, how long will it take to solve technical problems?
- How satisfied are employees going to be with the external provider's services? What can they do about it if they aren't?
- How exact are forecasted material requirements going to be, and how much capital will be tied up in the finished goods warehouse?

Providers: technical performance indicators

A provider of IT services will instead look at parameters that are easier to manage from his perspective:

- How many servers with how many CPU cores are we using?
- How much main memory do we need? How will L1, L2, and L3 caches (buffering data between the processors and main memory) be sized?
- What network speeds, capacities, and data-transfer rates are needed?
- Which solutions will we use to ensure data quality?
- How many people and how many telephone lines are we going to need in support?

Certainly, many of the technical performance indicators will have an impact on the business-related ones, but neither customers nor service providers will be able to tell for certain whether adding a few CPUs or adding a few gigabytes of memory would work better in terms of improving response times by 10% nor will they be able to specify how many CPUs or how much memory would be needed.

11.1.3 IT Systems are Complicated and Complex

Oh no, not statistics again!

Not only customers but also experienced IT experts struggle for air when trying to derive the service level of certain processes from technical parameters. Sometimes—if you are lucky—there are statistical algorithms that can do the job. The Erlang formulas (the Erlang-B formula and Erlang-C formula), which are based on the Erlang probability distribution, can be used to estimate how many call center agents are needed to ensure that only a maximum of 10% of your customers will have to wait more than 10 seconds for their calls to be answered by a real person. The R package `queueing` provides you with many more queueing models that go far beyond these relatively simple (well, simple for mathematicians) Erlang formulas.

However, such requirements are (relatively) simple special cases. With services supported by systems, service levels don't usually depend on one single factor or influencer. Systems consisting of many hardware and software components are not only complicated but also complex.

[»] **Complicated versus Complex**

In day-to-day communication, the two terms *complicated* and *complex* are often used synonymously, but they do have very different meanings. A system, a problem, or a model are complicated if many different factors interact, making it difficult to understand the mechanism as a whole. If we knew more, such as all dependencies and cause–effect relationships, then we would be able to predict how a complicated system would behave.

In a complex system, the situation is different. Even if we knew everything about the system's components and their interdependencies, we still wouldn't be able to predict the behavior of the system as a whole. A complex system is more than the sum of its parts.

Whole more than the sum of its parts

Let's take an example from the automotive industry. Nowadays, a top-of-the-range car contains about 100 electronic control devices. Each one is based on pure logic, following clearly defined rules. Nevertheless, interactions between these systems can lead to unpredictable events on a more or less regular basis. When, for example, the driver switches on a couple of interior lights at the same time, sudden voltage fluctuations in the system's bus could lead to erroneous measurements of sensors that monitor the brake systems of the driving wheels. This could make the traction control kick in, reducing engine power or even initiating the braking of individual wheels.

Theoretically, such relationships can always be modeled and/or predicted. In practice, however, you are unlikely to foresee all possible scenarios and combinations of events, let alone tell if and when they are going to occur. Ultimately this means that the failure probability of complex systems cannot be predicted using deterministic models but only via statistical calculations.

11.2 Scenario: Sizing an IT System

IT service provider SAP Tuk-Tuk

In Chapter 5, we were doing business with a service provider called SAP Rickshaw. As well as classic SAP outsourcing, SAP Rickshaw Holdings also offers the complete, end-to-end handling of classic business processing: business process outsourcing (BPO). For this purpose, SAP Rickshaw Holdings operates subsidiaries in a number of countries. Each of these subsidiaries has specialized in certain top-level processes. One of them is called SAP Tuk-Tuk (SAP TT) and focuses on material requirements planning (MRP), purchasing, and inventory control.

Service level with material requirements planning

A couple of months ago, TT received an order from a major German maker of toilet seats, which has a couple of plants in China, to optimize MRP for its production sites located there. Negotiations and figuring out which parameters to use for measuring service levels turned out to be extremely complicated. Although the customers insisted on minimizing capital tied down in working assets within the warehouses while maintaining a high ability to deliver, TT described its contractual obligations in terms of capacities (number of purchasing managers, servers, and software licenses). Finally, a compromise was found.

During the first exploration phase, the customer will still be responsible for the business side of the process. TT will be given a fully configured system complete with algorithms; this system will be built in Germany based on SAP Demand Signal Management (an SAP HANA-based solution to analyze sales data). TT will not develop its own specific forecasting algorithms during this first phase nor change or modify the ones supplied in any way.

Experts in the toilet seat manufacturer's German headquarters will be supplied with sales and sales forecasting data from their partners, which TT will process and use to generate sales forecasts, which will be sent back to Germany. Purchase orders will be created from these forecasts and released in Germany.

Later (after the first year), TT will be given the additional job of improving the quality of these forecasts (i.e., their exactness) by reconfiguring the forecasting solution (SAP Demand Signal Management) or adding another application to it. This second phase is out of scope for this case study.

Quality criteria with case study

To measure TT's service level, two criteria have been chosen:

- The forecasting quality of the sales forecasts delivered—that is, the deviation between forecasted sales and quantities actually sold as a percentage, which is measured monthly
- The speed of data processing—that is, the time passing between the arrival of new data and these data being taken into account with a forecast

Forecasting quality

Forecasting quality is heavily influenced by the algorithms used and how good these are at detecting patterns or outliers. We have dealt with topics like these in earlier chapters.

Speed of data processing

In this case study, we therefore focus on the second parameter: the speed of data processing. With software provided by TT's customer (the maker of toilet seats) and the data volumes given, the service level TT is able to supply will primarily be determined by the type, the sizing, and the configuration of hardware deployed. TT is free to select its hardware vendor, number of servers and CPUs, and the configuration of these machines—for example, in terms of main memory (within the limitations defined by SAP, stemming from the fact that SAP Demand Signal Management is running on SAP HANA, which is an appliance).

Data acquisition out of scope

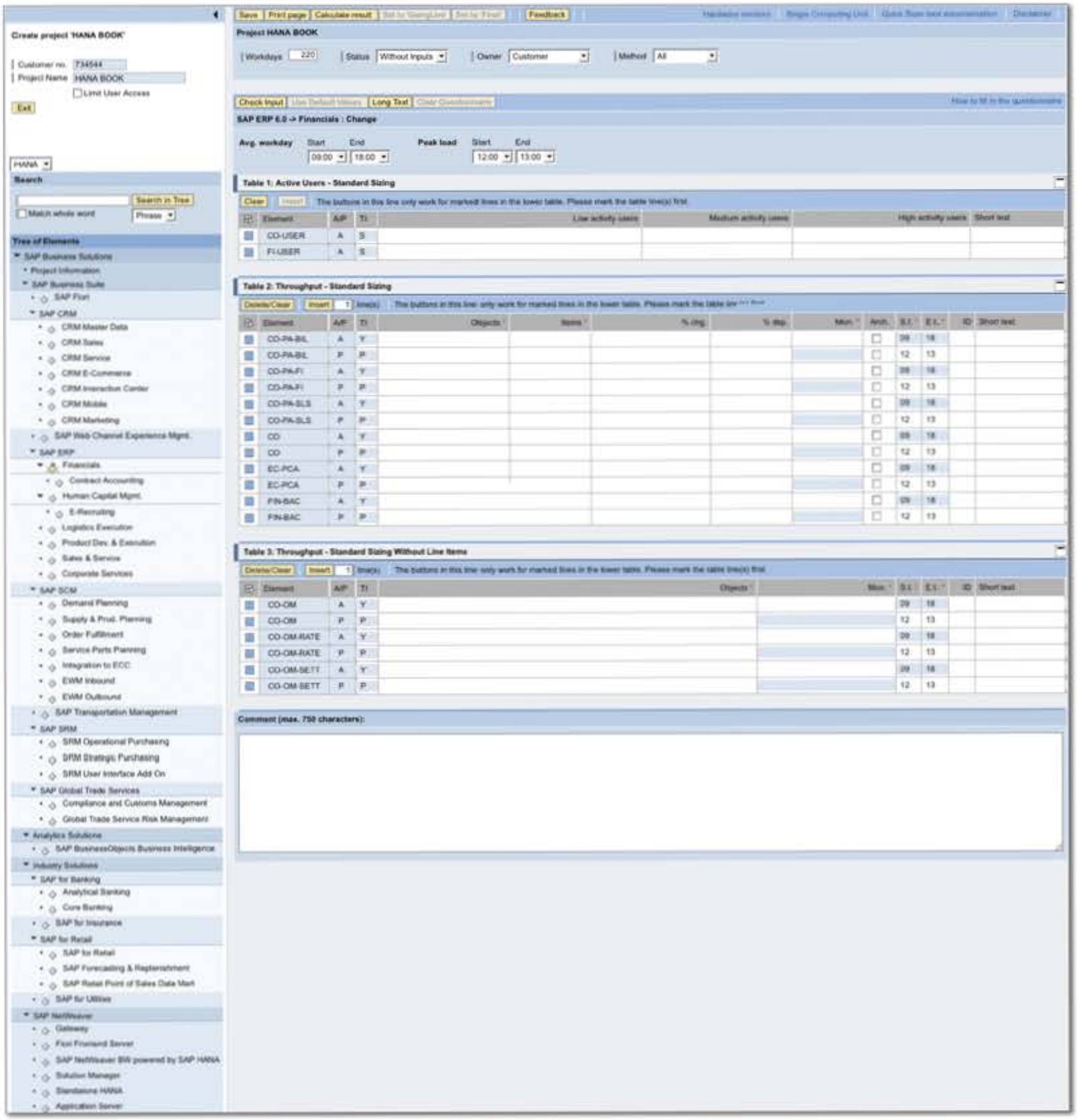
TT is also not responsible for obtaining sales data. These data are acquired from the POS systems of sales partners in China by a Chinese mobile phone provider and are pushed into TT's systems. Delivering data to Germany is also not within TT's area of accountability. The decisive timespan for measuring TT's speed of data processing is the time passing between the delivery of 10 new data records at Chiang Mai (timestamp of the last one) and TT making available a new sales forecast on one of their servers. A sales forecast not only consists of one single number but of the clearly defined set of lists (reports), diagrams, and dashboards. Picking up the data from this server is the customer's responsibility.

11.3 Sizing: Costs, Risks, and Opportunities

Sizing IT environments is a problem that has been a pain in the neck of CIOs and consultants since the early days of IT. With new solutions, hardly anybody knows what kind of computing power or memory will be required, and even if preliminary tests with an application have been performed before (as in our case) it will still be extremely difficult to estimate the marginal benefit of additional CPUs compared to the marginal performance increase that can be gained by adding 10% more main memory.

Quick Sizer not yet SAP HANA-oriented

SAP offers a tool to help you dimension an SAP HANA environment, the so-called SAP Quick Sizer (*http://service.sap.com/quicksizer*), but for SAP HANA the scope of this product is still limited. The SAP Quick Sizer's classic version (*http://service.sap.com/qs*) doesn't contain any information about SAP HANA-based systems; for SAP Demand Signal Management on SAP HANA, for example, you would have to use its SAP HANA-specific version (*http://service.sap.com/hanaqs*), which did not cover all SAP HANA-based solutions as of the time of writing.



Create project 'HANA BOOK'

Customer no. 734544
Project Name HANA BOOK
□ Limit User Access

HANA

Project HANA BOOK

Workdays 220 | Status Without Inputs | Owner Customer | Method All

SAP ERP 6.0 -> Financials : Change

Avg. workday Start 09:00 End 18:00 | Peak load Start 12:00 End 13:00

Table 1: Active Users - Standard Sizing

The buttons in this line only work for marked lines in the lower table. Please mark the table line(s) first.

Element	A/P	TI	Low activity users	Medium activity users	High activity users	Short text
CO-USER	A	S				
FI-USER	A	S				

Table 2: Throughput - Standard Sizing

1 line(s) The buttons in this line only work for marked lines in the lower table. Please mark the table line(s) first

Element	A/P	TI	Objects	Items	% chg.	% dp.	Mon.	Arch.	S.t.	E.t.	ID	Short text
CO-PA-BIL	A	Y						□	09	18		
CO-PA-BIL	P	P						□	12	13		
CO-PA-FI	A	Y						□	09	18		
CO-PA-FI	P	P						□	12	13		
CO-PA-SLS	A	Y						□	09	18		
CO-PA-SLS	P	P						□	12	13		
CO	A	Y						□	09	18		
CO	P	P						□	12	13		
EC-PCA	A	Y						□	09	18		
EC-PCA	P	P						□	12	13		
FIN-BAC	A	Y						□	09	18		
FIN-BAC	P	P						□	12	13		

Table 3: Throughput - Standard Sizing Without Line Items

1 line(s) The buttons in this line only work for marked lines in the lower table. Please mark the table line(s) first

Element	A/P	TI	Objects	Mon.	S.t.	E.t.	ID	Short text
CO-OM	A	Y			09	18		
CO-OM	P	P			12	13		
CO-OM-RATE	A	Y			09	18		
CO-OM-RATE	P	P			12	13		
CO-OM-SETT	A	Y			09	18		
CO-OM-SETT	P	P			12	13		

Comment (max. 750 characters):

Figure 11.2 SAP Quick Sizer (SAP HANA Version)

Online information about sizing SAP HANA appliances

On SAP's website for SAP HANA, there are quite a few documents that address hardware sizing—for example, for an SAP HANA-based SAP BW system (*http://www.saphana.com/docs/DOC-2114*) or for SAP Business Suite on SAP HANA (*http://www.saphana.com/docs/DOC-2933*). Although informative and enlightening, none of these documents are sufficiently comprehensive to help TT reliably size an SAP HANA appliance for SAP Demand Signal Management (see Figure 11.2).

11.3.1 Problem: Complexity Makes Modeling Difficult

Sizing remains case specific

There is a good reason that neither SAP nor its partners like to pop their heads above the parapet when it comes to hardware sizing: the dependencies among the number of CPU cores, main memory, or cache per CPU core are more than just a bit vague. Sometimes increasing the number of processing cores by 10% will give you more performance improvement than doubling the working memory, but at other times it is the other way around. Furthermore, quite often a system's performance doesn't change linearly but exponentially or by large, variable increments.

Clock rate often meaningless

Look at the advertising for personal computers, and you will see that manufacturers like to make a show of high clock rates while at the same time being stingy with (expensive) main memory. Nevertheless, the performance of many applications we use at home suffers much more from the frequent swapping of data to the hard disk than from a low clock rate. If the data volume that TT has to work with suddenly grows due to the shopping extravaganzas taking place around the Chinese New Year, for example, then TT may suddenly suffer from similar effects—a classic example of complexity.

Trial and error with sizing

When sizing its new solution, trial and error might not be the worst approach TT could try. TT could start with a certain variant, try to identify bottlenecks, and work out which bottleneck would be cheapest to eliminate, step-by-step adapting its capacities to the customer's requirements. Suppliers such as Amazon Web Services (AWS) have prepared themselves for exactly that kind of requirement; they provide their customers with the option of starting small, extending the performance of virtual machines step-by-step. But regardless of whether TT plans to choose local, proprietary hardware or virtual machines with a service provider, they still have to make some basic decisions in terms of configuration and operational parameters.

11.3.2 SAP Solution Manager as a Sensor

IT Performance Reporting

SAP customers can use SAP Solution Manager as a tool for monitoring performance. Its IT Performance Reporting component collects data from the Computing Center Management System (CCMS), supporting the surveillance of system status and performance. To that extent, SAP Solution Manager serves the same purpose as a vehicle's sensors in Chapter 8 or noninvasive or invasive body sensors in Chapter 9. In this case study, however, we are neither trying to predict the risk of accidents nor to forecast crises; instead we want to estimate the marginal benefit of potential improvements to an existing system, trying to work out in which area an investment will give us maximum performance improvement for the dollar.

Run SAP like a Factory

SAP's own consulting teams recognized that need and responded to it quite a while ago with a concept called "Run SAP like a Factory." Unlike CCMS, this approach not only looks at technical performance indicators but also at business processes. Data that are related to the service level with business processes are stored and analyzed within SAP BW. Using a deductive approach, one tries to get to the bottom of problems, derive measures, execute these measures, and then work out whether the situation has improved.

Problem: One Factor Is Not Enough

Deductive approach in OCC

In Run SAP like a Factory, operational data are not combed through and analyzed automatically or impartially. Instead, experts use an operations control center (OCC) to hypothesize about dependencies using their existing knowledge. On the basis of these hypotheses, they then deduct (or speculate) about what root causes one should work on. Although this approach can lead to improvements, it relates to a systematic search like a test drilling in the middle of the desert relates to a proper seismic analysis. One could strike gold (or oil), but then again you might get stuck with local optima, not even guessing that there are much bigger treasures to be found elsewhere.

Law of diminishing returns in agriculture

Let us look at an analogy from agriculture. There is a dependency between the amount of fertilizer used per acre and per year and the resulting yields. On the other hand, gains cannot be increased at will by using more and more fertilizer; at a certain point, this will actually have a negative effect, oversalting the soil and spoiling it for future growing seasons. In economics, this is called the law of diminishing returns. Although there is an optimum amount of fertilizer for sustainably increasing the crop, this optimum amount might not necessarily lead to maximum short-term returns; indeed, exceeding this optimum amount might even decrease the yield.

Global and local optimum

Yields in agriculture don't only depend on the amount of fertilizer used. Solar radiation, temperature, water/rain, and cultivation methods play an equally important role, and the optimum combination of all these factors will lead to far higher returns than just trying to optimize the amount of fertilizer. Furthermore, many of these factors are interdependent. An extraordinarily warm summer may, for example, improve crop growth, but from a certain threshold onwards it also supports the reproduction of pests and makes plants more susceptible to diseases.

If you narrow down your view to just a few factors, you will only reach global optima with a lot of luck, if ever at all. The reason that we still tend to do this is because many of the factors, like the weather, cannot be controlled and are often not forecasted accurately. The dependency of various elements creates complexity, something our brains don't like.

Problem: Nonlinear Dependencies

When searching for relevant influencing factors to dimension IT systems, one could choose decision tree scoring. If you do so, you will face three problems:

- We know a lot about IT systems and their cause–effect relationships because we designed them, but because these systems are complex we can't predict their behavior in all cases.
- The architectures and technologies of IT systems keep changing rapidly; insights gained yesterday are outdated tomorrow.
- As noted, the respective dependencies are complicated and, even worse, complex. Simple chi-squared tests or linear regression models won't do.

Limits of linear regression

As indicated by their name, linear regression models are used to detect and describe linear relationships. You can, however, use so-called linearizing transformations to turn nonlinear relationships into linear ones and then be able to use multiple linear regression once again. The problem with this is that you need to know what kind of nonlinear relationship you are trying to detect or model. In the case of main memory, for example, you would have to know whether the number of gigabytes you need would depend on the square or the cube of desired response times. The conclusion is that TT's existing knowledge would not be sufficient to apply any of the algorithms discussed in previous chapters.

11.3.3 Business Considerations

Managerial strategies

From TT's perspective, TT can't get things right. Whatever initial hardware configuration the company chooses will lead to a performance that is either unnecessarily good (and thus too expensive) or cheap but not sufficient. The most important question from a managerial perspective is therefore the configuration's adaptability. If the environment has been dimensioned too sparingly, then TT's vendors are probably more than happy to sell them more boxes. But, if TT uses a sledgehammer to crack a nut, then it will probably be stuck with an oversized cluster or will only be able to return parts of it, incurring major losses. One of TT's main objectives will therefore have to be flexibility, either by appropriate clauses in the contract with its suppliers or via a completely virtualized and freely scalable environment.

Effects of under or oversizing

When being too conservative in terms of sizing, TT runs a high risk of having to pay penalties to its customers. Such penalties have to be taken into account in the respective business cases. If TT spends too much money on hardware, then it will incur high fixed costs that will have an impact on pricing and competitiveness. Regardless of whether TT's systems are too big or too small (either one is going to happen anyway), it will have to decide which control parameter should be adjusted afterwards, either getting maximum performance out of a certain investment or maximum savings without losing a lot of computing power.

Not all parameters continuous

In IT as well as in manufacturing, not all knobs provide continuous, smooth adjustments. Certain operational parameters cannot be set as if you had fine control. If you have ever added additional memory to your computer at home, you probably know that manufacturers assemble memory slots in such a way that upgrades are only possible in specific (major) steps; sometimes you have to throw away all the old memory chips, replacing them with bigger ones, because all slots are already occupied. The fact that certain step increments have to be observed and that you cannot add half a server also have to be taken into account in our scenario.

11.3.4 Conclusion: Better Approximation to Reality, More Flexibility

Model selection algorithm

For TT, we need some kind of model-selection algorithm that provides the following functionalities:

- TT needs to be able to test-drive as many potential linear or nonlinear dependencies among input parameters (number of CPU cores and main memory) and output parameters (speed of data processing) as possible. Ideally, TT's model-selection algorithm should be able to automatically combine various input and output parameters, testing them for all kinds of relationships.
- Possible constraints when changing operational parameters (for example, the step size when adding memory chips) have to be modeled as realistically as possible. Otherwise TT could end up with theoretical optima that cannot be implemented.

11.4 Solution: Data Linearization before Analysis

Actuals are always realistic

When measuring actual operational parameters, the prerequisite of only looking at implementable options is always fulfilled by definition; you cannot measure something that cannot be implemented. On the other hand, you still need to be aware of restrictions such as step size. If you aren't, then you can't create realistic simulations.

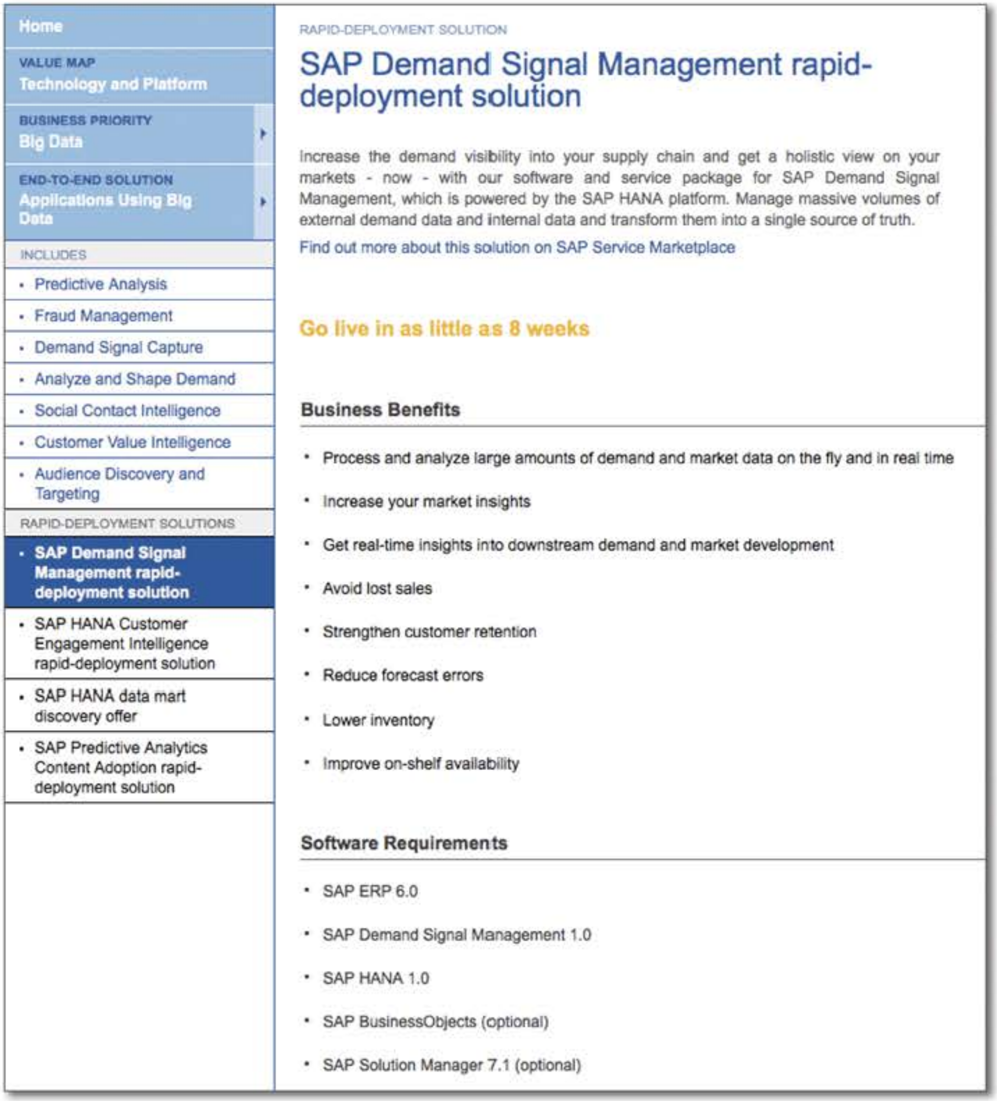
Easier than you think

Modeling as many potential dependencies as possible sounds more demanding than it is. The trick lies in the linearizing transformations, first mentioned in the "Problem: Nonlinear Dependencies" section. We will take a closer look at these, but before we do so let us once again, and for the last time, have a look at applicable solutions from SAP.

11.4.1 Related Value Maps in SAP Solution Explorer

SAP Demand Signal Management

The solution we mentioned earlier—SAP Demand Signal Management—can be used in more than one industry and can therefore be found in a couple of industry-specific value maps as well as in one cross-industry value map (Technology and Platform • Applications Using Big Data • Demand Signal Capture). In addition, there is a rapid-deployment solution (RDS) that might be of interest to TT (see Figure 11.3).


Home
VALUE MAP
Technology and Platform
BUSINESS PRIORITY
Big Data
END-TO-END SOLUTION
Applications Using Big Data
INCLUDES
• Predictive Analysis
• Fraud Management
• Demand Signal Capture
• Analyze and Shape Demand
• Social Contact Intelligence
• Customer Value Intelligence
• Audience Discovery and Targeting
RAPID-DEPLOYMENT SOLUTIONS
• SAP Demand Signal Management rapid-deployment solution
• SAP HANA Customer Engagement Intelligence rapid-deployment solution
• SAP HANA data mart discovery offer
• SAP Predictive Analytics Content Adoption rapid-deployment solution

RAPID-DEPLOYMENT SOLUTION

SAP Demand Signal Management rapid-deployment solution

Increase the demand visibility into your supply chain and get a holistic view on your markets - now - with our software and service package for SAP Demand Signal Management, which is powered by the SAP HANA platform. Manage massive volumes of external demand data and internal data and transform them into a single source of truth.

Find out more about this solution on SAP Service Marketplace

Go live in as little as 8 weeks

Business Benefits

- Process and analyze large amounts of demand and market data on the fly and in real time
- Increase your market insights
- Get real-time insights into downstream demand and market development
- Avoid lost sales
- Strengthen customer retention
- Reduce forecast errors
- Lower inventory
- Improve on-shelf availability

Software Requirements

- SAP ERP 6.0
- SAP Demand Signal Management 1.0
- SAP HANA 1.0
- SAP BusinessObjects (optional)
- SAP Solution Manager 7.1 (optional)

Figure 11.3 SAP Demand Signal Management RDS

SAP Solution Manager is key

The fundamental question for TT, however, is not (at the moment) forecasting demand but instead optimizing their own operational environment, and the data needed for this—operational data and service level–related parameters—can be found in certain components of SAP Solution Manager. SAP Solution Manager is not a product or a solution in the strict sense but rather a collection of tools, most of which are freely available. It comprises the following functionalities:

- Implementing and Upgrading SAP Solutions
- Solution Documentation
- Test Management
- Business Process Operations
- System Administration
- SAP Engagement and Service Delivery
- Incident Management
- Change-Request Management and Change Control
- Root Cause Analysis
- Reporting

Technical and business process monitoring

By monitoring technical components and business processes, SAP Solution Manager can deliver the data that TT needs. One advantage that is not to be underestimated is the fact that measurements from SAP Solution Manager can easily be passed on to SAP BW and then be analyzed there; for this, quite a bit of pre-defined BI Content is available. SAP BW on SAP HANA is therefore the ideal platform for our scenario. We will probably also need SAP Predictive Analysis (PAL) to exploit all these data.

11.4.2 Functional Requirements

Achieving SLA compliance

With the solution we are looking at here, we don't have to consider the underlying business requirements (demand planning); instead, TT is trying to tailor the IT facilities supporting these business processes to reach the objectives defined by their SLAs. TTs management has organized a workshop for this. In the course of this workshop, SAP Solution Manager and its components, as defined in Section 11.4.1, have been demonstrated to the project team. All other subsidiaries of SAP Rickshaw are already using SAP Solution Manager, and the holding company already runs SAP BW and SAP PAL on SAP HANA; TT could use that appliance for their purposes in terms of simulation and analysis. Taking into account the insights from the workshop plus existing knowledge, TT's board has decided on four key requirements:

- Provide data
- Define restrictions
- Freely combine input and output parameters
- Check data for dependencies

Provide Data

Provide data via SAP BW

The project team will be instructed to check, activate, and, if necessary, extend the BI Content for SAP Solution Manager mentioned toward the end of Section 11.4.1. All operational or process-specific measurements will be made centrally available within SAP BW.

Define Restrictions

Realistic approach

The restrictions mentioned in Section 11.3.3 and Section 11.3.4 are to be defined, but not yet modeled. In a later stage, they will be collected as semantically neutral metadata, similar to the semantically neutral metadata for sensors discussed in Chapter 8.

Freely Combine Input and Output Parameters

Operational data, service levels

TT plans to design an application that can generate all possible combinations of input and output parameters. As defined previously, input parameters are technical/operational measurements of the performance of their systems, and output parameters are linked to service levels.

If there are three input parameters, then there are $3! / (3! * (3 - 3) !) + 3! / (2! * (3 - 2)!) + 3! / (1! * (3 - 1)!) = 1 + 3 + 3 = 7$ options to combine them (A/B/C, A/B, A/C, B/C, A, B, and C). This number grows quickly with more input parameters; if you had 10, then you would already have 1,023 options; with 50, you'd have some *1.1 * 1015* or more than one quadrillion ways of combining them. We will list all of them in detail in our next book.

Check Data for Dependencies

Not only linear relationships

The combinations of input and output parameters generated will be checked for dependencies. First, TT would like to find out whether there are any such dependencies at all; second, they would like to find out whether such dependencies are meant to be modeled. From the management's perspective, it is important that TT doesn't only take linear relationships into account.

11.4.3 Building Blocks of the Solution

Based upon our considerations in Section 11.4.2, a potential solution for TT could be built using the components discussed in the following subsections.

Provide Data

ETL process within SAP BW

Data from SAP Solution Manager will be made available via an ETL tool. As long as only data from SAP Solution Manager are required, SAP BW's built-in ETL functionality should be sufficient and would be ideal due to the existing BI Content. If other sources of data have to be taken into account, then we could once again tap into the whole bandwidth of all the instruments discussed in previous chapters (SAP Data Services, SAP Event Stream Processor, and so on).

Define Restrictions

Intranet forum for brainstorming

Because the conditions and constraints mentioned in Section 11.3.1 are only to be collected but not yet modeled at this stage, TT has decided to try a pragmatic approach. All TT employees have been asked to contribute all restrictions they are aware of to a newly created forum on TT's intranet as free text. The entries collected within this forum are later to be analyzed together with other, external sources via text mining and to be converted, step-by-step, into a more structured format. As an incentive to actively participate, smartphones are raffled off among those who add entries.

Furthermore, TT is recording who has made which entry so that it will be possible to follow up or clarify things in detail when later structuring the data. Further activities in terms of extracting metadata similar to those described in Chapter 8 are not yet in scope.

Freely Combine Input and Output Parameters

Algorithms can only look at samples

As mentioned under the corresponding header in Section 11.4.2, the number of potential combinations quickly gets out of control as the number of input parameters increases. Although the algorithm to generate all possible combinations on the input and output side of things is mathematically trivial, it makes enormous demands in terms of computing and storage capacity.

Even though SAP Solution Manager can certainly handle more than 100 input parameters, even with the most powerful in-memory database the number of input parameters will have to be restricted to a more or less randomly generated simple selection.

Check Data for Dependencies

Calculate correlation coefficients

When searching for dependencies, we now use our magic wand, the linearizing transformation. Let's assume that we want to check whether t—the time needed to generate a sales forecast (the key parameter from the customer's perspective)—depends on the number of CPU cores c using the formula $t = 27 / c$; SAP Solution Manager then provides us with the three data records shown in Table 11.1.

Number of CPU Cores	Time Needed for Sales Forecast
8	9.54 s
16	6.75 s
32	4.77 s

Table 11.1 CPU Cores versus Processing Speed: Data Series

Although the numbers in Table 11.1 exactly represent the hypothesis set by the preceding formula ($t = 27 / c$), this is not a linear relationship. When analyzing these numbers by calculating their correlation coefficient, we don't get -1 but instead -0.96. -1 would signal a perfect dependency, and the number would be negative because the time needed to generate a sales forecast decreases instead of increases with a higher number of CPU cores. You may argue that the correlation coefficient is still pretty high and very close to -1, which should be good enough. However, this result is due to the relatively low number of data records;

if we had 500 values calculated on the basis of the preceding formula, the correlation coefficient for a linear relationship would drop to -0.60.

Correlation coefficient for linearized data

But don't worry, rescue is at hand. We can transform the values within Table 11.1, turning a nonlinear relationship into a linear one. The only thing we have to do is look at the dependency between the logarithms of both values. Using *log*—the logarithm for base 10—produces the numbers shown in Table 11.2.

log (c)	log (t)
0.00	1.48
0.30	1.28
0.48	1.19

Table 11.2 CPU Cores versus Processing Speed: Logarithms for the Data Series

The correlation coefficient for the preceding data equals -1 exactly, indicating a perfect (inverse) linear dependency.

[+]

Data from Other Subsidiaries

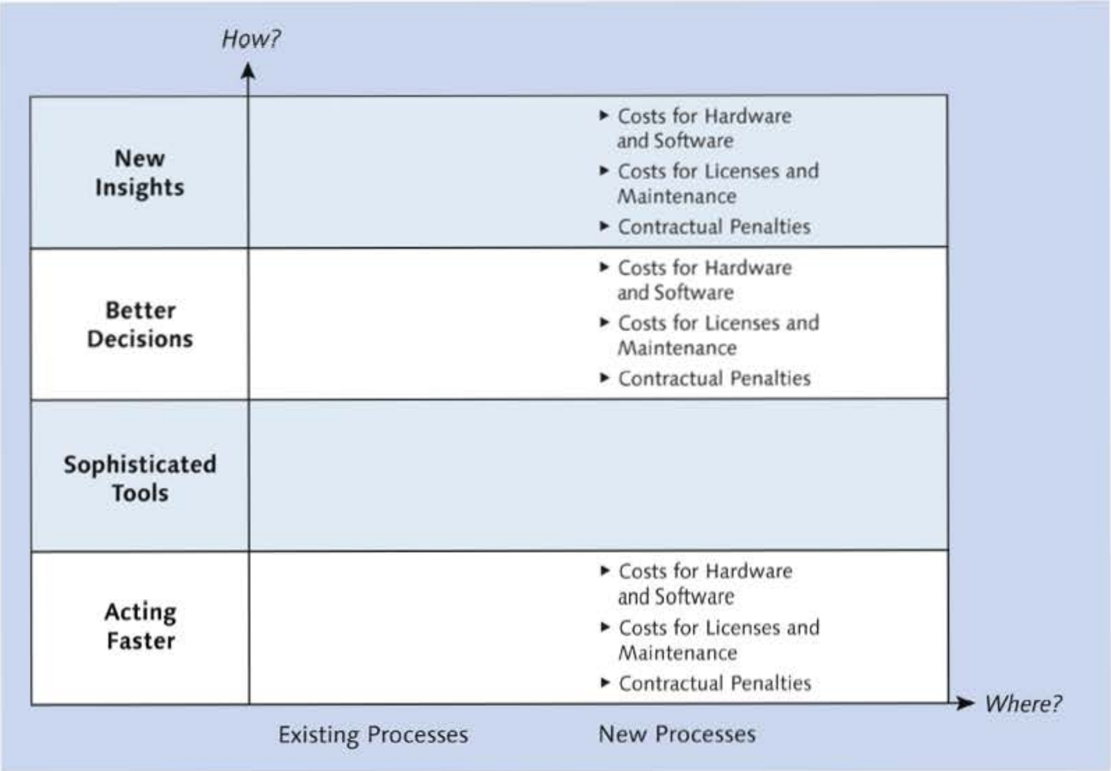
To begin with, TT will lack any kind of empirical data for research like that shown previously, but there is no reason that TT shouldn't use data series of other companies within the group. These other companies might use different solutions, and the respective response times might be meaningless for TT, but there are good reasons to believe that the dependency between certain operational parameters and the systems' performances in terms of response times will be the same.

11.4.4 Potential Benefits and Value Drivers

Using an inductive approach

In this case study, we are once again working with a new business process. By "new business process," we don't mean forecasting demand, for which we are only using a new solution (SAP Demand Signal Management) for an old problem. The innovation lies in the inductive and open-minded analysis of dependencies among operational parameters and service levels. Most approaches used in this area so far are based upon experience and deductive reasoning; this also applies to Run SAP Like a Factory.

SAP Solution Manager, for example, comes with a variety of dashboards. However, because certain key figures have already been selected for monitoring, decisions have already been made (based on someone else's preconceptions) about which operational parameters are relevant and could influence outcomes; consequently SAP Solution Manager doesn't give you all the variety and freedom that you need. You shouldn't look the gift horse of experience in the mouth, however, and there is no reason that TT couldn't use the preconfigured out-of-the-box tools within SAP Solution Manager; on the other hand, there also is no reason that TT shouldn't make a good thing better.



	Existing Processes	New Processes
New Insights		▸ Costs for Hardware and Software ▸ Costs for Licenses and Maintenance ▸ Contractual Penalties
Better Decisions		▸ Costs for Hardware and Software ▸ Costs for Licenses and Maintenance ▸ Contractual Penalties
Sophisticated Tools		
Acting Faster		▸ Costs for Hardware and Software ▸ Costs for Licenses and Maintenance ▸ Contractual Penalties

Figure 11.4 Service-Level Management Benefit–Value Driver Matrix

Lower costs and lower risks

As this approach to service-level management is relatively new, we have once again put all value drivers on the right-hand side of the benefit–value driver matrix (see Figure 11.4). Essentially, the desired effect on TT's shareholder value results from reducing investments in hardware and software and the support and maintenance fees often linked to these investments. Furthermore, TT can reduce the risk of having to pay penalties because they are not matching the customer's expectations. The SOPHISTICATED TOOLS row remains empty, because multiple nonlinear regression and transforming nonlinear data are not new concepts.

11.5 Implementation Scenario and Architecture with SAP HANA

Unlike in previous chapters, we are assuming that solutions from SAP only will suffice in this case study.

11.5.1 Implementation Scenario and Framework Architecture

SAP BW on SAP HANA

Because we don't need any data from ERP solutions and bearing in mind that there already exists rich BI Content for SAP Solution Manager, we are considering the SAP BW on SAP HANA scenario (see Figure 11.5) to be the variant that best suits our requirements; it is also the easiest one to implement.

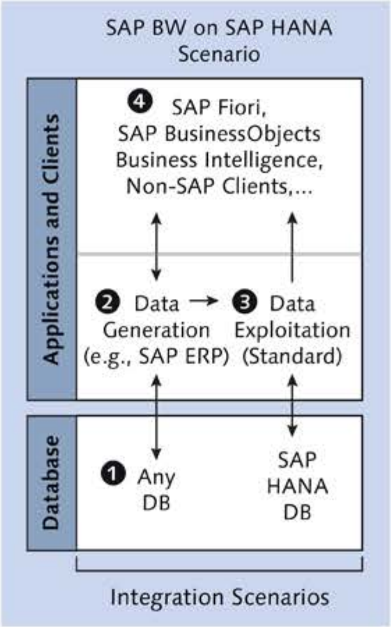

SAP BW on SAP HANA Scenario
Applications and Clients
❹ SAP Fiori, SAP BusinessObjects Business Intelligence, Non-SAP Clients,...
❷ Data Generation (e.g., SAP ERP)
❸ Data Exploitation (Standard)
Database
❶ Any DB
SAP HANA DB
Integration Scenarios

Figure 11.5 Service-Level Management Implementation Scenario

Databases ❶

SAP Solution Manager main source

SAP Solution Manager will be our primary source of data that we can tap into via standard SAP BW extraction. We therefore don't have to spend a lot of time thinking about non-SAP data sources and can simply assume that all data we need can be easily made available in our SAP HANA-based SAP BW. If this is not the case due to other sources of data being needed, we can use the data-logistics solutions mentioned in previous chapters.

Products for Data Generation ❷

No direct data flows

Our primary product for data generation is SAP Solution Manager. Its data are extracted into SAP BW and then land within our SAP HANA database. Our intention in this scenario is that there should be no direct flows of data between the databases. SAP BW gets its data from extractors, and no data are written back (called *retraction*) to SAP Solution Manager.

Products for Data Exploitation ❸

Linearizing data and building models

If we assume that all data we will use are first transformed/linearized, then we will probably only need SAP PAL on top of SAP BW. The transformation of data can be done via standard functions within SQLScript, and as long as we only want to use multiple linear regression models can be built via the `LRREGRESSION` function within SAP PAL. If the models that have been built are also to be used to make predictions—for example, to test the model's quality or to support simulations—then you will also need the `FORECASTWITHLR` function.

Clients ❹

Control panel to monitor model creation

At the client level, the solutions from the SAP BusinessObjects BI portfolio should be more than enough for our purposes. However, you may think about a special kind of user interface here. It is often a lot easier for humans than for machines to spot dependencies among data at a first glance. One could therefore imagine making nonlinear raw data available to a crowdsourcing community, asking the members of the community to come up with ideas for linearization. Furthermore, it would be possible to visualize automatically generated models, letting them undergo an additional, rigorous verification process (again via crowdsourcing). If TT wanted to use that option, it would have to differentiate between purely internal clients and clients that would have to be accessible via extranets.

11.5.2 Data Architecture

Figure 11.6 shows a proposed data architecture layout that embraces the two most important elements of our approach, namely using linearizing transformations and then multiple linear regression.

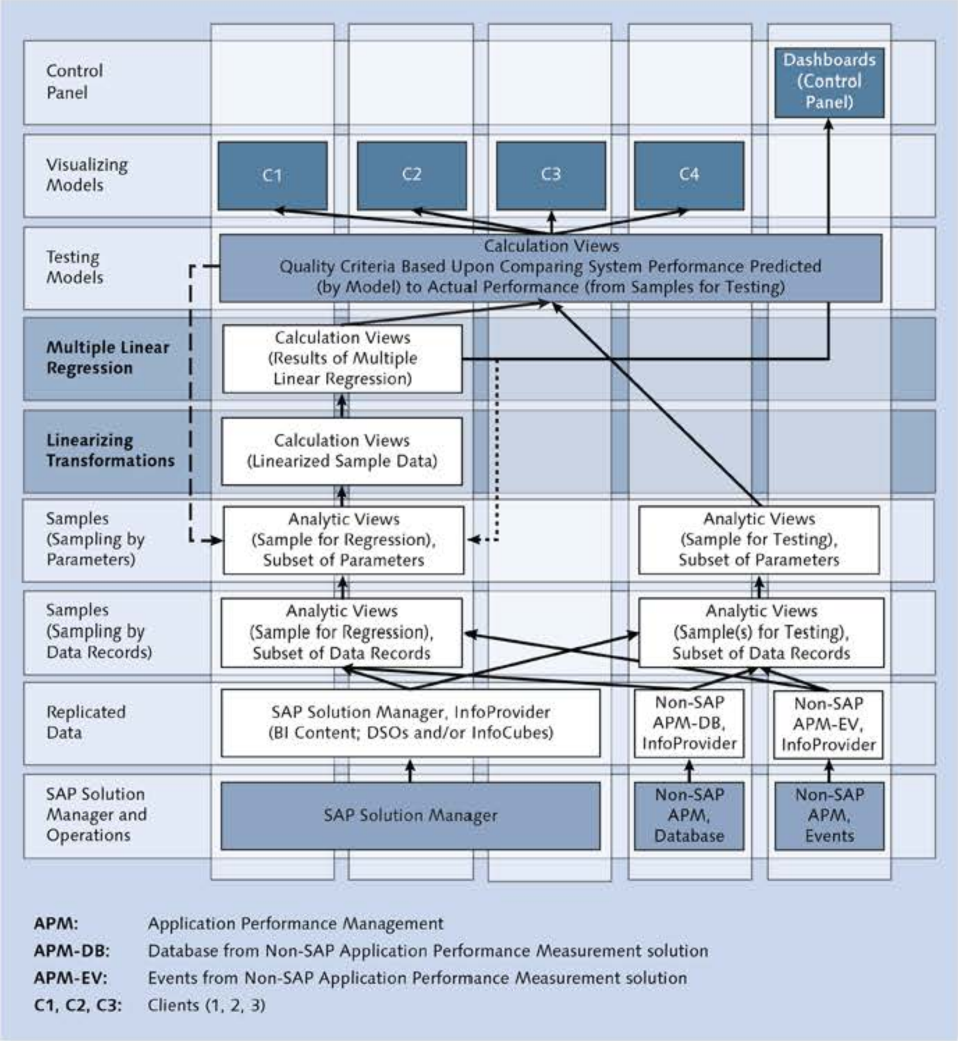

Control Panel
Dashboards (Control Panel)
Visualizing Models
C1
C2
C3
C4
Testing Models
Calculation Views
Quality Criteria Based Upon Comparing System Performance Predicted (by Model) to Actual Performance (from Samples for Testing)
Multiple Linear Regression
Calculation Views (Results of Multiple Linear Regression)
Linearizing Transformations
Calculation Views (Linearized Sample Data)
Samples (Sampling by Parameters)
Analytic Views (Sample for Regression), Subset of Parameters
Analytic Views (Sample for Testing), Subset of Parameters
Samples (Sampling by Data Records)
Analytic Views (Sample for Regression), Subset of Data Records
Analytic Views (Sample(s) for Testing), Subset of Data Records
Replicated Data
SAP Solution Manager, InfoProvider (BI Content; DSOs and/or InfoCubes)
Non-SAP APM-DB, InfoProvider
Non-SAP APM-EV, InfoProvider
SAP Solution Manager and Operations
SAP Solution Manager
Non-SAP APM, Database
Non-SAP APM, Events
APM: Application Performance Management
APM-DB: Database from Non-SAP Application Performance Measurement solution
APM-EV: Events from Non-SAP Application Performance Measurement solution
C1, C2, C3: Clients (1, 2, 3)

Figure 11.6 Service-Level Management

It also includes the idea of making certain diagrams available via extranets. As in other chapters, we will take you through the layers of the architecture step-by-step.

SAP Solution Manager and Operations

SolMan and other APM solutions

As usual, the lowest level of our data architecture contains our data sources. Our primary source of data is SAP Solution Manager (often also called *SolMan*). For the sake of completeness, we have also taken into account other similar application performance management (APM) solutions. For these solutions, we are assuming that their data are either stored in classic databases or are delivered as event streams. We are not expecting unstructured or weakly structured data.

Replicated Data

Replicating data into SAP BW objects

On the level above the data sources, you find replicated data, much like in Chapter 7, but because TT uses SAP BW we are assuming that these data—unlike in Chapter 7—are stored not in regular SAP HANA tables but in SAP BW objects, such as DataStore objects, for example, which are represented by analytic views in SAP HANA. The structure of these replicated data or their data models reflects—at least to a certain extent—the structure of the BI Content supplied by SAP.

Samples (Sampling by Data Records)

Subsets of the population (data records)

As we did in Chapter 6, we are going to develop and then test models, which is why we once again need more than just one sample from our population. At the beginning of Section 11.4.3, we extensively discussed sample sizing and sampling.

Here, however, we will have to select a sample using two dimensions: the data record and the fields within that data record. First, we'll have to follow the same process as before and select a couple of subsets from all available data records. This is what we mean by SAMPLES (SAMPLING BY DATA RECORDS) in Figure 11.6.

Samples (Sampling by Parameter)

Subsets of the population (parameters)

We will also have to sample by fields. In Section 11.4.2, we demonstrated that the number of theoretically possible combinations of input or output parameters quickly gets out of control, even when just considering 50 parameters, so we will not only have to look at a subset of our data records but also restrict our research to a subset of all parameters theoretically available within these data records. The size of this subset will be determined by TT's computing power and will therefore change over time.

To be able to draw samples on the basis of parameters instead of data record IDs, you will need a relatively abstract data model with a clear separation of content-related and semantically neutral metadata. If you have that, then you can use a set of random numbers to address parameters via their metadata object IDs. For further details about metadata objects and linking them to data objects, see Chapter 8.

After selecting the parameters you want to work with, you will have to prune your samples from the previous step accordingly. You will have to do this for each and every sample drawn by data records, which is OK, because all of these samples are identical in terms of the fields contained within them.

Linearizing Transformations

Algorithms for linearization

The input and output data still remaining within your samples after the two preceding steps could have simple linear or complex nonlinear relationships. If we want to rely on multiple linear regression, however, we should make sure that we are dealing with linear relationships only, but because we don't know the dependencies we are looking for we may not know which mathematical functions to use for linearizing our data.

Logarithm is often the first, but definitely not the only, option. For linearizing transformations, it may once again make sense to use trial and error or crowdsourcing. For a trial-and-error approach, you would have to define a portfolio of linearization algorithms; each algorithm would be represented by a procedure, and you would draw a sample of these procedures, applying the selected algorithms to some or all of your fields. In the end, we are therefore talking about not only two dimensions, but multiple dimensions when sampling.

Multiple Linear Regression

Standard function in SAP PAL

Multiple linear regression can be handled by one single function within SAP PAL (LRREGRESSION). All this function needs is a data matrix. If linear regression can't identify any significant dependencies (measured by the correlation coefficients), then you shouldn't proceed with building and testing a model and should instead stop.

Next, you should check for and keep partial dependencies, clean up your samples by removing some parameters and adding others, and start again with another linear regression run.

Testing Models

Check the model's quality

The newly built models can be tested in more or less the same way as described in Section 11.5; for linear regression in SAP PAL, you would use the FORECASTWITHLR function for this. However, unlike in Section 11.5.1, we are not trying to select the best one from a couple of structurally identical models.

Instead, we are trying to find the best modeling approach. Once we have done that, we will fine-tune the resulting model by changing the control parameters that were used when constructing it (as described in Section 11.5.1).

If none of the models looks promising, you would go back to square one, starting with new data records, new parameters, and new algorithms for linearization.

Visualizing Models

If a model is considered promising or conspicuous—whichever expression you prefer in your context—it should be cross-checked internally or externally (via crowdsourcing). To help investigators, your data artists should invent some kind of visual representation (perhaps a dashboard or a simple diagram), making it as easy as possible to judge the model's quality.

Schema for evaluating models

The assessment of models by humans should follow a standardized schema, and, as recommended for speech recognition and tasks executed by humans in Chapter 8, Section 8.5.2, it makes sense to present exactly the same models to different people. The results of these assessments by humans should be stored in the system and should be incorporated in the dashboards on the control panel layer.

Control Panel

Monitoring the solution as a whole

The control panel will not be used for evaluating individual models, the algorithms that helped build them, or their predictive power. It is instead provided to watch the functionality and the success rate of the system as a whole. Based upon the information from the control panel—which would normally consist of dashboards—one could fine-tune the portfolio of available input and output data, the supply of linearizing transformations, or sample sizes and sampling algorithms.

11.6 Conclusion

At the end of the last case study within this book we have to confess that we really see big data and SAP HANA as fascinating toys. SAP HANA might have been an in-memory database in its earlier days, but now it has become a very powerful platform that combines enormous computing power with access to highly sophisticated algorithms.

It is by logic that we prove, but by intuition that we discover.

Attributed to Henri Poincaré, French mathematician

12 Discovering Potentials, Designing Data Architectures

Matera didn't really look like a place in Europe any more. Technically, it was, but the lands surrounding the famous cave dwellings at Sassi were dry and dusty; on the rocky banks of the Gravina, only Garigue and Macchie were thriving, sprinkled with scattered, resilient succulents. The landscape seemed desert-like.

A wedding brought Derek here. One of his colleagues in Brussels fell in love with the hot-blooded daughter of a fisherman from Southern Italy and invited him to a wedding party that was likely to go on for about two weeks. After yesterday's long stag night, Derek was in need of some fresh air. Hunting for something interesting to photograph, he finally reached the Matera Gorge and found a strange supporting wall. The construction had made him smile. Most people in this area were fairly poor, but they certainly didn't suffer from a lack of ingenuity.

An almost vertical slope was rising before him, and the whitish-gray rock, made from volcanic ash so long ago, had been breaking up for some time. On the right, a big boulder held back the debris, and further to the right a concrete wall seemed to serve the same purpose. A resourceful landowner had closed the gap between these two supporting structures by piling up about two dozen old washing machines. It would take a while for their metal sheets to rust away in this dry climate, and until then the long-discarded appliances would serve their purpose well enough.

Derek remembered a lecture about big data that he had recently attended in London. The speech had been about the critical success factors when exploiting data gold. Until a couple of years ago, the limiting factor had been tech-

nical feasibility; today it was human creativity. A few years down the road, ownership of, or control over, data would separate the wheat from the chaff.

Figure 12.1 Supporting Wall near Matera, Basilicata, Italy

In this chapter, we will summarize some of the insights from our case studies. We will look at why quite a few organizations are still desperately looking for benefits from big data applications or SAP HANA and in which areas real competitive advantage can be found. We will also explain how to pick the right implementation scenario and why you will need completely new intellectual approaches when designing big data architectures.

War for data gold

We will finish this chapter and the book with a short look at future perspectives. In doing so, we are not going to focus on technical innovations in SAP's pipeline but on the rules that are going to define the war for data gold over the next couple of decades.

12.1 Speed Is Nothing but a Means to an End

Higher speed and better decisions

In Chapter 1, we emphasized that the potential benefits of big data solutions, and therefore of in-memory databases like SAP HANA, result from a quantum leap in terms of decision speed and quality. In the course of eight case studies, we have illustrated how to realize these benefits:

- Decisions that have been made by humans until now can instead be automated.

- Decisions that have always been made automatically can now be based upon a much broader and deeper range of data.
- Decision support tools that could only be used sporadically and by very large organizations in the past and that needed experienced mathematicians to operate them are now becoming components of common standard software. Consequently, a much larger group of customers will be able to utilize the respective algorithms.
- All your decisions are based upon assumptions about parameters and models now and in the future. As parameters and models change, it will no longer take you days or weeks to react; instead, you can make the required adaptions more or less immediately.

Speed is just the beginning

The extremely high performance of underlying systems like SAP HANA is not an end in itself, but just a means to an end. When looking for potential benefits, you need to work out how existing business processes can be redesigned or which new ones can be created on the basis of such technical possibilities. A sales person working with a major hardware vendor recently told us that many of his customers were still reluctant to dip their toes into big data. Many of them said that they had completely virtualized their IT environments over the last couple of years, so they believed that they had enough processing power and infrastructures that were extremely scalable. They just didn't see the need for an in-memory database or a big data appliance to give them more processing power.

From our perspective, this point of view ignores the essential benefits to be gleaned—the quality-based elements that can let you take a big step forward and deliver new business insights and benefits. Big data or SAP HANA isn't only about making your reporting 1,000 times faster or activating data within a DataStore object located in an SAP HANA-based SAP BW system in a hundredth of the time. High computing power is a prerequisite for using sophisticated mathematical, statistical, and decision-supporting algorithms, but without the libraries of SAP Predictive Analysis (PAL), without the integration of R, the link to event streams or storage solutions for weakly structured or unstructured data (HDFS), and without new, relatively abstract metalanguages like RDL, more speed would just lead to quantitative improvements but not to qualitative ones.

Brand-new design options

Therefore, we don't consider SAP HANA, or big data as a whole, to be some kind of tuning tool for reports meant to assuage the cries of distress from clerical assistants moaning about long response times. In top management, for example, minutes or seconds rarely make a difference in reporting. What really stresses a CFO who is in an airport's executive lounge and wants to look at sales figures on his iPad is more likely to be the waiting time between slides and troublesome Internet connections than reporting performance.

Although SAP HANA only represents a relatively small subset of big data (see Chapter 2, Section 2.1), the same considerations apply. Because SAP's new platform is well embedded both in SAP's own portfolio of products and in the overarching big data space, this creates major opportunities for fundamental improvements. Potential benefits are to be found wherever decisions can be made faster, with a better hit rate, and automatically instead of manually. Some organizations will grab this chance; for others it will pose a threat to their survival.

12.2 SAP HANA Implementation and Data Architectures

In all our fictitious scenarios, we provided some ideas about possible implementation scenarios. We have also thought about potential data architectures—not in detail but at a conceptual level. Our suggestions within the case studies can also be used to derive some more general rules and recommendations.

12.2.1 Implementation Scenarios

Decision matrix

In Chapter 2, Section 12.2, we have presented 10 implementation scenarios developed by SAP; these scenarios are assigned to three categories: Replication, Integration, and Transformation. Table 12.1 summarizes our thoughts about which scenario to select when implementing specific requirements.

Criterion/ Scenario	Data Mart	App	Content	Accelerator	Cloud on SAP HANA	SAP Business One Analysis	SAP Business Suite on SAP HANA	SAP Business One on SAP HANA	SAP BW on SAP HANA	New SAP HANA Apps
Reading from Databases	✓	✓	✓	✓	✓	✓				
Reading from Products for Data Generation									✓	
Writing into an SAP HANA Database					✓				✓	
Sending Data/ Messages to Products for Data Generation				✓						
Integration with SAP (ERP) Solutions			✓			✓	✓	✓		
Integration with SAP BW									✓	
One Consolidated Database (SAP HANA)							✓	✓	(✓)	✓

Table 12.1 Implementation Scenarios Decision Matrix

This table will help you make the right decisions about your own business cases. All criteria listed there address the same question: What kind of relationship or link do you need between your new SAP HANA-based product for data generation/exploitation and the rest of your infrastructure?

In the following sections, we will explain what is meant by each criterion within the table. A tick in one of the table cells indicates that the functionality specified in the respective row is required for your case.

Reading from Databases

External data sources

Your newly created application (for example, a product for data generation or data exploitation) is to be fed with data from non-SAP databases. The term *databases* is meant to be interpreted broadly here; it could refer to classic, relational databases, to persistent NoSQL databases, to other in-memory databases, or to event streams.

The key thing is that data have to be acquired from somewhere, usually from products outside the SAP world, and that these data have to be made available by either replicating them to SAP HANA or by creating virtual tables that refer to remote destinations.

Reading from Products for Data Generation

Listening to OLTP directly

Your application receives data directly from a product for data generation—for example, an ERP system. This is the case when extracting data from SAP Business Suite or other SAP applications via SAP's Service API to SAP BW, at least to a certain extent.

Some extractors within SAP BW access database tables, but many (and often proprietary ones) use function modules that were originally designed to be used within the sending application itself.

Extraction into SAP BW, however, is a special case that we used to explain the mechanism. One could think of many other scenarios in which SAP or non-SAP solutions send data directly to a new application built within SAP HANA.

Writing into an SAP HANA Database

Persistent storage

Big data applications are often thought of as products for data exploitation. Every now and then, they also generate data, which not only end up with clients or with a product for data generation outside of SAP HANA but which also have to be stored longer term within an SAP HANA database—perhaps to be used in the future by the same application that originally created them.

We are separating this requirement from a scenario in which all relevant data are stored together within the same SAP HANA database (by "all relevant data" we mean the data from a product for data generation outside of our application plus the data generated by our application). This latter case would correspond to One Consolidated Database (SAP HANA).

Sending Data/Messages to Products for Data Generation

Instructions for other applications

Sometimes your application will want to directly send data to a product for data generation or tell a product for data generation to do something—for example, to block or release certain payments. For consistency reasons, we wouldn't recommend that a product for data exploitation write directly into another solution's database, such as another vendor's OLTP system.

In such cases, it makes more sense for your product for data exploitation within SAP HANA to communicate with the other product for data generation, either telling it to do something or instructing it to write certain data into its own database. Whether the recipient is going to use these instructions at runtime or store them persistently doesn't matter for this criterion.

Integration with SAP (ERP) Solutions

Easy OLTP integration with SAP Business Suite

In Chapter 2, Section 2.1.3, we emphasized that its integration with the rest of the SAP world is one of the greatest trump cards of SAP HANA for SAP's existing customer base. If you are one of these existing customers, then using SAP ERP or data warehousing solutions such as SAP Business Suite, SAP Business One, SAP BW, or other SAP platforms and tools (such as SAP Solution Manager) may save you a lot of time and effort on integration. The scenarios ticked with this requirement provide you with integration tools for use with other SAP products.

Integration with SAP BW

Easy data warehousing integration with SAP BW

Migrating an existing SAP BW system onto an SAP HANA database is relatively easy, and if the data you are planning to analyze with a new product for data exploitation are already available within SAP BW such a migration makes a lot of sense. This consideration not only applies to data that are already held within SAP BW but also to any data that can be easily made available there—for example, due to existing objects within SAP BI Content. Data from SAP Solution Manager, as mentioned in Chapter 11, are one example of this.

One Consolidated Database (SAP HANA)

A tick in this row signifies that the data you are planning to use are already held within the SAP HANA database on which you are planning to implement your application. As a consequence, you will neither have to extract nor replicate these data.

For the SAP BW on SAP HANA scenario, the tick is in parentheses; the reason for this is that there are two subscenarios:

- Your SAP BW system is running on SAP HANA and also extracts data from solutions whose data are not yet in SAP HANA or are held in a separate SAP HANA database; in this case there is no database sharing, and the tick wouldn't apply.
- Your SAP BW system and the applications from which it processes data hold their data within the same SAP HANA database; therefore they share the same database, and there is no need for extraction. In this case, the tick will apply.

Redesigning your data architecture

In the first of these two cases, redesigning your data architecture sooner rather than later might be a good idea. In the long run, this will save you a lot of time when maintaining extractors and the associated structures.

12.2.2 General Recommendations for Data Architecture

In all our case study chapters, we selected an implementation scenario and also thought about data architectures. Despite the fact that all these data architectures address specific, individual cases, some general principles and rules have emerged. We are going to clarify and summarize them in the following sections.

Virtuality

Persistence is an exception not the rule

Data structures within SAP HANA tend to be less durable then data structures within SAP Business Suite or SAP BW for two reasons:

- Within a data warehouse LSA model, persistent objects are used to optimize reporting performance, but if response times are extremely short anyway such structures become unnecessary. In this context, one might speculate whether OLAP cubes in data warehouses are now nothing but a remnant of the past.
- In some of our case studies, intermediate results have an experimental character. You are conducting hundreds of thousands of tests, you are constructing many different decision trees, or you are testing a variety of theoretically suitable models. Most of these intermediate results are not going to pass the struggle for survival test, disappearing into meaninglessness just after they were conceived. There is generally no need to store these data using persistent objects.

After all, what does "persistent" mean?

For both reasons, quite a few data that in the past would have been stored persistently now only exist at runtime. For a solution in which *persistent* only means *held in working memory*, the difference between persistent and virtual objects becomes blurred anyway.

Transparency

Transparent, understandable data flows

Knowledge that is hidden in programs, routines, or procedures often remains undocumented and falls into oblivion; this becomes an almost insurmountable barrier that hinders change and innovation. Just remember the hype around the millennium bug and the effort that had to be invested to identify and fix problems in millions of lines of code.

If your customers change their behavior and if your responses to what your customers are doing are hidden somewhere in undocumented code, then your first problem will be identifying all affected routines (this is impact analysis; see Chapter 8, Section 8.4.4).

Abstraction

Unstable, ever-changing environments

The closer that your data structures are relying on current conditions or implicit assumptions, the more likely it is that the structures and your organization will go down with the ship if you cling to these conditions and assumptions. Booksellers didn't anticipate that even the most diehard bibliophiles would sooner or later see the advantages of e-books, and the music industry has underestimated the impact the Internet would have on its business model. Such misjudgments often go hand-in-hand with data structures that are no longer in line with the times. How important are CD sales for the listening habits of music lovers today? How valuable are reports and dashboards presenting CD sales figures in 2014?

Design independent of current situation

We have explained the fact that data structures tend to age with the environmental conditions under which they were created, using the example of frequent-flyer status levels. Why should a certain number of categories be optimal in the long run, and what empirical proof do you have that they ever were? The more independent your data worlds become from a certain snapshot of reality or from a certain idea, the easier detecting change and responding to it by adapting your data structures will be.

Semantic Neutrality/Process Orientation

Data models that rest on stable properties

Apart from abstraction in general, we frequently emphasized semantic neutrality. Data structures are not designed in a vacuum; you have to get your bearings from some kind of structuring criteria. However, we recommend using criteria that tend to be stable and not volatile over time.

One example of stable criteria are the levels of measurement defined in Chapter 4, Section 4.4.3. Some 40 years ago, album sales in record shops were a good measure of a band's success; a couple of years later—following the emergence and market penetration of cassette tapes and then the

Walkman—play counts on radio stations might have become a better indicator. Today, retrieval rates from streaming services such as Spotify might make more sense than anything else. The description, the content, and the data sources we derive this key figure from keep changing over time, but we are still talking about something that is measured on a ratio scale.

Value Drivers/Decision Alternatives

Keep value drivers in mind

We know from experience that the domains in customers' real live data warehouses (that is, in products for data exploitation) are structured on the basis of all kinds of criteria. Source systems, time characteristics, a company's current organization, business functions, or performance considerations may determine a data warehouse's horizontal structure; see Chapter 4, Section 4.4.3. We have yet to encounter a data warehouse in which data flows, layers, and domains are clearly related to value drivers and shareholder value.

In a relatively stable world, one may hope that reports generated via existing inflexible structures—which have demonstrably ensured the organization's survival so far—will lead to decisions that increase shareholder value. However, if data are now changing monthly, weekly, or even daily, rather than annually, then the assumption that management is clinging tightly to an illusion has a lot to recommend it.

Analytical structures should be based upon value drivers, supporting value driver-oriented decisions, and if those value drivers or the decisions to be made on the basis of them are subject to permanent change, then you will need structures that are able to keep up. In Chapter 6, we illustrated what such structures could look like.

Flexibility/Adaptability, Relocating Decisions

New definition for "customizing"

Using the example of pricing in Chapter 7, we explained that customizing as practiced in common OLTP systems and therefore also in SAP Business Suite no longer satisfies today's requirements in terms of the flexibility of automated decisions. About 40 years ago, SAP's decision to move decision logic from program code to customizing tables was a giant step forward; this concept might well have been key to SAP's enormous success.

Settings updated automatically in real time

Relocating settings that control processes to an in-memory database (that is, into decision tables within SAP HANA) is the next logical step. The crucial difference between these two scenarios is not that these settings are now held in main memory; the real revolution stems from the fact that the tables that contain the settings don't necessarily have to be maintained manually anymore. Instead, these tables can be updated automatically and in real time by algorithms. For existing SAP customers, this concept makes SAP HANA even more attractive than other in-memory databases, because SAP is also actively working on moving processing logic from the application layer to the database layer. From our perspective, only a few customers of SAP have yet to fully appreciate the doors (in terms of real-time customizing) that this may open in the future.

Openness for External Access, Integrating Crowdsourcing/ Web Services

When talking about speech recognition and extracting metadata in Chapter 8 and also when discussing service-level management in Chapter 11, we indicated that some things cannot be automated—not even with big data.

Open your systems to external experts

Nevertheless, quite a few crowdsourcing communities on the Internet have come into existence in parallel with the rise of big data; coincidence or not, these communities and their portals fit very well into this gap. Regardless of whether you are trying to detect simple trends by looking at diagrams or to develop highly complicated algorithms, there are already fast-growing forums dedicated to these areas on the web. To tap into this potential, you will have to incorporate some kind of openness with your big data solutions right from the beginning. This includes making certain views or user interfaces available via extranets or even via the Internet. You will also have to think about how to make training data for algorithms available without revealing confidential user or business data or without running into the risk of your data being deanonymized. Luckily, there are a lot of new approaches to anonymization. Data are no longer only transcoded but are also made hazy or blurred to protect the innocent (you and me).

Cyclical Structures

If necessary: back to square one

Some of our data models (like the one in Chapter 11) contain cyclical structures; at a certain point and under certain conditions, the process stops, starting again with an earlier step. Because it would be pointless to start comparing the quality of models if you already know that, for one of them, a third of the parameters have no impact whatsoever on its results or predictions, it would be better to reselect some or all of the parameters and loop back to a previous phase. The principles described under "Virtuality" and "Abstraction" make this very easy.

This doesn't mean that you will have to delay your decisions until the system has found the best of all possible solutions—the Holy Grail—the global optimum. Instead, you could make the solution use the best model found so far for immediate decisions and at the same time continue to improve a predictive model in the background. If a similar decision—for example, regarding the pricing of beer—has to be made 10 minutes later, then it will be made on the basis of an enhanced model and might therefore be different, even though all environmental/input parameters are the same as before.

Induction Instead of Deduction

Prejudice-free design

In Section 12.2.2, under "Induction Instead of Deduction," we stressed the fact that data structures should not be founded on unproven, arbitrary assumptions and prejudice. Deducting hypotheses from past experience and then starting to build structures for which implementation might take years and swallow up millions of dollars based upon nothing but guesswork doesn't seem like a good and workable concept to us. Instead, you should continue to question each and every existing structure (see Chapter 4), and if you have the chance to do so you should instead let the system derive structures from observable facts automatically.

A Wide Variety of Modeling Approaches

Best modeling approach subject to change

In the first seven case studies, we discussed models that are more flexible by magnitudes then anything you will find in most organizations (with the exception of algorithmic trading firms on stock exchanges). Nevertheless, we still focused on certain algorithms or types of algorithms in all these cases and therefore accepted restrictions regarding possible insights and outcomes.

Within the last case study, however, we lifted that restriction, showing that a big data application could calculate not only one type of model with different control parameters but also a whole class of models (multiple nonlinear regression). Naturally, this approach can be extended to more than one model class; it would, for example, be possible to let nonlinear regression and machine learning fight against each other for survival in two separate data flows.

12.3 Outlook: Fantasy, Creativity, Mindfulness, and Control over Data

Technology will continue to evolve

We have summarized the principles that our data models are based upon in this chapter, and we have also emphasized that we have not yet arrived at the end of the road. Technology will continue to evolve, and we will be able to handle data volumes we can't even dream of yet at unimaginable speed. At the same time, the concentrated creativity of global communities will produce algorithms that turn the visions of science fiction authors into reality soon. Software such as iTranslate Voice is not too far away from automated simultaneous translation and therefore resembles the idea of the Babel fish introduced in the quote at the beginning of Chapter 2 (and ten years ago, would you have dreamt of autonomous cars being a reality in 2014?).

The dark side of big data

At the same time, big data is giving rise to dangers unthought of a couple of years ago. Movies like *Minority Report* depict a dark future in which we are going to be punished for crimes that we may commit but haven't committed yet. Unfortunately, the dark side of big data is also more real than many of us may be aware of. In some countries, the decisions of parole committees are made on the basis of statistical analyses about reoffending probabilities. Personal fates are therefore no longer exclusively in the hands of other humans but are determined by machines/algorithms.

Even if the error rate of such algorithms is proven to be a lot lower than that of human decision makers, it still means that human beings are kept in prison just because a tin box considers the risk of this person committing a crime to be too high. In many cases, these algorithms are nontransparent to police officers or courts, thus potentially violating basic rights put down in the Fourth Amendment of the United States Constitution and the European Convention on Human Rights.

Regardless of whether you are going to focus on the opportunities or the risks of big data or solutions like SAP HANA, we are on the cusp of a quantum leap, which—like all human inventions—can turn out to be a curse or blessing.

Ideas are the bottleneck now

At the moment, the limiting factor when exploring brand-new realms for big data applications is not computing power or hardware costs. We need new ideas and more creativity to create exciting business cases, shareholder value, and real competitive advantage. Old thinking patterns that stem from the early days of information technology and data processing or reflect performance restrictions that no longer exist have to be jettisoned.

If we are open-minded and receptive to new ideas when designing new data models, then big data and SAP HANA might lead us to astonishing insights that were inconceivable in the past, rewarding us with more business and personal growth than ever before. With this book, we are hoping to make a small contribution to this.

Despite the fact that—as with all new inventions—creativity is key right now, it will not remain the main bottleneck forever. Some organizations are going to recognize the chances of new paradigms and build flexible and open structures that can handle whatever will come their way during the next decade. At the same time, these companies will try to get hold of as much data as they can, well aware that what seems useless today might become extremely valuable tomorrow.

Future success factor: control over data

Others are going to stay in the realm of the "known knowns," working with facts that look familiar to them. In a couple of years, these organizations might no longer be able to access the data they need and will have to pay a high price for buying them from others, maybe even from their competitors. Therefore, thinking about whether and where SAP

HANA could make sense in your environment is not a topic for think tanks or research departments or an issue that can be left to mature for another year or two.

It is a number-one priority for top management here and now.

The Authors

For more than 20 years, **Michael Mattern** has been supporting multinational organizations as a project/program manager, senior consultant, and enterprise architect, helping these companies reap the financial rewards of major SAP implementations. He graduated from Augsburg University (Diplom-Ökonom, focus: Operations Research/Statistics and Corporate Finance), and is a certified SAP Associate Enterprise Architect.

As an IT and management consultant, Michael has worked on projects with Accenture, Bayer, Bridgestone, British American Tobacco, Fonterra, Lufthansa, Nestlé, Swisscom, and Vaillant—among others.

His main areas of expertise are developing reporting and planning solutions, implementing cross-solution business processes, and designing flexible and scalable data models. Apart from his work as a consultant, Michael has also supervised training courses and workshops for SAP Education in Belgium, Germany, Switzerland, and the United Kingdom. Together with his wife and a couple of cats, Michael lives in Switzerland and Oceania.

You can contact Michael via LinkedIn.

Ray Croft started his life in IT back in the 1960s, at a time when developers had to program computers with a main memory of only 1 KB—less than the size of an e-mail today.

In the course of his career, Ray has worked as a developer, analyst, and system designer for leading companies like British Aerospace, Citibank, and RSA Insurance Group. Two of his many areas of expertise are using artificial intelligence in resource scheduling and developing and evaluating IT business cases.

Ray retired from his full-time professional life a couple of years ago, but still supports selected projects as a senior adviser and coach. He also works as an actor in films and TV commercials, and as a voiceover artist and narrator.

Originally from England, Ray now lives in Australia with his partner.

You can contact Ray via LinkedIn.

Marcia E. Walker is a widely published authority and respected speaker on technology and business topics. She is particularly well-known for bridging the space between the lines of business and technology communities. Marcia advises industrial clients, government agencies, and non-governmental organizations on holistic approaches to successful information technology programs—including the application of automation, MES, ERP, and cloud technologies. A Fulbright scholar, she brings a global perspective to her work and enhances her solution recommendations with an in-depth understanding of the organizational dynamics that can enrich or impede global transformation endeavors.

Her most recent role in the corporate realm was as a senior director of the Customer Value Office at SAP. In addition, Marcia successfully managed large client engagements with the manufacturing and life sciences practices of Deloitte Consulting, after which she progressed through roles of increasing responsibility at Rockwell Automation and the industry business of Schneider Electric.

You can contact Marcia via LinkedIn.

Index