

Modeling ETL for Web Usage Analysis and Further Improvements of the Web Usage Analysis Process

A Thesis
Presented to
The Academic Faculty

by

Thilo Maier

In Partial Fulfillment
of the Requirements for the Degree
“Doctor Rerum Politicarum”



Faculty of Business Administration and Economics
Catholic University Eichstätt-Ingolstadt
March 2006

Copyright © 2006 by Thilo Maier

Modeling ETL for Web Usage Analysis and Further Improvements of the Web Usage Analysis Process

Approved by:

Professor Dr. Klaus D. Wilde
Erstgutachter (Advisor)

Professor Dr. Gholamreza Nakhaeizadeh
Zweitgutachter (Second Advisor)

Date Approved: _____

*Dedicated to my parents,
out of gratitude
for all their love and support*

ABSTRACT

Currently, many organizations are trying to capitalize on the Web channel by integrating the Internet in their corporate strategies to respond to their customers' wishes and demands more precisely. New technological options boost customer relationships and improve their chances in winning over the customer. The Web channel provides for *truly duplex communications* between organizations and their customers and at the same time, provides the technical means to capture these communications entirely *and* in great detail. *Web usage analysis* (WUA) holds the key to evaluating the volumes of behavioral customer data collected in the Web channel and offers immense opportunities to create direct added value for customers. By employing it to tailor products and services, organizations gain an essential potential competitive edge in light of the tough competitive situation in the Web.

However, WUA cannot be deployed offhand, that is, a collection of commercial and non-commercial tools, programming libraries, and proprietary extensions is required to analyze the collected data and to deploy analytical findings. This dissertation proposes the WUSAN framework for WUA, which not only addresses the drawbacks and weaknesses of state-of-the-art WUA tools, but also adopts the standards and best practices proven useful for this domain, including a *data warehousing* approach.

One process is often underestimated in this context: Getting the volumes of data into the data warehouse. Not only must the collected data be cleansed, they must also be transformed to turn them into an applicable, purposeful data basis for Web usage analysis. This process is referred to as the *extract, transform, load* (ETL) process for WUA. Hence, this dissertation centers on modeling the ETL process with a powerful, yet realizable model – the *logical object-oriented relational data storage model*, referred to as the LOORDSM. This is introduced as a clearly structured mathematical data and transformation model conforming to the *Common Warehouse Meta-Model*. It provides consistent, uniform ETL modeling, which eventually supports the automation of the analytical activities. The LOORDSM significantly simplifies the overall WUA process by easing modeling ETL, a sub-process of the WUA process and an indispensable instrument for deploying *electronic customer relationship management* (ECRM) activities in the Web channel. Moreover, the LOORDSM fosters the creation of an *automated closed loop* in concrete applications such as a recommendation engine. Within the scope of this dissertation, the LOORDSM has been implemented and made operational for practical and research projects. Therefore, WUSAN, the container for the LOORDSM, and the LOORDSM itself can be regarded as *enablers for future research activities* in WUA, electronic commerce, and ECRM, as they significantly lower the preprocessing hurdle – a necessary step prior to any analysis activities – as yet an impediment to further research interactions.

Contents

DEDICATION	v
ABSTRACT	vii
LIST OF FIGURES	xiii
LIST OF SYMBOLS OR ABBREVIATIONS	xvii
I INTRODUCTION	1
II APPLYING CRM TO ELECTRONIC COMMERCE	5
2.1 Foundations of Electronic Commerce	5
2.1.1 Electronic Commerce Growth	5
2.1.2 Defining Electronic Commerce	6
2.1.3 Classification of Electronic Commerce Applications	7
2.1.3.1 Business-To-Business Electronic Commerce	8
2.1.3.2 Business-To-Consumer Electronic Commerce	9
2.1.4 Strategic Aspects of Electronic Commerce	11
2.2 Foundations of Customer Relationship Management	13
2.2.1 Defining Customer Relationship Management	14
2.2.2 Strategic Aspects of Customer Relationship Management	15
2.2.3 Taxonomy of Customer Relationship Management	18
2.2.4 Electronic Customer Relationship Management	20
2.2.4.1 Defining Electronic Customer Relationship Management	20
2.2.4.2 ECRM Research Areas	21
2.3 Summary	23
III EFFECTIVE WEB USAGE ANALYSIS	25
3.1 Foundations of Web Usage Analysis	25
3.1.1 Introduction to Web Mining	25
3.1.1.1 Web Content Mining	26
3.1.1.2 Web Structure Mining	26
3.1.1.3 Web Usage Mining	27
3.1.2 Web Usage Analysis for ECRM	32

3.1.3	Web Personalization	33
3.2	Data Collection for Web Usage Analysis	34
3.2.1	Web Server Logs	35
3.2.2	Web Application Server Logs	37
3.2.3	Implications of the Preprocessing Phase and Open Issues	39
3.3	A System for Effective Web Usage Analysis	41
3.3.1	Prerequisites for a Web Usage Analysis Architecture	42
3.3.2	Standards Relevant to Web Usage Analysis	46
3.3.2.1	The Common Warehouse Meta-Model	46
3.3.2.2	The Predictive Model Markup Language	50
3.3.3	The Web Usage Analysis System (WUSAN) Architecture	51
3.3.3.1	The Data Access Component	52
3.3.3.2	The Population (ETL) Component	52
3.3.3.3	The Data Warehousing Component	54
3.3.3.4	The Analysis Component	54
3.3.4	Related Research Activities	56
3.4	Summary	57
IV	MODELING ETL FOR WEB USAGE ANALYSIS	59
4.1	Subsumption of the LOORDSM	59
4.2	Modeling Complex Transformations for Preprocessing	61
4.2.1	Modeling Meta-Data	61
4.2.1.1	Outline	61
4.2.1.2	Mathematical Model	62
4.2.1.3	Example	66
4.2.1.4	Summary	68
4.2.2	Modeling Streams	68
4.2.2.1	Stream Properties	69
4.2.2.2	Stream Classes	70
4.2.2.3	Summary	72
4.2.3	Transformation Modeling	72
4.2.3.1	Outline	72

4.2.3.2	Mathematical Model	73
4.2.3.3	Example	79
4.2.3.4	Summary	82
4.3	The LOORDSM	82
4.3.1	The LOORDSM from a Theoretical Perspective	83
4.3.1.1	Outline	83
4.3.1.2	Mathematical Model	83
4.3.1.3	Summary	86
4.3.2	The LOORDSM from an Implementational Perspective	86
4.3.2.1	Dimensions	86
4.3.2.2	Fact Tables	89
4.3.2.3	Star Schemas	89
4.4	Summary	92
V	DEPLOYING THE LOORDSM AND OPEN ISSUES IN WUSAN	95
5.1	Showcase Description	95
5.1.1	KDD Cup 2000 Data	95
5.1.2	Recommender Systems	96
5.1.3	The Challenge of Real-Time Personalization	97
5.1.4	Requirements for the Data Pool and Methodical Requirements	99
5.1.4.1	Requirements for Anonymous Users	99
5.1.4.2	Requirements for Known Users	100
5.2	Creating a Closed Loop	101
5.2.1	Required Mappings	101
5.2.2	Essential Tasks for Creating and Populating a Data Mart	103
5.2.3	Creating the Data Warehouse with WUSAN	104
5.2.3.1	Order Mart	104
5.2.3.2	Order Line Mart	106
5.2.3.3	Clickstream and Session Marts	106
5.2.4	Automating the Recommendation Engine	108
5.3	Performance Issues	109
5.3.1	Implementational Aspects	109

5.3.2	Aggregate Tables	110
5.4	Summary	111
VI	SUMMARY	113
Appendix A	— ADDITIONAL ISSUES OF ELECTRONIC COMMERCE	117
Appendix B	— MORE ON CRM STRATEGY IMPLEMENTATION	123
Appendix C	— MINING BASES	129
Appendix D	— MORE ON STREAMS	131
Appendix E	— MORE ON MAPPINGS	141
Appendix F	— WUSANML	147
Appendix G	— MORE ON DEPLOYING THE LOORDSM	149
	REFERENCES	179
	INDEX	197

List of Figures

Figure 1	EC matrix [adapted from EC-MATRIX].	8
Figure 2	Positive feedback cycle of CRM [Srivastava et al., 2002].	15
Figure 3	Functional chain of CRM [Bruhn, 2003, adapted from].	16
Figure 4	Multi-channel environment [derived from Payne, 2003b,a].	19
Figure 5	ECRM research framework [compare Romano Jr. and Fjermestad, 2003]. . .	22
Figure 6	High-level framework for ECRM systems [adapted from Pan and Lee, 2003].	23
Figure 7	Data-centric Web mining taxonomy [adapted from Srivastava et al., 2004]. .	26
Figure 8	The Web usage mining process [adapted from Srivastava et al., 2004; Cooley, 2000].	27
Figure 9	Server-side usage data collection for WUA.	35
Figure 10	Web application server architecture [adapted from Mariucci, 2000].	37
Figure 11	A general architecture for Web usage mining [adapted from Cooley et al., 1997].	43
Figure 12	Information supply chain [adapted from Poole et al., 2002, chapter 2].	47
Figure 13	Layers of the CWM [OMG-CWM, chapter 5].	48
Figure 14	Relationships among the standards [Thess and Bolotnicov, 2004, section 5.3].	51
Figure 15	Overall architecture of the WUA system (WUSAN).	53
Figure 16	WUSAN's four-layer architecture.	60
Figure 17	Meta-data modeling with the CWM data mining package (UML class diagram).	65
Figure 18	The XELOPES <code>MiningTransformer</code> interface (UML class diagram). . .	74
Figure 19	The XELOPES <code>MiningStreamTransformer</code> interface (UML class di- agram).	74
Figure 20	Horizontal decomposition of a vector transformation.	75
Figure 21	One-to-one mapping.	76
Figure 22	One-to-multiple mapping.	76
Figure 23	Multiple-to-one mapping.	77
Figure 24	Multiple-to-multiple mapping.	77
Figure 25	Composing complex transformations.	78
Figure 26	Data flow for populating a data mart.	83
Figure 27	The LOORDSM illustrated as a UML class diagram.	87

Figure 28	Role meta-data comprise the meta-data of a fact table in a star schema. . . .	88
Figure 29	Nesting of star schemas to model a more complex snow flake schema. . . .	90
Figure 30	Alternative modeling for the gradual ETL approach.	91
Figure 31	Template for a fictitious EC Web site.	97
Figure 32	Mappings related to the data flow for the recommendation engine.	102
Figure 33	Foreign key references of the order mart.	105
Figure 34	Modeling through a GUI.	105
Figure 35	Foreign key references of the order line mart.	107
Figure 36	Foreign key references of the session mart.	107
Figure 37	Foreign key references of the clickstream mart.	108
Figure 38	Tasks for the recommendation engine that can be automated.	108
Figure 39	Strategic functional chain of CRM proposed by Sue and Morin [2001]. . . .	124
Figure 40	Strategic process alignment framework for CRM, adapted from Payne [2003b].	127
Figure 41	The training phase of data mining algorithms.	129
Figure 42	The application phase of data mining algorithms.	129
Figure 43	The <code>MiningInputStream</code> class models the prototype of a stream.	131
Figure 44	The <code>UpdatableStream</code> interface provides writing stream access.	132
Figure 45	WUSAN's <code>MiningCollectionStream</code> class (based on Java collections). 132	
Figure 46	XELOPES' <code>MiningArrayStream</code> class (based on arrays).	133
Figure 47	The flat file stream classes (UML class diagram).	134
Figure 48	WUSAN's database streams (UML class diagram).	135
Figure 49	The filter stream classes (UML class diagram).	136
Figure 50	The <code>MiningUpdatableSqlSource</code> class manages access to an RDBMS. 137	
Figure 51	The <code>VectorFilter</code> class (UML class diagram).	138
Figure 52	Internal structure of an instance of the <code>VectorFilter</code> class.	139
Figure 53	Crucial methods and variables of the <code>OneToOneMapping</code> class.	141
Figure 54	Crucial methods and variables of the <code>OneToManyMapping</code> class. . . .	142
Figure 55	Crucial methods and variables of the <code>MultipleToMultipleMapping</code> class.	143
Figure 56	Crucial methods and variables of the <code>MiningTransformationStep</code> class. 144	
Figure 57	Methods and variables of the <code>MiningTransformationActivity</code> class. 145	

Figure 58	Foreign key references of the customer mart.	161
Figure 59	Multi-dimensional stream based on a drill-through.	171
Figure 60	Assembling an MDX statement with JPivot.	172
Figure 61	Order line mart visualized with JPivot (German localization).	177

LIST OF SYMBOLS OR ABBREVIATIONS

<i>A</i>	(Mining) Attribute.
<i>a</i>	(Mining) Vector.
<i>A</i>	Data Matrix.
ACM	Association for Computing Machinery.
API	Application Programming Interface.
ARFF	Attribute-Relation File Format.
ASP	Active Server Pages.
B2B	Business-To-Business.
B2C	Business-To-Consumer.
C2B	Consumer-To-Business.
C2C	Consumer-To-Consumer.
CGI	Common Gateway Interface.
CLV	Customer Lifetime Value.
CPO	Chief Privacy Officer.
CRISP-DM	Cross Industry Standard Process for Data Mining.
CRM	Customer Relationship Management.
CSV	Comma Separated Value.
CWM	Common Warehouse Meta-Model.
<i>D</i>	Dimension.
DBMS	Database Management System.
DMG	Data Mining Group.
DNS	Domain Name System.
DTD	Document Type Definition.
EC	Electronic Commerce.
ECRM	Electronic Customer Relationship Management.
EDI	Electronic Data Interchange.
E-P3P	Platform for Enterprise Privacy Practices.

ETL	Extract, Transform, Load.
<i>F</i>	Fact Table.
GIF	Graphics Interchange Format.
GUI	Graphical User Interface.
HTML	Hypertext Markup Language.
HTTP	Hypertext Transfer Protocol.
ICQ	“I seek you” Instant Messaging Program.
ICTs	Information and Communication Technologies.
IP	Internet Protocol.
ISC	Information Supply Chain.
J2EE	Java 2 Platform Enterprise Edition.
JDBC	Java Database Connectivity.
JDK	Java Development Kit.
JDM	Java Data Mining.
JOLAP	Java OLAP.
JPEG	Joint Photographic Experts Group.
JSP	Java Server Pages.
KDD	Knowledge Discovery in Databases.
LAMP	Linux, Apache, MySQL, and PHP.
LOGML	Log Markup Language.
LOORDSM	Logical Object-Oriented Relational Data Storage Model.
$M(A)$	Domain of A .
M_A	Meta-Data Mapping of A .
M_{A_1, \dots, A_m}	Meta-Data Mapping of A_1, \dots, A_m .
$M_{\mathbf{A}}$	Meta-Data Mapping of \mathbf{A} .
$M_{\mathbf{A}}(\mathbf{A})$	Stream.
\overline{M}_A	Inverse Meta-Data Mapping.
$\mathbb{M}(A_1, \dots, A_m)$	Set of Data Matrices with M_{A_1, \dots, A_m} .
MD5	Message-Digest Algorithm 5.

MDX	Multi-Dimensional Expressions.
MOLAP	Multi-Dimensional OLAP.
\mathbb{N}	Natural Numbers.
\mathbb{N}_0	$\mathbb{N} \cup \{0\}$.
NCSA	National Center for Supercomputing Applications.
ODBC	Open Database Connectivity.
OECD	Organization for Economic Co-Operation and Development.
OLAP	Online Analytical Processing.
OLE	Object Linking and Embedding.
OPUS	Online Publications of the University of Stuttgart.
OSI	Open Systems Interconnection.
p	Primary Key Mapping.
P3P	Platform for Privacy Preferences.
PDF	Portable Document Format.
PHP	Hypertext Preprocessor (PHP is a “recursive acronym”).
PMML	Predictive Model Markup Language.
\mathbb{R}	Real Numbers.
RDBMS	Relational DBMS.
ROLAP	Relational OLAP.
\mathcal{S}	Star Schema.
SMEs	Small and Medium-Sized Enterprises.
SQL	Structured Query Language.
T_{A_1, \dots, A_m}	Transformation with Preimage Attributes A_1, \dots, A_m .
$\Theta(A)$	Set of Missing Values.
t_{A_1, \dots, A_m}	Real-Valued Transformation of T_{A_1, \dots, A_m} .
T_{MA_1, \dots, A_m}	Meta-Data Transformation of T_{A_1, \dots, A_m} .
TCP/IP	Transmission Control Protocol/Internet Protocol.
UML	Unified Modeling Language.
URL	Uniform Resource Locator, previously Universal Resource Locator.

VAN Value-Added Network.

VoIP Voice over IP.

VPN Virtual Private Network.

W3C WWW Consortium.

WAMP Windows, Apache, MySQL, and PHP.

WEKA Waikato Environment for Knowledge Analysis.

WUA Web Usage Analysis.

WusanML WUSAN Markup Language.

WUSAN Web Usage Analysis System.

WWW World Wide Web.

XELOPES Extended Library for Prudsys Embedded Solutions.

XHTML Extensible Hypertext Markup Language.

XML Extensible Markup Language.

Chapter I

INTRODUCTION

Nowadays, in the retail, banking, telecommunications, and utilities sectors, customers are being canvassed as never before, since it has become relatively easy for them to swap organizations, which are faced with the challenge of creating customer loyalty in highly competitive marketplaces. From the point of view of organizations, the tough competitive situation became aggravated when many businesses moved to the Internet so as to leverage the Web channel for their business activities in line with a revised corporate strategy. Especially the Web channel can be regarded as a paradigm for studying this *disruptive change* of the competitive scenery. Although these changes emerged over years, they caused an abrupt shift in market shares in favor of those organizations best able to deal with the new technologies: old economy organizations have encountered greater difficulties than new economy organizations as these had a chance to adapt their organizational structures a priori to the new environment [Dueck, 2002, chapter 4]. Furthermore, the Web channel resulted in an increased transparency for customers, causing a rapid shift from *seller markets* to *buyer markets*, thereby making the situation for organizations even worse.

Given this background, activities that aim at investigating how organizations could best cope with the competitive situation in the Web channel have been launched in many research fields, resulting in innumerable contributions to areas such as economics, computer science, or the social sciences. This list of involved research areas is definitely incomplete; yet, it makes clear that the challenge of winning over the customer in the Web channel is truly multidisciplinary.

This thesis is part of the broad field spanned by the mentioned research areas. It paves the way from what is exclusively a business view of winning over the customer in the Web channel towards laying the foundations for actually coping with the emerged challenge, primarily from an *information and communication technologies* (ICTs) perspective, which is *the* essential perspective for implementing any activities.

Chapter 2 “*Applying CRM to Electronic Commerce*” singles out the retail sector and investigates the competitive situation for organizations conducting *electronic commerce* (EC). After defining the term EC, the chapter discusses the factors that make a difference in the competitive position of an EC market. General influences and technological trends, which are still turning the retail market upside down after it had been relatively constant and calculable over a long period, are considered. Then, strategic implications are discussed and *customer relationship management* (CRM) is introduced as a well-established corporate strategy that centers on the customers of an organization and aligns its business processes accordingly in a customer-centric manner. By deploying CRM, organizations expect a clear competitive edge that will preserve or even ameliorate their market position in the new competitive environment [Peppard, 2000].

Dueck [2001, chapter V] states that organizations taking CRM seriously must create a customer-centric data warehouse, which includes customer behavior, customer choices, and customer profiles, along with detailed data about the products and services offered. A data warehouse establishes the basis for deploying CRM or rather *electronic CRM* (ECRM) [Pan and Lee, 2003]. As the chapter reveals, ECRM focuses on EC channels, which, for the first time, implement *truly duplex communications* between an organization and its customers and,

at the same time, provide the technical means to capture these communications entirely *and* in great detail. As the Web channel is currently the only true EC channel, chapter 3 “*Effective Web Usage Analysis*” centers on this channel and examines its special characteristics in view of the technical prerequisites, data collection, and a system for *Web usage analysis* (WUA) that supports the realization of a CRM strategy for this channel type. The Web channel provides the collection of behavioral data about customers and prospects in unparalleled volumes and in unprecedented detail.

The question soon arises as to how such volumes of detailed behavioral customer data generated on a daily basis can be evaluated and leveraged for ECRM. WUA holds the key to this problem. After defining and introducing this notion, chapter 3 goes into the technical details and summarizes what the process of analyzing Web usage data looks like, what methods can be applied for WUA, and what prerequisites must be fulfilled from a technical point of view in order to collect behavioral data that are up to the standards required for a precise and flexible data pool for WUA. Although it is accepted as a standard approach for embedding the Web channel in an ECRM strategy, WUA is far from being standardized, meaning that it cannot be deployed offhand. This fact seems to be the main reason why the deployment of WUA has not caught up with the promises that were made during the initial hype. On the other hand, Web technology offers immense opportunities to create direct added value for customers by employing WUA to improve and tailor products and services, a potential competitive edge organizations must take advantage of. This gap has been addressed in various research contributions but still turns out to be a challenging field for current research activities.

Chapter 3 concludes with the first main contribution of this thesis: it sketches the *WUA system* (WUSAN), a collection of tools, programming libraries, and proprietary extensions (implemented in Java) that form a prototypical integrated framework for WUA. Not only does this framework address the drawbacks and weaknesses of state-of-the-art WUA tools and approaches, it also adopts the standards and best practices proven useful for this domain, including the mentioned data warehousing approach. In this context, one problem is often grossly underestimated: getting the volumes of data into the data warehouse, that is, not only must the data be cleansed, they must also be transformed, so as to turn them into an applicable, purposeful data basis that is beneficial for analyses in view of winning over the customer in the Web channel. However, the related steps of the preprocessing phase of the WUA process cannot compensate deficiencies in collecting the data in the Web channel. Consequently, chapter 3 goes into the details of *data collection for WUA* and presents established data collection approaches that help to reduce the preprocessing efforts.

Based on a comprehensive review of current publications, several open issues for a WUA system are subsequently identified, the most important of which concern the modeling and deployment of transformations during the preprocessing phase of the WUA process. At this point, the main challenge to be addressed by this thesis is identified: modeling the *extract, transform, load* (ETL) process for WUA with a powerful, yet realizable model – the logical object-oriented relational data storage model, referred to as the LOORDSM.

Chapter 4 “*Modeling ETL for Web Usage Analysis*” introduces the LOORDSM. Based on the preparatory work of the previous chapter, this chapter comprises four main contributions emphasizing the theoretical perspective:

- (1) It presents the LOORDSM with its clearly structured formal mathematical data and transformation model. For the first time, a mathematical meta-data and transformation model conforming to the *Common Warehouse Meta-Model* (CWM) is inferred and leveraged for consistent, uniform ETL modeling.

- (2) It provides a structured XML interface for ETL transformation modeling. This means that the theoretical model conveys its structured approach to WUSAN's WusanML interface, providing users with a standardized XML user interface.
- (3) It describes how the LOORDSM provides for compatibility with the CWM, thereby allowing the integration of arbitrary CWM compatible generic data pools into the ETL process and the WUA framework WUSAN.
- (4) It discusses how the LOORDSM supports automating the preprocessing phase of the WUA process, the phase being particularly challenging for the WUA domain.

The LOORDSM is introduced both in a mathematical perspective that concisely delineates its ideas, instruments, and mechanisms and a nuts and bolts perspective, that is, *Unified Modeling Language* (UML) class diagrams visualize how the LOORDSM has been concretely deployed in Java. In brief, the UML class diagram of the LOORDSM in figure 27 on page 87 can be regarded as the most compact summary possible of this thesis.

Chapter 5 “*Deploying the LOORDSM and Open Issues in WUSAN*” brings forward the proof that the LOORDSM and its embedding WUSAN framework are actually practicable and deployable. A fictitious, yet realistic, showcase is employed as a thread to discuss the concrete benefits of the LOORDSM in practical projects. This chapter summarizes the contributions of this thesis from a practical perspective (besides the benefits from a theoretical perspective mentioned before):

- (1) The LOORDSM significantly simplifies the overall WUA process by easing modeling ETL, a sub-process of the WUA process, which is an indispensable instrument for deploying ECRM activities in the Web channel.
- (2) The LOORDSM with its clearly structured formal mathematical data and transformation model greatly fosters the creation of an *automated closed loop* in concrete applications such as a recommendation engine.
- (3) XML interfaces make it possible to simplify the set up of the ETL process by supporting conceptual tasks graphically, that is, all required XML models can be assembled semi-automatically.
- (4) WUSAN and especially the LOORDSM can be regarded as *enablers for future research activities* in WUA, EC, and ECRM, as they significantly lower the preprocessing hurdle – an indispensable step prior to any analysis activities – as yet an impediment for further research interactions.

Finally, the summary on page 113 concludes this thesis and points out several implications its main contributions may have on the involved research areas. To come to the point, it can be stated that this dissertation does not provide a secret recipe for tackling the challenge of winning over the customer in EC. However, it makes a significant contribution in improving an important part of the WUA process, thereby streamlining the entire WUA process. Simplifying and structuring ETL for WUA has a significant influence on the performance of the overall WUA process both in practice and in academic research, since a cumbersome practical hurdle, impeding advanced practical and research activities on real-world behavioral data from EC Web sites, has been smoothed out: modeling ETL for WUA.

Remark (PDF version of this thesis). This document has been published as a PDF document on the official OPUS server of the Catholic University Eichstätt-Ingolstadt. Since the final URL was not known at the time this document went to press, you can find the PDF version of this document by searching for its title or author at the following URL:

<http://www.opus-bayern.de/ku-eichstaett/>.

The PDF version contains three types of hyperlinks (that do not affect the print layout):

- (1) *red boxes* that indicate intra-document hyperlinks to chapters, sections, figures, pages, and the list of symbols or abbreviations,
- (2) *green boxes* that indicate intra-document hyperlinks to the bibliography, and
- (3) *blue boxes* that indicate external hyperlinks.

Some of the external hyperlinks point to WUSAN's Java API documentation and require a login and a password to be accessed. For comments, suggestions and password requests, please send an owl to thilo@reiam.de.¹

¹No screech owls please.

Chapter II

APPLYING CRM TO ELECTRONIC COMMERCE

“Treating different customers differently is an old concept, dating back to the very beginnings of trade and commerce. We began to lose sight of this concept [...]. The astonishing success of mass production as a means for adequately feeding, clothing, and equipping unprecedented numbers of people pushed the concept even further into the background. Now, in a moment of global peace and prosperity, we finally have the chance to catch our breath, look around at the world we have created, and ask ourselves: “How can we make this better?” There are many answers. One of those answers is returning the focus of business to the individual relationships between buyers and sellers. If we have learned anything from the last two hundred years, it is that the individual *does* matter.” [Peppers and Rogers, 1999, introduction]

This introductory chapter discusses the foundations of EC (compare section 2.1) and the concept of CRM (compare section 2.2 on page 13). Further, it outlines how both ideas complement each other in the age of electronic trade with ICTs evolving rapidly, drawing organizations and customers closer to the re-establishment of real mutual individual relationships, as suggested above.

2.1 Foundations of Electronic Commerce

The buzzword EC became widespread when the Internet bubble hit its peak in 2000. Since then, the Internet hype has been attenuated by the decline and fall of the stock market, due to the fact that few of the visionary predictions concerning EC have actually materialized. Hence, practitioners and researchers are now critically reflecting on the prospective influence of EC on different parts of the old and new economies, investigating the technical prerequisites that still have to be developed and improved, considering the role that EC plays in organizations' complex processes and strategies, and sifting out applications and business models that are neither sustainable nor profitable. In this section, EC is briefly defined, and constraints, challenges, and chances that EC activities involve are discussed.

2.1.1 Electronic Commerce Growth

Based on the periodical survey USCB-SURVEY, the US Census Bureau of the Department of Commerce announced the estimate of US retail EC sales for the third quarter of 2004 – adjusted for seasonal variation and holiday and trading-day differences (but not for price changes) – was \$17.6 billion, an increase of 4.7% from the second quarter of 2004. Total retail sales for the third quarter of 2004 were estimated at \$916.5 billion, an increase of only 1.4% from the second quarter of 2004.

The third quarter 2004 EC estimate increased 21.5% from the third quarter of 2003, while total retail sales increased only 6.2% in the same period. EC retail sales in the third quarter accounted for 1.9% of total retail sales. This amazing development of EC retail has been more or less constant for many years and is likely to remain stable, or even increase, in the near future.

Greenspan [2004] projects the share of EC in total retail to reach 5% by 2008. This figure is still small, partially due to some very large retail categories not achieving high penetration, for instance, the grocery sector. These trends are consistent with the report EC-REPORT, which estimates the overall share of EC in retail to be 1.5% of the total sales volume in the EU, with similar growth rates as in the US (but some slight local variations among European countries).

Growth in online retail is in part due to new buyers, not just veterans, and, in 2004, 30% of the US population bought products and services online – a number that is projected to reach 50% by 2008 [Rush, 2004]. This increase is accompanied by the trend that the average online retail spending per buyer in the US in 2004 amounted to \$585, up from \$540 in 2003 and is projected to be close to \$780 per buyer in 2008. A similar trend is anticipated for the EU, where the EC market will grow from €29 billion in 2003 to €117 billion in 2009, with 61% of European Internet users then buying online, spending €843 on average [Salcedo et al., 2004].

Given the phenomenal growth of retail EC volume, which according to EC-REPORT can also be observed in other economic sectors to a greater or lesser extent, organizations should take this astounding trend as an incentive to become involved with EC or to boost their existing EC activities, since EC is gaining on traditional retailing very quickly.

2.1.2 Defining Electronic Commerce

Definitions of EC abound, which is a sign of lack of consensus about what EC actually is. Wilkins et al. [2000] analyze a variety of definitions of EC and conclude that those definitions allow for partial aspects of EC only. In response to this finding, the authors propose the application of the *ideal type concept*.¹ In this approach, the distinctive features of the overall concept to be defined are emphasized in definitions covering partial aspects of the concept only. Following up on this approach, the ideal type definitions of the term EC set out below are taken as a jumping-off point to synthesize definition 2.1 on the next page, which is a comprehensive definition accounting for all important aspects of EC.

- (1) “*EC is the automation of commercial transactions using computer and communications technologies.*”, see Westland and Clark [1999, p. 1]. This ideal type refers to ICTs and the automation of transactions as distinctive features of EC.
- (2) “[*EC*] *is concerned specifically with business occurring over networks which use non-proprietary protocols that are established through an open standard setting process, such as the Internet.*”, compare Hawkins et al. [1999, p. 28]. This ideal type mentions networks, their technical foundations, and relevant network protocols as distinctive features of EC.
- (3) “*EC involves the undertaking of normal commercial, government, or personal activities by means of computers and telecommunications networks; and includes a wide variety of activities involving the exchange of information, data or value-based exchanges between two or more parties.*”, compare Chan and Swatman [1999]. This ideal type emphasizes ICTs networks, associated business activities, and transactions for exchanging material and immaterial goods.
- (4) “[*EC*] *falls into two broad categories. First is the use of technology to re-engineer business processes that are primarily internal to the organization. [...] Second, it relates to the use of technology in how an organization interfaces with business partners, whether they are customers or suppliers – an external focus.*”, compare Peppard [2000]. This ideal type reduces EC to the application of ICTs for internal and external business processes.

¹A short introduction to ideal types can be found in WIKIPEDIA.

- (5) “EC is the sharing of business information, maintaining business relationships, and conducting transactions by means of telecommunications networks.”, compare Zwass [1996]. He further states that EC not only includes buying and selling goods but also involves various processes within individual organizations that support that goal. This ideal type considers ICTs as a means of managing business information, business relationships, and business transactions and extends the notion of EC to intra-organizational processes.

Based on the above collection of ideal type definitions, the most significant features of EC are synthesized into the following fundamental definition.

Definition 2.1 (Electronic Commerce). *EC* refers to the sharing of business information, maintaining business relationships, and conducting automated, value-based commercial transactions over ICTs networks that make use of Internet protocols, open standards, and standard software. The interface between organizations and their customers, the interface between organizations, and the organizations’ internal processes, which are a prerequisite to conduct the mentioned business activities, make use of the benefits of Internet-based ICTs networks and contribute to EC activities.²

Remark. Definition 2.1 picks up the focus of EC on networks that use non-proprietary Internet protocols³, open standards⁴, and standard software⁵, which are a relatively new phenomenon. Traditional EC through electronic data interchange (EDI), fax communication, inter-enterprise messaging, and file transfer have been the dominating EC technologies for several years.

Traditional EC relies, for the most part, on highly proprietary value-added networks (VANs) and private messaging networks, which are characterized by relatively high costs and limited connectivity, on the one hand, and by strong security, reliability, and confirmation of receipt, on the other [Pyle, 1996]. These characteristics are in contrast to the Internet, which is interactive, inexpensive to use, and open to multimedia applications but does not guarantee delivery and security precautions, unless they are explicitly provided. Since many organizations that have been using traditional EC technologies for years are now replacing their legacy ICTs by new Internet-based technologies, for example, virtual private networks (VPNs), it makes sense to restrict ICTs networks in definition 2.1 to Internet-based networks in anticipation of the imminent convergence towards Internet technology.

2.1.3 Classification of Electronic Commerce Applications

Harmsen [2001] mentions that EC applications can be classified into four major categories as depicted in figure 1 on the following page. The following two sections briefly introduce the two dominant EC classes, namely *business-to-business* (B2B) EC in section 2.1.3.1 on the next page and *business-to-consumer* (B2C) EC in section 2.1.3.2 on page 9.⁶

²Some authors introduce the term *e-business* instead of EC. Normally, e-business focuses exclusively on the Internet as a transaction channel, whereas EC originated from legacy electronic channels such as electronic data interchange (EDI). Since most legacy electronic channels are being replaced by channels based on Internet technology (see above remark), both notions can be used interchangeably. In the remainder of this thesis, the term EC is preferred to the term e-business.

³For instance, TCP/IP and HTTP [W3C-HTTP] with the former representing the *transportation layer* and the latter representing the *presentation layer* of the *Open Systems Interconnection* (OSI) reference model [compare Day and Zimmermann, 1995].

⁴For instance, XML [W3C-XML] as a core technology and related special instances such as XHTML [W3C-XHTML].

⁵For instance, J2EE [SUN-J2EE], PHP [PHP], and the Apache Web server [APACHE].

⁶*Consumer-to-business* (C2B) EC and *consumer-to-consumer* (C2C) EC are discussed in the appendix section A.1.1 on page 117 and section A.1.2 on page 117, respectively, for the sake of completeness.

	Business	Consumer
Business	B2B	B2C
Consumer	C2B	C2C

Figure 1: EC matrix [adapted from EC-MATRIX].

2.1.3.1 Business-To-Business Electronic Commerce

B2B EC refers to definition 2.1 on the previous page limiting business relationships to relationships among organizations. It is directed towards alleviating transaction inefficiencies in the supply chain and promises to reduce the costs of inter-business transactions (*before, during, and after* transactions) by automating procurement [Lucking-Reiley and Spulber, 2001].

B2B EC can be conducted between organizations directly; but, in practice, *intermediaries* unite buyers and sellers in virtual marketplaces in order to facilitate the consummation of transactions [Grover and Teng, 2001]. Intermediaries reduce search costs by consolidating markets and provide an assortment of goods and services that gives buyers the cost efficiency of one-stop shopping [Lucking-Reiley and Spulber, 2001].

Organizations consequently map parts of their supply chain, especially the procurement process, to the Internet by deploying new EC applications. The adoption of Internet protocols and the use of Internet infrastructure (which is primarily used to set up extranets) have transformed traditional B2B EC over EDI into a flexible system that includes a much wider range of organizations than before. The list below sets out key catalysts that favor the rapid adoption of B2B EC and factors that are responsible for its sustainable growth.⁷

- (1) **Existing streamlined and automated internal business processes.** Many organizations have been using EDI for procurement and are thus able to upgrade to current ICTs for EC more easily, as they do not have to redefine existing business processes [Riggins, 1999]. For organizations of all sectors of the economy, it has also become important to deploy ICTs for EC in order to streamline EC-related internal business processes.

Therefore, numerous organizations are currently in a state of transition since they are about to replace their legacy EDI systems by extranets. These extranets are based on Internet technology and provide the technological link between external business partners and the organizations' internal business processes [Turban and King, 2003, appendix 5A]. The fact that internal structures are already aligned to an overall business strategy facilitates the transition to the new mode of EC, which is based on Internet technology [Coltman et al., 2002]. Since automating internal business processes appears on the agenda of most organizations anyway, the step towards B2B EC activities is relatively easy and does not add significant overhead [EC-REPORT, p. 18].

- (2) **Implementation of leading-edge ICTs.** Large organizations generally dispose of leading-edge ICTs in terms of financial and human resources. Consequently, the initial hurdle to implementing EC-enabling ICTs is relatively low [Coltman et al., 2002]. Although small and medium-sized enterprises (SMEs) lagged behind in the EU, they are now catching up due to an incipient migration towards more sophisticated ICTs [EC-REPORT, p. 18].

⁷B2B EC has the potential for real, transformative change in the near future. According to McCall [2001], worldwide B2B EC volume is projected to reach \$8.5 *trillion* in 2005, still growing with rates beyond 40%. This is a tremendous contrast to the B2C EC magnitudes mentioned in section 2.1.1 on page 5 and accounts for the fact that B2B EC activities are currently on the fast track compared to B2C EC activities.

- (3) **Reduction of costs by online purchasing and supply chain integration.** Organizations are cost conscious, inasmuch as every dollar saved in procurement is equivalent to a dollar of profit [Coltman et al., 2002]. Using EC to decrease costs for supplies and to make related processes more efficient is a major driver for B2B EC. Since many organizations have to manage large numbers of transactions with their suppliers on a daily basis, even fractional improvements of these processes can aggregate to quite substantial savings [EC-REPORT, p. 22]. B2B EC will lead to a reduction in transaction costs [Lucking-Reiley and Spulber, 2001], an improvement of product quality, both an improvement of and a cost reduction in customer service, improvement of productivity, and a reduction in production costs [Hawkins et al., 1999, p. 37]. Once organizations have made substantial investments in ICTs to support their supply chain, they give incentives to encourage partners to join B2B EC applications in order to take advantage of further mutual efficiency gains [Coltman et al., 2002].
- (4) **Openness as underlying technical and philosophical tenet.** The widespread adoption of Internet technology as the foundation for business platforms is essentially due to its non-proprietary standards, its open nature, and the huge industry that has evolved to support it. The economic power that stems from joining a large network will help to ensure that new standards remain open. More importantly, the emerging strategy is openness, with many of the most successful EC ventures granting business partners and consumers unparalleled access to their internal systems and processes [Hawkins et al., 1999, p. 11]. EC standards help to organize and exchange information in a way that is consistent across entire organizations, including their ICTs-based systems [EC-REPORT, p. 32].
- (5) **Globalization of the marketplace.** EC drives changes that are already underway, for example, establishment of electronic links between businesses and globalization of economic activity. End customers and organizations increasingly have the ability to communicate and transact business anywhere, anytime [Riggins, 1999]. This has profound impacts on the competitive environment, not the least of which is the erosion of economic and geographic boundaries [Hawkins et al., 1999, p. 10]. Thus, two consequences can be observed: (i) *increased competition* and (ii) *increased possibilities*. While most organizations engage in B2B EC as a defensive reaction to global competitors that are trying to take the lead in B2B EC, some organizations seize the opportunity to make their products and services known to global audiences and extend their market reach [EC-REPORT, p. 85].

2.1.3.2 Business-To-Consumer Electronic Commerce

B2C EC refers to definition 2.1 on page 7, limiting business transactions to those between organizations and consumers. Although B2B EC represents the bulk of all EC activities in terms of business volume, most attention about EC has focused on B2C EC [Hawkins et al., 1999, p. 38].⁸ According to Coltman et al. [2002], enterprises operating in the B2C segment currently attract only a minority of consumers compared to the total market (compare section 2.1.1 on page 5), while nevertheless reaching a critical mass of consumers to be considered profitable. As consumers still have to become acquainted with the Internet as an everyday medium and hence are still experimenting online, they are hard to assess by organizations. On the other hand, many business-related barriers must still be overcome, for example, security, privacy⁹,

⁸The reason for this may be the fact that B2B EC takes a back seat since organizations do not make a big thing of their procurement activities, whereas B2C EC is much more discernible in everyday life.

⁹See appendix section A.2 on page 118.

network access, and low bandwidth [Coltman et al., 2002]. Moreover, evidence has emerged that many consumers do not employ exhaustive search strategies and may become locked into one attractive site, not making use of any alternatives [Riggins, 1999].¹⁰

The largest segment of B2C EC involves *virtual* products that can be delivered directly to the consumers' computers over the Internet: these are primarily entertainment (for example, download of music, video on demand, and online games), travel (for instance, paperless airline and train tickets), media (for example, online editions of established print media or TV channels), financial services (for instance, brokerage, banking, consumer credits, and insurance), personal digital assistant services (for example, e-mail, date book, and mobile information services), and software (digital delivery) [Torlina et al., 1999].

The second increasingly important segment of B2C EC are *physical* products, which require sophisticated logistics to be delivered to consumers on time and reliably. To date, the main physical products sold through B2C EC can be divided into the following three categories [Evans, 2004]. (1) *Plateau growth categories*, including those categories that are expected to grow "only" less than 10% over the next five years (personal computers¹¹, peripherals, books, toys, and video games). (2) *Steady growth categories*, including those categories that will grow between 10-30% over the next five years (apparel, office products, consumer electronics, sports goods, footwear, flowers, and large appliances). (3) *Steep growth categories*, including those categories that will grow between 30-50% over the next five years (music (non-digital delivery), groceries, garden supplies, housewares, home improvement accessories, auto parts, medical supplies, over-the-counter drugs, personal care products, and nutraceuticals).

Given these astounding growth figures and the overall trends in B2C EC mentioned in section 2.1.1 on page 5, the question arises as to what precisely drives B2C EC. Two main factors become apparent: (i) *the price* and (ii) *customer service*.¹²

- (1) **Greater transparency and greater flexibility for consumers.** In B2C EC, consumers can place orders at any time and browse products independent of office hours. The availability of prompt pricing information and concise product information on the Internet in combination with organized Web-based marketplaces increases market transparency, simplifying systematic price and product comparison and lowering search costs [Bakos, 1998]. As a consequence, buyers become aware of potential substitutes that can result in a substantial reduction of costs (in turn leading to greater pressure on sellers) [Harmsen, 2001].
- (2) **Increasing market competition.** Enhanced information access for consumers increases competition even where concentration is already high. This has an impact on pricing, which is a positive effect for consumers but involves risks of eroding profits for online retailers. And while organizations are forced to look for further cost-saving potentials, which could have a detrimental effect on product and customer service quality [EC-REPORT, p. 139], on the other hand, they improve CRM in order to stand out from competitors as a reaction to fierce competition. This leads to increased differentiation [Bakos, 1998].
- (3) **Mass customization.** Large organizations with mass customer bases are continuously adopting the concept of *mass customization*¹³ for their B2C EC activities by integrating

¹⁰The author mentions that customer lock-in can be exploited as a strategic aspect of B2C EC.

¹¹Personal computers have the largest penetration of any category: 45% of total sales.

¹²Both factors shape up as advantages for consumers rather than for organizations. Nevertheless, all driving factors concerning the alignment of internal business processes (finally leading to a competitive advantage) mentioned for B2B EC in section 2.1.3.1 on page 8 also apply to organizations' B2C EC activities.

¹³The notion *mass customization* was coined by Davis [1987, p. 169]: "Mass customization of markets means

their ICTs into CRM activities. That is, ICTs are deployed for data processing of customer information, marketing, sales, and customer service, aiming at collecting detailed information about customers [EC-REPORT, p. 139]. Collected and consolidated information can then be used to establish real one-to-one relationships between organizations and their customers, fueling the positive feedback cycle of CRM in figure 2 on page 15 and ultimately leading to profitable customer relationships (compare figure 3 on page 16).

- (4) **New services on the Internet.** B2C EC enables organizations to provide new and better value-added services to customers, that is, services that were impossible or extremely laborious to deploy prior to the massive adoption of ICTs in retailing [Harmsen, 2001; Riggins, 1999]. For instance, providing additional information to customers with Web-based inventory systems – combined with shipment tracking systems – leads to a more transparent shipping procedure and helps customers to assess how long the processing of orders will presumably take.
- (5) **Broadband meets mass market.** Given the example of South Korea, Lee et al. [2003] state that easy and inexpensive access to broadband Internet for both organizations and consumers is a driving factor for B2C EC. Due to a high broadband Internet penetration and availability, South Korean consumers are more receptive to new services and applications offered over the Internet, broaden their range of online activities, and spend significantly more time and money online than consumers in countries where baseband Internet access is dominant.

Example (The world’s largest B2C Web site, adapted from Banham [2004]). While sometimes still referred to as an online bookstore, *Amazon* [AMAZON] now offers a broad product line from apparel, sports goods, and jewelry to new services including a feature that let its 39 million active customers¹⁴ make “1-Click” contributions to the American Red Cross in the aftermath of the devastating tidal wave in South East Asia of December 2004.

In addition to trading for its own account, Amazon is now offering its technology and B2C EC expertise to third parties, relying on two strategies: (i) *Merchants@amazon.com*, that is, setting up merchandizing relationships designed to open up an incremental sales channel to retailers wishing to access Amazon’s customers through Amazon’s Web site and (ii) *Merchant.com*, that is, allowing a retailer to control the look and feel of the Web site while letting Amazon handle technology services, order fulfillment, and customer service. Amazon has never focused on short-term profits, instead implementing a long-term vision for building its technology infrastructure – a research-and-development strategy more like that of a pharmaceutical company. As a result, Amazon today has the world’s largest online consumer laboratory offering unparalleled possibilities to make experiments on improving customer service and personalizing the shopping experience.

Remark. The remainder of this thesis focuses exclusively on B2C EC. Hence, from this point on, EC refers to B2C EC, and all other EC categories are denoted explicitly.

2.1.4 Strategic Aspects of Electronic Commerce

Definition 2.2 (Strategy). A *strategy* is a broad-based formula for how a business is going to compete, what its goals should be, and what plans and policies will be needed to carry out those

that the same large number of customers can be reached as in mass markets of the industrial economy, and, simultaneously, they can be treated individually as in the customized markets of pre-industrial economies.” While this definition refers to physical products, it can be extended to virtual products and customer service.

¹⁴This number is based on the number of e-mail addresses from which orders originated in 2003.

goals. Strategy means search for actions that will significantly change the current position of an organization, shaping its future [Turban and King, 2003, chapter 11].

The drivers for B2C EC discussed on page 10 in the previous section turned out to be stimulations for customers rather than for organizations. Since customers are well aware of the advantages that EC implies for them, organizations are trying to turn this awareness into a competitive edge by seeking new ways of improving the end-to-end customer shopping experience, in turn boosting sales and winning customer loyalty by establishing innovative ways of satisfying ambitious customer needs. In detail, organizations are faced with the following ramifications:

- (i) Increasingly discerning customers are demanding new services, more detailed information, and better quality at a lower price [Chu and Morrison, 2003].
- (ii) Markets are transforming from seller markets to buyer markets due to a globalized competition. Also, in mature, saturated markets, it is difficult for retailers to sustain differentiated brands and value propositions [Piller and Schoder, 1999].
- (iii) Organizations are forced to evaluate their strategic positions in terms of the two EC drivers price and customer service [Harmsen, 2001].
- (iv) Rapid evolution and adoption of new ICTs present both opportunities and risks for organizations seeking to innovate [Chu and Morrison, 2003].

In order to respond to these challenges, all efforts to target them must be bundled in an organization's corporate strategy. EC is not a mere technology but an integral strategic constituent for all types of organizations that offer electronic channels to their customers.

Porter [1985] proposes conceptual typologies and empirical taxonomies of strategies using the three variables *cost leadership*, *differentiation*, and *focus*. These variables have been proven to decisively influence an organization's performance. Lederer et al. [1996] concisely discuss the three resulting basic strategy types, which are applicable to any sector of the economy:

- (1) **Cost Leadership Strategy.** Cost leadership means providing a standardized product or service at very low per-unit costs for many price-sensitive buyers. In EC, gaining cost leadership can be achieved by decreasing distribution costs and transaction costs through a reduction in overhead, for example, inventory and personnel. Moreover, costs can be reduced through direct customer contacts, for instance, by eliminating inefficiencies associated with paper processing, by streamlining processes and information flow (internally and externally), and by introducing electronic procurement [Fruhling and Digman, 2000; Chang et al., 2003].
- (2) **Differentiation Strategy.** Differentiation means providing products and services that are considered unique industry-wide and that address many buyers who are relatively price-insensitive. For EC, this can be accomplished by implementing mass customization (see item 3 on page 10), that is, by offering customized products and services and by tailoring sales and service processes to the individual requirements of the customers [Fruhling and Digman, 2000; Chang et al., 2003].
- (3) **Focus Strategy.** Focus means providing products that fulfill the needs of particular buyers who are fewer in number in an industry. While cost leadership and differentiation strategies attempt to address a whole industry, focus strategy addresses specific and smaller clusters

of buyers. For EC, focus can be achieved by extending the market reach for specialized products and services and by narrowing the competitive scope of business activities to specific specialized segments [Fruhling and Digman, 2000].

The question arises as to what basic strategy is the most promising to cope with the ramifications of EC evolution. According to Piller and Schoder [1999], a vast majority of organizations does not pursue a mere cost leadership strategy. This lies in the fact that cost structures of organizations start converging, once organizations reach an advanced level of their corporate life-cycle. Furthermore, EC-enabling ICTs, which contribute to the reduction of process costs, are also available to competitors [Chang et al., 2003]. Since only one cost leadership strategy can be successful, organizations following this approach will be exposed to a competitive environment where only few organizations can survive [Harmsen, 2001].

Hence, most organizations concentrate on differentiation or focus, or on a hybrid strategy combining both approaches [Lederer et al., 1996; Chu and Morrison, 2003].¹⁵ In EC, differentiation and focus are realized through *customer orientation*.¹⁶ This term reflects the organizations' understanding of their target buyers in order to continuously create added value for them [Chang et al., 2003]. To this end, they accumulate knowledge about their customers (primarily by collecting behavioral data in electronic channels), which is an intangible asset that is difficult to imitate by competitors, ultimately leading to a strategic advantage [Chang et al., 2003]. Organizations thus avoid competing solely in pricing. Since varying differentiation and focus strategies can coexist successfully within the same market, organizations following this approach should therefore be in a more secure competitive position.

The point now is how customer orientation can be deployed for EC. The answer is CRM, a concept to manage and cultivate sustainable and profitable long-term customer relationships for the benefit of both organizations and their customers. This concept, which originates from brick-and-mortar (old economy) organizations, is currently subject to change and is about to be adapted to new markets that are driven by the rapid development of ICTs. The concept of CRM is discussed in the next section.

2.2 Foundations of Customer Relationship Management

In the past years, CRM has been established as a holistic approach to enforce customer orientation in organizations. According to Scribner [2002, 2001], the main objectives of CRM activities comprise (i) the set-up of sustainable and profitable relationships between organizations and their customers, (ii) the continuous optimization of customer relationships, and (iii) the fosterage of customer relationships to maximize the profits for both organizations in terms of monetary profits and their customers in terms of benefits they draw from these relationships. The imperativeness to find alternative routes to competitive advantage has been driven by profound changes and trends in the business environment, some of which are congruent with those EC is affected by¹⁷ [compare Pan and Lee, 2003; Peppard, 2000; Payne, 2003b]:

¹⁵This does not imply that they do not pay attention to costs. In fact, the reduction of costs is an essential strategic component, but for EC it is not sufficient.

¹⁶According to definition 2.1 on page 7, EC revolves around value-based commercial transactions. That is, the primary focus of differentiation lies on transactions rather than on the products sold through transactions. As most physical products sold through EC are standardized (compare page 10), there is limited room to differentiate them. Even many virtual products are standardized and can hardly be differentiated (again, compare page 10). In consequence, transactions and all related customer-facing processes are subject to differentiation in EC.

¹⁷Mentioned in section 2.1.3.2 on page 9 and section 2.1.4 on page 11.

- (1) Globalization of markets making time and location irrelevant and consequently, growth and diversity of competition.
- (2) Transition from seller markets to buyer markets due to EC, accompanied by the escalating expectations and empowerment of customers.
- (3) Shift from transactional marketing to relationship marketing, along with the realization that customers are business assets and not a mere commercial audience, leading to the acceptance of the necessity for a trade-off between *delivering* and *extracting* customer value.
- (4) Development and availability of new ICTs and therefore greater utilization of ICTs in managing and maximizing the value of information.
- (5) Recognition of the benefits of using information proactively rather than solely reactively.
- (6) Transition in structuring organizations – on a strategic basis – from functions to processes, backed by ICTs.
- (7) Substantial progress in developing, improving, and launching scalable algorithms for mass customization.

Many organizations today are aware of the importance of CRM and its potential to achieve the competitive edge. Some are already about to change their business processes and build solutions based on ICTs that enable them to acquire new customers, retain existing customers, and maximize the customer lifetime value (CLV) of every customer [Peppard, 2000]. While the projected compound annual growth rate in CRM spending of 6.7% through 2006 (totaling \$17.7 billion by then [see Greenspan, 2003a]) shows that organizations are steadily continuing their efforts in CRM, there is still significant confusion surrounding the definition of CRM and its role. This section aims at clarifying the theoretical foundations of CRM in view of differentiation for EC.

2.2.1 Defining Customer Relationship Management

Many authors refer to the concept of CRM as a mere deployment of ICTs, reducing it to its technological constituent, which provides the prerequisite to analyze customer-related processes [Paas and Kuijlen, 2001]. Hippner [2004] states that – although ICTs play an important role in CRM – an organization’s general conditions and strategies may not be ignored. Hence, he proposes the following definition of CRM:

Definition 2.3 (Customer Relationship Management). CRM refers to a customer-oriented corporate strategy, which, by means of ICTs, attempts to establish profitable long-term customer relationships by deploying and strengthening integrated and personalized concepts for an organization’s marketing, sales, and customer service departments.

Scribner [2001] describes customer relationships as learning relationships in which organizations exchange personalized services for customer loyalty. Customers convey their personal preferences explicitly and implicitly to organizations through communications and purchasing behavior. Organizations capture customer behavior and preferences so as to create personalized products and services. In the course of time, learning relationships become unique and are perceived as a surplus value, creating a lock-in situation for customers [Amit and Zott, 2001].

Following Peppard [2000], the premises of the above definition are the facts that (i) existing customers are more profitable than new customers, (ii) it is less expensive to sell an incremental product to an existing customer than to sell it to a new customer, (iii) customer retention can be maximized by matching products and levels of service more closely to customer expectations, and (iv) attracting new customers is generally expensive. The central objective of CRM is thus to maximize the CLVs, leading to higher long-term profits. According to Berson et al. [2000], the corporate strategy mentioned in the definition above involves five activities geared for (1) *customer profitability*, (2) *customer acquisition*, (3) *cross-selling*, (4) *customer retention*, and (5) *customer segmentation*, each of which is briefly discussed in the section below.

2.2.2 Strategic Aspects of Customer Relationship Management

CRM strategy implementation involves five essential activities aiming at the following aspects:

- (1) **Customer Profitability.** *Perpetuation and deepening of existing customer relationships [compare Schumacher and Meyer, 2004, section 2.3]:* Acquiring new customers generally involves large costs, for instance, due to expensive marketing activities and complex target campaigns. As a consequence, retained customers are more profitable, especially since customer relationships become rewarding after a grace period only [Hippner, 2004].
- (2) **Customer Acquisition.** *Transform potential buyers into regular customers [see Schumacher and Meyer, 2004, section 2.3]:* On the whole, regular customers are loyal to an organization and are reluctant to switch to a competitor. Hence, they are more tolerant in view of minor problems concerning their customer relationships than new customers. Moreover, regular customers – in contrast to new customers – attach more importance to a good customer relationship than to adjustments in the product price.¹⁸ Finally, regular customers contribute to an organization's positive reputation by spreading a positive image.
- (3) **Cross-Selling.** *Discover and exploit cross-selling and up-selling potentials [see Schumacher and Meyer, 2004, section 2.3]:* Within the positive feedback cycle of CRM in figure 2, it is possible to place information about products and services that correspond to individual customer preferences. By tailoring product offers and services to the potential needs of customers, it is feasible to cover their demands for products and services efficiently by cross- and up-selling activities, leading to additional revenues. Generally speaking, it can be said that customer relationship behavior anticipates customer demands. Thus, an interactive dialog is key in revealing customer preferences [Peppard, 2000].

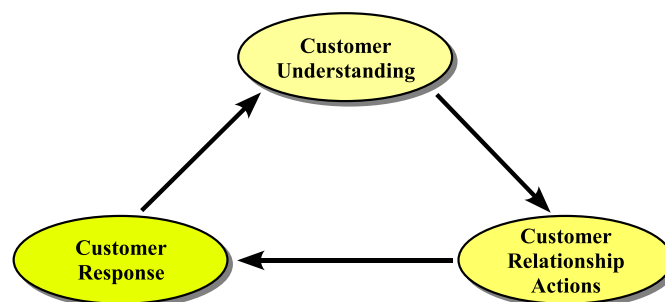


Figure 2: Positive feedback cycle of CRM [Srivastava et al., 2002].

¹⁸This translates into reduced price competition [Scribner, 2001].

- (4) **Customer Retention.** *Increasing customer satisfaction leads to augmented customer retention [compare Peppard, 2000]:* Due to the rising saturation of seller markets, repeated transactions with the same customers may contribute a large portion to an organization's total revenue [Schumacher and Meyer, 2004, section 2.3]. Customer satisfaction¹⁹ and customer retention²⁰ are considered to be key components of economic success [Hippner, 2004; Bruhn, 2003]. Figure 3 depicts the underlying *functional chain of CRM*, which displays the connection between CRM activities and profitable customer relationships for both organizations and their customers.

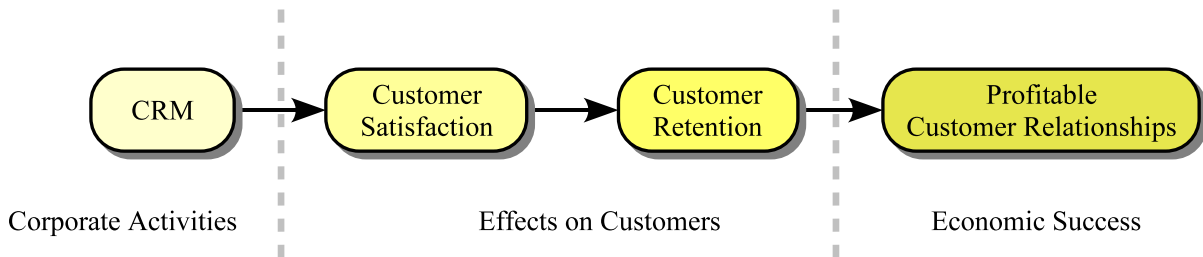


Figure 3: Functional chain of CRM [Bruhn, 2003, adapted from].

- (5) **Customer Segmentation.** *Improving communications with customers [compare Srivastava et al., 2002]:* An essential element of CRM is communicating with customers. This consists of two components: (i) decide what message to send to each customer segment and (ii) select the appropriate channel for the message. In order to tailor communications as best as possible to customer preferences, customer segmentation can be applied so as to create homogeneous customer segments that are targeted using the same communications strategy. It is crucial to measure the impact of communications by conducting response analyses, that is, (a) determine how effective a campaign has been, (b) validate the goodness of customer segmentation, and (c) calibrate and refine the models for the various communication channels by setting up feedback loops to learn from customers.

In terms of the functional chain of CRM in figure 3, which is comprised of the three sections (a) *corporate activities* within the scope of customer relationships as input, (b) *effects on customers*, and (c) *economic success* for organizations as output, implementing a CRM strategy means configuring the first element of the functional chain. Corporate actions ideally improve customer satisfaction, which has a positive influence on customer retention, which eventually leads to profitable customer relationships and economic success [Bruhn, 2003].²¹ In order to

¹⁹Customer satisfaction describes the degree to which customers are comfortable with the business relationship they are involved in. Das et al. [1999] propose a customer satisfaction model to characterize the factors that influence customer satisfaction in the domain of EC: (1) *service quality*, which is influenced by *employee satisfaction* and *employee contextual knowledge*, (2) *solution quality*, which is influenced by *employee contextual knowledge*, and (3) *price*.

²⁰Customer retention is the effort carried out by organizations to ensure that their customers do not switch over to competitors [Srivastava et al., 2002]. Customers do not want to go through the trouble of teaching their preferences to multiple organizations. They hence have an incentive to remain loyal, leading to both higher switching costs for customers and advantages in competition for organizations over time [Scribner, 2002].

²¹CRM activities have two effects on profitability. On the one hand, they create additional revenue or safeguard existing revenues. On the other hand, they contribute to *cost reductions* [Schumacher and Meyer, 2004, section 2.3]. The longer customer relationships last, the more can be known about customer preferences. Learning from customer relationships enables organizations to serve their customers' needs more precisely, preventing not only costs that may occur if customer relationship actions are not thoroughly targeted but also costs arising from

leverage the functional chain, organizations must align all customer-facing processes and all dependent internal processes in a customer-oriented way. To this end, an overall CRM strategy that defines the customer segments to be addressed through different channels is required.

All actions concerning the first element of the functional chain should be applied on a constant basis, feeding the positive feedback cycle of CRM in figure 2 on page 15. According to [Srivastava et al., 2002], within the positive feedback cycle, improved *customer understanding* results in more efficient *customer relationship actions*, producing better and more frequent *customer response*, in turn yielding more detailed customer data. This is the starting point for refined customer understanding in the next iteration of the feedback cycle.

Hippner [2004] mentions two crucial prerequisites for successful CRM strategy implementation that have to be addressed concurrently with any CRM-related efforts:

- (i) It is fundamental to deploy an integrated information system that not only consolidates customer-related information (one face *of* the customer) but also synchronizes all communication channels (one face *to* the customer).²²
- (ii) In order to practice effective CRM, it is compulsory to align all business processes to the customer base to support a customer-oriented corporate strategy.²³

Remark. Schögl and Schmidt [2002] characterize different aspects of CRM and identify the following three CRM perspectives.

- (1) **Strategic Perspective.** *Transition from product orientation to active management of customer relationships:* The strategic directions of organizations implementing CRM seek to forge and exploit customer potentials, eventually leading to specific customer services that are based on the customer's individual value for the organization.
- (2) **Process Perspective.** *Integrated and holistic view of customer relationships:* The process perspective aims at realizing CRM strategies in an integrated and holistic manner. In order to achieve this, organizations offer customized services across all customer touchpoints, with such service offerings being coherent and consistent at any point in time. This involves the synchronization of front- and back-office processes as mentioned in item ii.
- (3) **ICTs Perspective.** *Support and realization of marketing activities by means of ICTs:* The ICTs perspective stands for automating all customer management processes and all customer relationship processes that connect customers across a variety of touchpoints to the marketing, sales, and customer service departments of an organization. As mentioned in item i, an integrated information system is of central importance to this perspective.

The remainder of this thesis focuses on the ICTs perspective in order to meet the challenge of differentiation for EC with CRM. Clearly, the use of ICTs does not automatically represent added value for customers, unless ICTs are embedded into a sound CRM strategy that focuses on back-office processes as well. Rather, ICTs should be considered enablers that make it possible for organizations to create services of added value for their customers.

complaints. Furthermore, implementing a CRM strategy involves streamlining customer-facing processes, which contributes to the reduction in process costs (compare item 3 on page 9 analogously applying to B2C EC).

²²In section 3.3.3 on page 51, the WUSAN prototype is introduced. This system can be regarded as an integrated information system. Although its current implementation allows for data from the Web channel only, the system can be extended to capture data from various communication channels.

²³Fahey et al. [2001] discuss the central role of *knowledge management* to support the linking and alignment of EC-related processes and business processes for CRM.

2.2.3 Taxonomy of Customer Relationship Management

From the ICTs perspective of CRM, the following taxonomy of sub-areas of CRM has emerged [Berson et al., 2000, chapter 3]:

- (i) **Analytical CRM.** *Analytical CRM* refers to developing customer understanding through data analysis for the purpose of more effective CRM [Srivastava et al., 2002]. It comprises all the *data collection*, *data analysis*, and *decision support* activities associated with customer relationships [Schumacher and Meyer, 2004, section 2.2].

Large organizations that serve a mass customer base have difficulties in understanding the expectations and demands of the individual customers. They are thus driven to leverage analytical CRM in order to collect and analyze data covering all channels and customer touchpoints, aiming at creating a complete and holistic view of their customers. Tremendous progress in data management and data analysis provides the opportunity to develop fine-grained individual customer understanding, even for mass customer bases.

- (ii) **Operational CRM.** In *operational CRM*, it is assumed that optimal business processes in the front-office (that is, processes related to marketing, sales, and customer service) will optimize management of customer relationships [Paas and Kuijlen, 2001]. Nowadays, optimal business processes can be achieved only by means of ICTs, that is, hardware and software products that support customer contacts such as campaign management systems, call-center support systems, and sales force automation systems [Schumacher and Meyer, 2004, section 2.2].

- (iii) **Collaborative CRM.** *Collaborative CRM* refers to all activities and tools suitable for controlling and coordinating all customer touchpoints that are used to organize communications between an organization and its customers [Schögl and Schmidt, 2002]. It is crucial to manage all communication activities in a multi-channel environment (see figure 4 on the facing page) to avoid overlapping communication activities, which – in the best case – convey coherent information and – in the worst case – communicate inconsistent information to customers [Schumacher and Meyer, 2004, section 2.2].

Remark (Multi-Channel Integration). Shaping communications between organizations and their customers in a multi-channel environment is crucial to CRM. In this context, Schögel et al. [2004] mention that customers are actually utilizing multi-channel choices, and – in addition – multi-channel customers are spending significantly more money in retailing than single-channel customers [Greenspan, 2003b]. Figure 4 on the next page depicts five channel types that customers use to communicate with an organization’s marketing, sales, and customer service departments [see Payne, 2003a]:

- (1) **Sales Force.** Refers to sales agents that communicate face-to-face with their customers. While sales agents can deal with complex non-standard queries and determine their customers’ needs and preferences by and by, their knowledge is hard to harness for organizations, since it is based on experience and instinct. Above all, sales agents represent an extremely expensive channel.
- (2) **Outlets.** Refer to physical branch offices where customers can inspect products and gain large amounts of information about products and services through conversations with sales personnel. While customers can probably resolve complex queries in branch offices, they can only do so during office hours and, moreover, have to travel to and from branch offices.

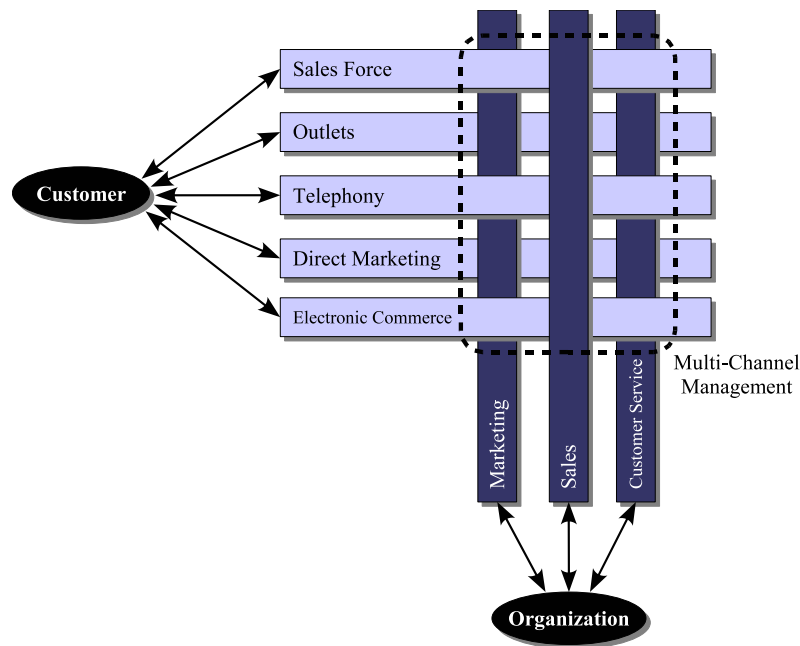


Figure 4: Multi-channel environment [derived from Payne, 2003b,a].

- (3) **Telephony.** Refers to call centers that provide tailored and cost-effective customer service backed by call routing systems and call center operation software. Call center agents can access customer profiles during conversations and gather additional information through standardized templates. While the amount of information exchanged is limited to verbal conversation, this channel can be made accessible 24/7 and is appropriate to deal with complex customer queries.
- (4) **Direct Marketing.** Primarily refers to directed mailing campaigns, which – when based on full and accurate customer information (for example, contact data and customer profiles) – can be tailored to customers’ individual interests and life events. While large amounts of information can be conveyed through this channel, it is not a fast and flexible medium in terms of customer responsiveness.
- (5) **Electronic Commerce.** *EC channels* refer to Internet-based channels that offer 24/7 access to organizations without employees being directly involved during communications and transactions.²⁴ With appropriate channel design, organizations can track individual customers in great detail (see section 3.2 on page 34). Not only will this channel type very soon be fulfilling the promise of mass customization, it also leads to a dramatic reduction in costs both in an organization’s marketing, sales, and customer service departments and in transaction processing (compare footnote 21 on page 17). If mobile ICTs are involved, customers can be offered location-based services, which relate to customizations depending on their current location.

²⁴While, for channel types item 1 on the facing page to item 3, employees are directly involved in communications and transactions, channel type item 5 does without direct employee involvement (for channel type item 4, the amount of manual processing required for the responses may be a significant factor). The more structure the communications and transactions in a channel have (in terms of recognizing and differentiating crucial business events), the less employee involvement is required during communications and transaction processing. Furthermore, structured communications and transactions simplify tracking customer behavior. So it could also be argued that the more structure a channel possesses, the more it can be regarded as an EC channel. The degree of employee involvement in a channel is evidence for the degree of structure it has.

2.2.4 Electronic Customer Relationship Management

The shift of consumers to utilize the Web for retail activities is not merely driven by the fact that organizations try to save money by leveraging the Web channel for their business activities. Instead, it is driven by a general evolutionary migration of consumers to the Web for all aspects of retail and business activities derived from the Internet becoming more available and consumers gaining more experience with ICTs [Feinberg et al., 2002]. This trend has a significant impact on the cultivation of customer relationships in a business environment where electronic channels are gaining in importance for all aspects of business transactions.

Today, due to advances in ICTs, the promises of one-to-one relationships, customer value analysis, and mass customization are becoming a reality [Peppard, 2000] and can be realized automatically without employees being directly involved in communications [Bernett and Kuhn, 2001]. As mentioned in section 2.2.2 on page 15, effective information management plays a crucial role in CRM and is critical for (i) product tailoring, (ii) service innovation (for example, tailored Web sites), (iii) providing consolidated views of customers, (iv) calculating CLVs of customers, and (v) establishing an integrated multi-channel capability (providing consistency of services across all channels). For Internet-based channels in particular, a shift from a transaction-based economy to a relationship-based economy is discernible [Romano Jr. and Fjermestad, 2003].

Consequently, organizations need the ability to track and manage Internet-based EC events that may demand immediate, personalized response. This is of special importance to an increasingly sophisticated customer base demanding higher service levels across multi-channel environments (see introduction of section 2.2 on page 13 and remark on page 19). As many organizations have a huge customer base, making any manual interactions during communications a tedious and expensive task, personalized response must be automated, minimizing employee involvement. The issue of deploying a CRM strategy for EC channels is addressed by the concept of ECRM.

2.2.4.1 Defining Electronic Customer Relationship Management

Definition 2.4 (Electronic Customer Relationship Management). ECRM restricts definition 2.3 on page 14 to EC channels (see item 5 on the previous page). ECRM comprises four aspects that are represented by the letters of the acronym ECRM [compare Peppard, 2000].

- (E) EC channels constitute the interface to customers for ECRM activities.
- (C) Channel management of a multi-channel environment of various channels (compare figure 4 on the preceding page). This includes unifying EC channels in terms of their technological infrastructure and tracking capabilities and coordinating all available channels.
- (R) Real Relationships in EC channels, that is, truly duplex communications (enabled by Internet technology), built on service excellence, added value, and convenience for customers.²⁵
- (M) Management of the total organization, that is, comprehensive back- and front-office synchronization and process integration.²⁶

²⁵Duplex communications refer not only to *direct* communications but also to *indirect* communications. Given a Web site, for instance, customers implicitly deliver behavioral data to organizations (indirect communication from customers to organizations). Organizations can then respond to customer behavior by adapting and personalizing Web sites, placing individual recommendations (indirect communication from organizations to customers).

²⁶More on strategic process alignment for CRM can be found in appendix section B.2 on page 126.

Remark. A first glance at the above definition gives rise to the objection that it is biased towards Internet technology and neglects other electronic channels. Today, electronic channels are steadily converging towards Internet technology, which is about to be established as an enabling technology for all electronic channels.

In terms of the *Open Systems Interconnection* (OSI) network reference model [compare Day and Zimmermann, 1995], this means that different electronic channels share the *data link layer*, the *network layer*, and the *transport layer* but differ in terms of the physical layer only [Walters, 2001, chapter 11]. For example, VoIP has the potential to replace today's wired telephone networks, if the physical layer is a wired network. Since the physical layer can be replaced by a wireless network, the same technology has the potential to replace today's cell phone networks, as well [Andritsou and Pronios, 2001]. From this perspective, the restriction to Internet technology in definition 2.4 on the facing page is justifiable in the long run.²⁷

Finally, the question arises as to which channels are relevant to ECRM. There is no simple answer to this question, since there are channels that cannot be uniquely assigned to one of the channel types in figure 4 on page 19. Nor is it possible to clearly determine whether or not employees are directly involved in communications and transactions. The degree of structure of channels depends heavily on organizational issues. Direct marketing, for example, can be realized through e-mail campaigns. Each e-mail may contain individual URLs that can be tracked upon invocation (representing a business event, for example, "order additional information" or "buy recommended product"). Since no direct employee involvement is necessary during communications and transactions in this case, direct marketing over e-mail can be regarded as an EC channel, whereas paper-based direct marketing involves manual processing and hence cannot be regarded as an EC channel.

In the future, channels such as telephony, ICQ, or core e-mail may be transformed into more structured channels. Speech recognition, text mining, and other methods may help to structure communications in these channels and to identify crucial business events automatically. Currently, the WWW is the only true EC channel that can be configured to be profoundly structured: that is, it is possible to recognize and differentiate crucial business events at different levels of granularity (see section 3.2 on page 34).

2.2.4.2 ECRM Research Areas

Based on a review of current ECRM research, Romano Jr. and Fjermestad [2003] identify five major research areas in their research framework for ECRM, which is depicted in figure 5 on the next page. The research framework divides ECRM research into the following areas: (1) *markets*, (2) *business models*, (3) *knowledge management and knowledge discovery*, (4) *technology*, and (5) *human factors*, each of which is briefly discussed in the following.

(1) **Markets** [compare Schögl and Schmidt, 2002; Romano Jr. and Fjermestad, 2003]. The Internet can be regarded as a marketplace instrument that offers mechanisms to allocate resources among participants. Allocations of market resources are currently transaction-based rather than relationship-based. ECRM research in this area primarily investigates how to integrate ECRM systems into markets in a way that they interact both effectively and efficiently. This includes research about building sustainable customer relationships over time with ECRM, research about how market share can be gained with ECRM, and research that investigates the shift of power from sellers to buyers within EC channels.

²⁷Bernett and Kuhn [2001] contrast traditional call centers with call centers based on Internet technology. The latter offer services going beyond answering calls that come in from switched telephone networks: for example, ICQ instant messaging or video conferences.

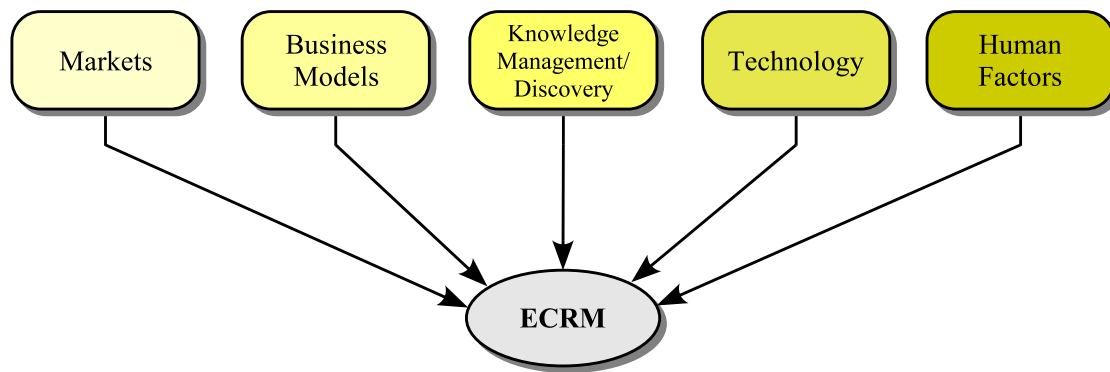


Figure 5: ECRM research framework [compare Romano Jr. and Fjermestad, 2003].

- (2) **Business Models** [compare Romano Jr. and Fjermestad, 2003; Feinberg et al., 2002]. Research in this area looks at how to design business models that are suitable to address the challenges posed by the shift towards EC channels. This includes the following aspects: (i) *Customer interaction*, that is, informational content, pull and push mechanisms, and communication channel choice. (ii) *Customer added value*, that is, mass customization, personalization, and economic incentives. (iii) *Customer profiling*, that is, collection and analysis of information about customers and value-added exchange of information. (iv) *Trust*, that is, strong branding, sensitive use of customer profiles, and security precautions. (v) *Virtual communities*, that is, information exchange about products and interests and market segment profiling.
- (3) **Knowledge Management and Knowledge Discovery** [see Schaarschmidt et al., 2001]. Conducting business across EC channels allows new data gathering strategies, calling for scalable methods to collect, analyze, process, and understand the collected data. Knowledge management and knowledge discovery methods need to be explored and developed in order to enable organizations to analyze large amounts of data from EC channels. Furthermore, the gathered data must be transformed into valuable insights and into information and benefits for both organizations and their customers. In order to achieve this goal, many organizations build data warehouses consisting of customer data (demographics and psychographics), customer activities (buying and browsing activities), and product data.
- (4) **Technology** [compare Romano Jr. and Fjermestad, 2003]. ECRM involves extensive use of ICTs to shape interactions with customers now and – even more – in the future. Hence, the following research questions concerning ECRM enabling ICTs arise: (i) ‘What will be the effect of technology on service quality (in terms of reliability, responsiveness, assurance, and empathy)?’ and (ii) ‘Is customer loyalty altered when customers interact with technology instead of employees?’ Apart from the technological influence on customer relationships and customer-facing processes, enabling ICTs for ECRM are themselves the subject of ECRM research – from a mere technological perspective.
- (5) **Human Factors** [compare Romano Jr. and Fjermestad, 2003]. From the point of view of the customer, EC channels change the way humans interact while spending money to acquire products and services. Behavioral aspects of ECRM are covered by virtual communities, interactions among customers, and interactions between customers and organizations. Additionally, emotional customer experience has effects on a number of issues: for example, customer satisfaction, trust and confidence, retention, willingness to interact and share information, actual purchases, attitudes, opinions, and customer recommendations.

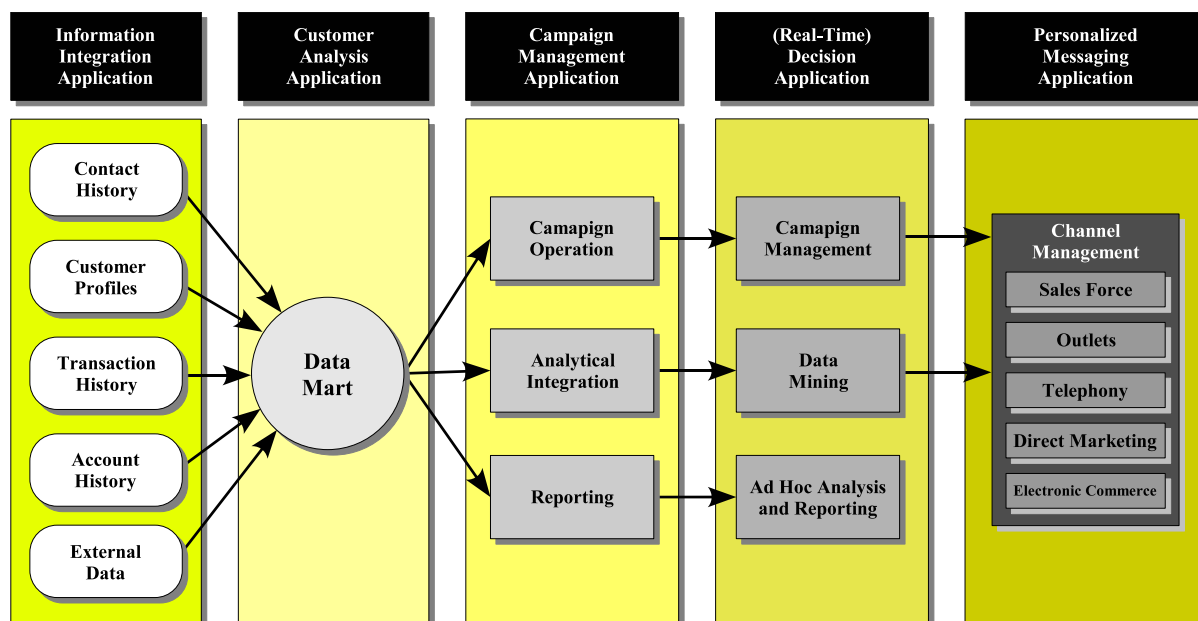


Figure 6: High-level framework for ECRM systems [adapted from Pan and Lee, 2003].

Remark. Figure 6 depicts the conceptual relationship among key ECRM applications. It can be regarded as a high-level framework for ECRM systems. According to Pan and Lee [2003], consolidating all customer-related information into a single view is a fundamental prerequisite of ECRM solutions. In order to realize this consolidated view, a multi-channel input stream that populates the single view with data from various sources is created, as shown in figure 6. This single view facilitates information-sharing across channels, creates a complete picture of the customer base, and helps to create meaningful multi-channel dialogs with customers. Moreover, it is the basis for ECRM strategy implementation. All analytical pursuits that address the five strategic CRM activities mentioned in section 2.2.2 on page 15 build on the single view that consolidates all available data.

In terms of research efforts and results, this thesis can be assigned to the ECRM research area item 3 on the preceding page *knowledge management and knowledge discovery*. This area defines the field of activity of this thesis, which is derived in chapter 3. Moreover, the research area item 2 on the facing page *business models* and the research area item 4 on the preceding page *technology* come into play in chapter 3, which discusses WUA – an approach toward analyzing the behavioral data of customers from the Web channel.

2.3 Summary

This chapter discussed EC and identified a long-term shift from traditional retailing to B2C EC within the next years. This shift will fundamentally change markets and cause the transition from seller markets to buyer markets. In section 2.1.3.1 on page 8 and section 2.1.3.2 on page 9, drivers for EC have been identified and it was concluded that EC aggravates the shift of power towards consumers in a market that is being subjected to profound changes at the same time. Rapid adoption of new ICTs, a globalized competitive environment, extreme market transparency, and changing customer habits have been identified as key drivers for this shift. It was then concluded that organizations must respond to new market conditions with a

differentiation or focus strategy, since cost leadership is neither sufficient nor sustainable. Following Scribner [2001], it was stated that retailers' inability to differentiate themselves from their competitors has been a long-standing issue for retailing, aggravated by the emergence of EC. Customer relationships can hence be viewed as the only defensible competitive advantage in retailing.

Then, the question arose as to how a differentiation or focus strategy can be established for EC. In section 2.2 on page 13, CRM was presented as an instrument to implement differentiation and focus. In section 2.2.1 on page 14 and section 2.2.2 on page 15, the basic principles of CRM were discussed and a taxonomy for CRM was introduced. The notion of collaborative CRM brought up the issue of multi-channel management, and the channels relevant to EC have been characterized and identified.

Further, in section 2.2.4 on page 20, ECRM has been recognized as a discipline to address differentiation in EC. It was highlighted that ECRM has the following benefits for organizations [compare Chatranon et al., 2001]: (i) more effective marketing efforts, (ii) more effective customer interactions, (iii) long-term profits from continuous customer relationships, (iv) shared customer knowledge throughout organizations through unified data collection and data storage, (v) reduction in service costs and related process costs, and (vi) more efficient and more effective sales activities. Last, *knowledge management and knowledge discovery* has been identified as a relevant ECRM research area for this thesis, and, by means of the high-level ECRM system architecture of figure 6 on the preceding page, creating a single view of customers has been identified as a key issue.

This is the starting point for the next chapter, which focuses on the Web channel – currently the only true EC channel that has the potential to realize automated real relationships with minimal employee interaction during communications. To this end, WUA is introduced as an instrument to deploy an ECRM strategy for the Web channel, and WUSAN is introduced as a framework to deploy a *closed loop* for WUA, thereby allowing for the implementation of differentiation for EC.

Chapter III

EFFECTIVE WEB USAGE ANALYSIS

The previous chapter discussed how the concept of ECRM can be applied to EC in order to implement differentiation for EC channels. The Web channel has been identified as the only true EC channel in terms of its capabilities to automatically collect comprehensive, fine-grained behavioral data about customers – a prerequisite for successful ECRM. Depending on a Web site’s underlying technical architecture, it is possible to track customers flexibly and precisely in an unprecedented way so as to acquire customer knowledge, purchases, browsing patterns, usage times, and personal preferences [Jian-guo et al., 2003].

Hence, it is essential to investigate how increased customer knowledge can be generated in the Web channel, and it is necessary to address the ECRM research challenge of knowledge management and knowledge discovery given the specific constraints of the Web channel. To this end, in section 3.1, the foundations of WUA are discussed, providing the basis for analyzing the Web channel. Then, in section 3.2 on page 34, data collection methods for capturing behavioral data in the Web channel are examined, as thorough data collection is crucial for the creation of a single view of customer behavior. Finally, in section 3.3 on page 41, WUSAN is introduced, and the central problem for this thesis is identified: the question as to how data transformations for WUA can be handled effectively with WUSAN.

3.1 Foundations of Web Usage Analysis

In view of the Web channel, many organizations have focused more on Web site traffic than on achieving strategic ECRM goals. However, the number of visitors to a Web site does not necessarily correlate with the number of real customers or the quality of customer relationships [Hochsztain et al., 2003]. WUA aims at analyzing usage behavior of Web sites from two perspectives: (i) the *data mining perspective*, which makes use of data mining methods and algorithms, and (ii) the *metrics perspective*, which applies metrics and descriptive statistics to measure whether strategic goals have been achieved.

This section investigates the notion of WUA by embedding it into its current research context. First, section 3.1.1 discusses the basics of *Web mining*, a higher-ranking discipline contributing to WUA, then section 3.1.2 on page 32 precisely defines the term WUA, and finally, section 3.1.3 on page 33 summarizes its primary application for ECRM *Web personalization*.

3.1.1 Introduction to Web Mining

Srivastava et al. [2004] propose the following data-centric definition of Web mining. Its sub-area *Web usage mining* describes one of the two central perspectives of WUA.

Definition 3.1 (Web Mining). *Web mining* is the application of data mining techniques to extract knowledge from Web data, that is, Web content, Web structure, and Web usage data.

Depending on the three types of Web data, three sub-areas of Web mining have emerged in the research community (compare figure 7 on the next page), each of which is briefly defined in the following sections: (1) *Web content mining* in section 3.1.1.1 on the following page,

(2) *Web structure mining* in section 3.1.1.2, and (3) *Web usage mining* in section 3.1.1.3 on the facing page.

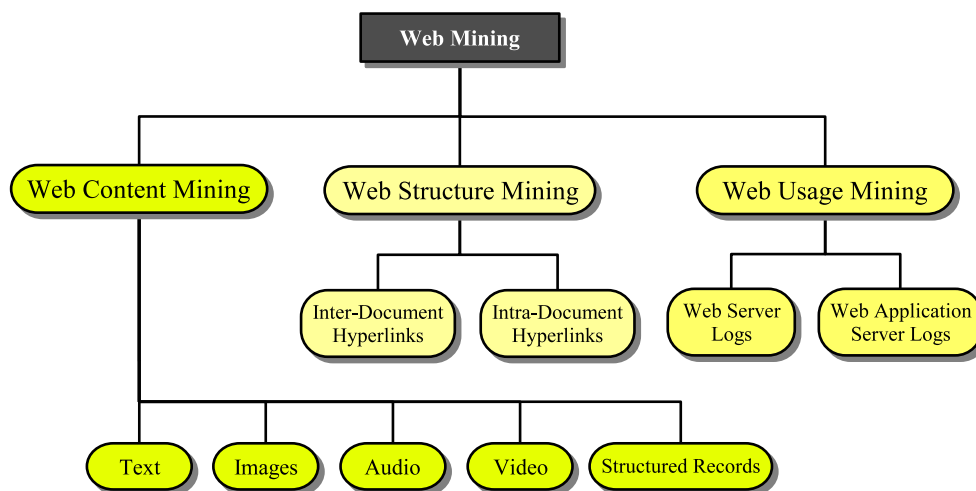


Figure 7: Data-centric Web mining taxonomy [adapted from Srivastava et al., 2004].

3.1.1.1 Web Content Mining

Definition 3.2 (Web Content Mining). *Web content mining* is the process of extracting useful information from the contents of Web documents [Srivastava et al., 2004].

In the above definition, content refers to the real data in Web pages, that is, *unstructured* data such as text, images, audio, and video, or *structured* records, for example, lists and tables [Srivastava et al., 2004]. Current Web content mining research has concentrated on text data and primarily applies text mining techniques to extract relevant facts from Web documents (referred to as *information extraction*) and to retrieve relevant Web documents (referred to as *information retrieval*) [Kosala and Blockeel, 2000]. Issues addressed in Web content mining are topic discovery, extracting association patterns, document clustering, and document classification [Jian-guo et al., 2003; Srivastava et al., 2004; Kolar and Joshi, 2004]. Google’s news service [GOOGLE-NEWS], which automatically extracts news from various information Web sites and groups them into predefined categories, is a paradigm for Web content mining.

3.1.1.2 Web Structure Mining

Definition 3.3 (Web Structure Mining). *Web structure mining* is the process of discovering structural information from the World Wide Web (WWW) [Srivastava et al., 2004].

In the definition above, structure refers to *hyperlinks* that connect different documents (*inter-document hyperlinks*) or contents within the same document (*intra-document hyperlinks*). Hyperlinks contain an enormous amount of latent human rating of the importance of the documents to which they point [Jian-guo et al., 2003]. This makes it possible to identify the relative relevance of documents in terms of *authorities*, that is, documents that provide the best source on a given topic and *hubs*, that is, collections of links to authorities [Chakrabarti et al., 1999]. Google’s search engine [GOOGLE-SEARCH] is a paradigm for Web structure mining. It implements the page rank algorithm [compare Page et al., 1998] and accounts for inter-document hyperlink structure of portions of the WWW in order to rank search results.¹

¹Chakrabarti [2002] contains a detailed introduction to Web structure mining and its applications.

3.1.1.3 Web Usage Mining

Definition 3.4 (Web Usage Mining). *Web usage mining* is the application of data mining techniques to discover usage patterns from Web usage data in order to understand and better serve the needs of Web-based applications [Srivastava et al., 2000].

As defined above, usage data refers to two categories of data that capture browsing behavior on Web sites [Srivastava et al., 2004]. (i) *Web server data* refer to logs collected by Web servers (for example, the Apache Web server [APACHE]). (ii) *Web application server data* refer to a more general category of specialized logs from Web application servers.² Not only do Web application servers establish sophisticated dynamic Web sites that model complex business events, they also provide an advanced tracking mechanism to capture such business events. Srivastava et al. [2004] and Cooley [2000, chapter 1] regard Web usage mining as a *process* that consists of three phases: (1) the *preprocessing phase*, (2) the *pattern discovery phase*, and (3) the *pattern analysis phase* (see figure 8).

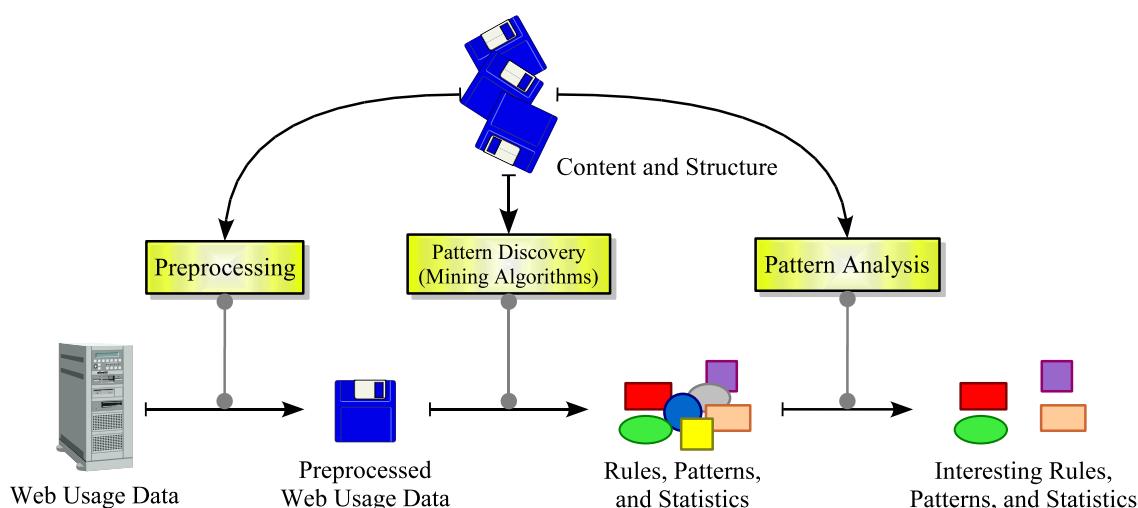


Figure 8: The Web usage mining process [adapted from Srivastava et al., 2004; Cooley, 2000].

- (1) **Preprocessing Phase.** During the preprocessing phase, data are cleaned from noise, inconsistencies in the data are cleared, and data from various data sources are integrated and consolidated so as to serve as input for the pattern discovery phase. The preprocessing phase is often the most time consuming and computationally intensive phase within the Web usage mining process [Anand et al., 2004]. It addresses a number of issues concerning the special characteristics of different log types. These issues need to be resolved before data mining algorithms can be applied. Srivastava et al. [2000] regard preprocessing as a key issue for Web usage mining. The primary tasks to be addressed in this phase are (i) *data cleaning*, that is, removing irrelevant information from logs and (ii) *transaction identification*, that is, grouping sequences of user requests into logical units, for example, business events, or user sessions³ [Cooley et al., 1997].⁴ The concrete preprocessing tasks

²Compare definition 3.6 on page 37.

³A *user session* (or simply *session*) is a sequence of consecutive page views before the user explicitly logs out or times out [Hu and Cercone, 2004].

⁴Transactions refer not only to purchases but also to a variety of other user actions within sessions, for example, “add product to wish list” or “browse product” [Mobasher, 2004].

are contingent largely on the log type, on the business questions to be addressed, and on the concrete data mining algorithms to be applied.

Cooley [2000] and Mobasher [2004] provide a detailed survey of preprocessing techniques required for various Web usage mining methods, especially Web personalization.⁵ The research community has examined many special problems associated with preprocessing for standard Web server logs⁶, most of which can be avoided by deploying a more sophisticated tracking mechanism.⁷

(2) **Pattern Discovery Phase.** During this phase, knowledge is discovered by applying data mining and statistical analysis techniques to preprocessed data. Most standard data mining techniques and algorithms can be transferred unchanged to the Web usage mining domain, since domain-specific problems must be addressed earlier during the preprocessing phase.

(i) **Statistical Analysis.** As for all data analysis tasks, descriptive statistics are an essential part of Web usage mining in order to get “a feeling for the data” before and after the preprocessing phase. Additionally, standard statistical techniques can be employed to gain knowledge about visitor behavior: for example, discovery of frequently accessed contents and Web pages, average view times, common entry and exit points, and statistics on sessions and users [Mobasher, 2004].⁸

Another form of statistically analyzing integrated Web usage data is *online analytical processing* (OLAP).⁹ The source for OLAP is a multi-dimensional data warehouse that integrates usage data and additional EC data at different aggregation levels for each dimension [Mobasher, 2004]. OLAP analysis can be used to analyze preprocessed and transformed data from a hierarchical and aggregated perspective, thereby allowing changes in aggregation levels along each dimension during analysis. It is most often applied to clickstreams that are stored in data marts with a special structure.¹⁰

(ii) **Association Rule Analysis.** Association rule analysis, or more precisely *market basket analysis*, has emerged as a popular tool for mining large commercial transactional databases. Given a set of binary variables X_1, \dots, X_n , $X_j \in \{0, 1\}$, a subset $I \subseteq \{1, \dots, n\}$ has to be found such that

$$\Pr \left[\bigcap_{i \in I} (X_i = 1) \right] = \Pr \left[\prod_{i \in I} X_i = 1 \right] \quad (3.1)$$

is large. Equation (3.1) is the *standard formulation* of the market basket problem. The set I is called *item set*. The estimated value of equation (3.1) is taken to be the

⁵Web personalization is discussed in section 3.1.3 on page 33.

⁶Sweiger et al. [2002, chapter 3] describe a variety of standard log formats relevant to Web usage mining.

⁷As this fact is discussed in section 3.2 on page 34, these Web server log-specific preprocessing tasks are skipped at this point (see, for instance, Cooley et al. [1999] for more details on these preprocessing tasks).

⁸A survey of Web usage statistics is given in Bertot et al. [1997]. Most tools available for Web usage mining merely calculate Web usage statistics, since they are limited to computing simple descriptive statistics from standardized log formats [see Wilde and Hippner, 2002]. Such tools are referred to as *Web log analysis tools*.

⁹An introduction to OLAP can be found in Han and Kamber [2001, chapter 2].

¹⁰It is important to note that OLAP tools do not automatically discover usage patterns in the data. In fact, the ability to find patterns or relationships in the data depends solely on the effectiveness of the OLAP queries performed against the data warehouse [Mobasher, 2004]. However, the effort needed to prepare data for OLAP is not in vain, since its data repository can be used as input for data mining algorithms.

fraction of observations in the database for which equation (3.1) on the preceding page is true:

$$\widehat{\Pr} \left[\prod_{i \in I} X_i = 1 \right] = \frac{1}{N} \sum_{\ell=1}^N \prod_{i \in I} x_{\ell i}. \quad (3.2)$$

N is the number of observations, and $x_{\ell i}$ is the value of X_i for the ℓ -th case. Equation (3.2) is called *support* $S(I)$ of item set I and can be transformed to

$$S(I) = \frac{|\{T \in \mathcal{T} : I \subseteq T\}|}{|\mathcal{T}|} \quad (3.3)$$

where \mathcal{T} is the set of observations (that is, the set of transactions) and $|\mathcal{T}| = N$. In association rule mining, a lower support bound l is specified, and one seeks all item sets I_k that can be formed from the variables X_1, \dots, X_n with

$$\{I_k : S(I_k) > l\}. \quad (3.4)$$

Each high-support item set I_k equation (3.4) is cast into a set of *association rules* $A \Rightarrow B$ with $A \cap B = \emptyset$, $A \cup B = I_k$, and $S(A \Rightarrow B) := S(I_k)$. Association rules are then assessed by two measures: (i) *confidence* and (ii) *lift*. The former is defined as

$$C(A \Rightarrow B) = \frac{S(A \Rightarrow B)}{S(A)}, \quad (3.5)$$

which is an estimate of $\Pr(B|A)$, and the latter is defined as

$$L(A \Rightarrow B) = \frac{C(A \Rightarrow B)}{S(B)},$$

which is an estimate of the association measure

$$\frac{\Pr(A \text{ and } B)}{\Pr(A)\Pr(B)}.$$

In addition to classic market basket analysis for EC, association rule analysis can be applied to the Web usage mining domain so as to discover pages and contents that are frequently referenced together in sessions. The presence or absence of rules not only helps to restructure Web sites and to address the five essential CRM activities mentioned in section 2.2.2 on page 15 (for example, by means of recommendation or personalization systems [Mobasher, 2004]) but also assists in optimizing high-performance dynamic Web sites by realizing prefetching strategies [Srivastava et al., 2000].¹¹

- (iii) **Sequence Analysis.** Given a set of customer transactions, each of which consists of the attributes *customer ID*, *transaction time*, and *transaction item*, for instance, purchased products or invoked Web pages, no customer may have more than one transaction with the same timestamp. A *sequence* is an ordered set of transactions that may contain each transaction type only once or repeatedly – depending on the concrete sequence analysis algorithm. Sequence analysis seeks to extract sequences fulfilling

¹¹A detailed introduction to (generalized) association rule analysis is provided by Hastie et al. [2001, section 14.2], and association rule algorithms are discussed in Agrawal and Srikant [1994] and Hipp et al. [2000].

a given support threshold, referred to as *large sequences*. It is related to market basket analysis but is even more complex, for the order of transactions matters.¹²

More formally, a *sequence* of items $\Sigma = (\sigma_1, \dots, \sigma_n)$ occurs in a transaction $T = (t_1, \dots, t_m)$, $n \leq m$ if there exist n positive integers $1 \leq a_1 < \dots < a_n \leq m$ and $\sigma_\ell = t_{a_\ell}$, $\ell = 1, \dots, n$. A sequence Σ is *contiguous* in T if there exists an integer $0 \leq b \leq m - n$ and $\sigma_\ell = t_{b+\ell}$, $\ell = 1, \dots, n$. In contrast to a sequence, each pair of adjacent elements $\sigma_\ell, \sigma_{\ell+1}$ in a contiguous sequence must appear consecutively in a transaction T that supports the pattern [Mobasher, 2004].

Given a set of transactions \mathcal{T} and a set $\mathcal{S} = \{\Sigma_1, \dots, \Sigma_k\}$ of frequent (contiguous) sequences over \mathcal{T} , the *support* of each Σ_ℓ is analogously defined as

$$S(\Sigma_\ell) = \frac{|\{T \in \mathcal{T} : \Sigma_\ell \text{ is (contiguous) subsequence of } T\}|}{|\mathcal{T}|}, \quad (3.6)$$

corresponding to equation (3.3) on the previous page. The *confidence* of the rule $A \Rightarrow B$ with A and B being (contiguous) sequences is defined as

$$C(A \Rightarrow B) = \frac{S(A + B)}{S(A)},$$

where $+$ denotes the concatenation operator. Since there is an analogy between association rule analysis and sequence analysis, association rule algorithms can be modified and extended for sequence analysis [Agrawal and Srikant, 1995].

In the Web usage mining domain, sequence analysis can be applied to predict user visit patterns (in order to take actions such as online advertising and online recommendations based on sequences) and to analyze frequent (contiguous or non-contiguous) paths in sessions [Srivastava et al., 2000].

- (iv) **Path Analysis.** Path analysis refers to an analysis technique developed exclusively for Web usage mining. It is about analyzing the clickstream paths taken by users during their sessions on Web sites. Regarding each click as a transaction, sequence analysis could be used to discover sequences of frequent page invocations consisting of pages that are not necessarily directly connected by hyperlinks. Unlike sequence analysis, path analysis investigates only sequences of linked page invocations and analyzes complete Web graphs with pages being *nodes* and hyperlinks being *edges*. It combines Web graphs with usage data and makes it possible to determine frequent traversal patterns, most frequent paths, and significant entry and exit pages.

The standard approach in path analysis is to analyze sample Web usage data and to calculate a Web graph with node weights representing the number of page invocations. This Web graph can then be evaluated to extract most frequent paths of a given length at varying starting nodes [Berkhin et al., 2001]. A more general approach to path analysis is proposed by Berendt and Spiliopoulou [2000]. The authors introduce the notion of a *generalized path* (referred to as a *g-sequence*) that may contain wildcards in its string representation. It is then possible to query the underlying Web graph in order to discover navigation paths fulfilling additional constraints that have been modeled by the generalized path string.¹³

¹²Agrawal and Srikant [1995] provide a detailed introduction to sequence analysis.

¹³This approach has been implemented in the “Web Utilization Miner” [compare HYPKNOWSYS].

- (v) **Regression and Classification.** Let $X \in \mathbb{R}^p$ denote a real-valued random input vector and $Y \in \mathbb{R}$ a real-valued random output variable with joint distribution $\Pr(X, Y)$. A function $f(X)$ for predicting Y given values of the input vector X is sought. A *loss function* $L(Y, f(X))$ is required for penalizing errors in prediction, and by far the most common and convenient is *squared error loss*

$$L(Y, f(X)) = (Y - f(X))^2$$

[Hastie et al., 2001, chapter 2]. The criterion for choosing f is the expected squared prediction error $E(Y - f(X))^2$, which yields the *regression function*

$$f(x) = E(Y|X = x).$$

Regression is the task to find a good estimate \hat{f} of the regression function f .

Classification is the discrete case of regression and its task is to map observations into one of several predefined classes. An important task prior to classification is extracting and selecting features that best describe the properties of a given class.¹⁴ In the WUA domain, the idea is to develop a profile of users belonging to a particular class, which is the basis for further efforts to achieve the ECRM goals [Srivastava et al., 2000]. Pabarskaite [2003] proposes decision trees to predict future user actions, especially pages leading to the termination of sessions.

- (vi) **Cluster Analysis.** Cluster analysis aims at grouping a collection of objects into subsets or *clusters* such that objects within each cluster are more closely related to one another than those assigned to different clusters [Hastie et al., 2001, section 14.3]. Central to cluster analysis is the notion of *similarity* (or *dissimilarity*) between the individual objects being clustered. A clustering method attempts to group objects based on the definition of similarity supplied to it.¹⁵

In Web usage mining, cluster analysis can be used to segment customers (in view of their shopping behavior or demographics), page invocations (in order to reduce the number of possible page invocations for analysis), or sessions (in view of users' browsing and shopping behavior) [Mobasher, 2004].

- (3) **Pattern Analysis Phase.** The pattern analysis phase is designed to convert discovered rules, patterns, and statistics into actionable knowledge and insights into the Web site being analyzed [Cooley, 2000, chapter 7]. Extracted rules are evaluated and formatted in a way that is understandable to humans, for example, by means of reports or visualizations. Interpreting the discovered rules in order to gain a good understanding of the analysis domain is a crucial part of the Web usage mining process [Ma et al., 2000]. It is then necessary to draw conclusions from postprocessed knowledge and to derive concrete actions that support the ECRM goals stated in section 2.2.2 on page 15.

It is a general data mining problem that the thresholds of algorithms must be set low enough to ensure that all patterns of potential interest are discovered. Consequently, hundreds or thousands of patterns and rules are returned by the algorithms. It is then a great challenge to separate interesting rules from those that are not particularly useful. The concept of

¹⁴An introduction to regression and classification is presented in Hastie et al. [2001].

¹⁵Grabmeier and Rudolph [2002] and Jain et al. [1999] provide introductions to cluster analysis algorithms and underlying similarity measures.

interestingness characterizes patterns that were previously unknown to analysts; yet this concept is extremely subjective and hard to quantify [Cooley, 2000, chapter 7].

Although research efforts aiming at quantifying interestingness in specific application domains and for specific data mining algorithms do exist [see, for instance, Cooley, 2000, chapter 7], there is no general solution to solve the problem of sifting out interesting rules. Therefore, the pattern analysis phase primarily aims at organizing and interpreting discovered rules by experienced analysts with sufficient domain knowledge. Analysts make use of various visualization techniques for data mining that help them to summarize and to abstract the discovered rules and models.¹⁶ Visualization heavily depends on concrete algorithms and model representations. Alternatively, analysts make use of query mechanisms such as SQL in a relational database management system (RDBMS) or apply OLAP techniques to analyze data mining patterns [Han, 1997].¹⁷

3.1.2 Web Usage Analysis for ECRM

Definition 3.5 (Web Usage Analysis). *WUA* is the process of analyzing Web site usage behavior by means of data mining and descriptive statistics. It is applied not only to supervise whether an organization's Web channel conforms to and supports its ECRM goals but also to infer and implement actions that help to achieve these ECRM goals.¹⁸ *WUA* consists of two sub-areas: (1) *Web usage mining* as defined in definition 3.4 on page 27 and (2) *Web reporting*, which refers to calculating descriptive statistics and business metrics from Web usage data.

Based on the results of *WUA* efforts, organizations can improve and optimize their Web sites with regard to their strategic ECRM goals. Section 3.1.3 on the facing page discusses how *WUA* can contribute to achieving these goals. Accomplishing ECRM goals is an iterative process rather than a straightforward matter. It involves experiments about how to alter *customer-facing functions*. Srivastava et al. [2004] consider Web sites to be an experimental apparatus for this purpose in that they not only provide the ability to measure human behavior at the micro-level but also offer the convenience of conducting various experiments on usage behavior with relatively little effort, once a sophisticated *WUA* architecture has been established – along with advanced Web site management. To this end, an elaborate technical Web site architecture is a key prerequisite. In this context, technical imperatives for data collection for *WUA* are discussed in section 3.2 on page 34.

Remark. In principle, the *WUA process* is identical to figure 8 on page 27. Since definition 3.5 considers the inference and implementation of actions in view of achieving strategic ECRM goals as an integral part of *WUA*, a *deployment phase* should be added to figure 8 on page 27

¹⁶Fayyad et al. [2001] is a good introduction to visualization techniques for data mining.

¹⁷This refers to the *mining then cubing* approach, that is, data mining is applied to data residing in a data warehouse, and particular mining results can then be further analyzed by OLAP operations [Han, 1997]. A complementary approach is the *cubing then mining* approach, that is, OLAP operations are used to select portions of a data warehouse for data mining. Another complementary approach is the *cubing while mining* approach, that is, OLAP operations are invoked during data mining to perform similar data mining operations on different abstraction levels.

¹⁸Hochsztain et al. [2003] precisely define the notions required to make a strategy measurable. (i) An *objective* is a desired end result or condition expressed in measurable terms that can be achieved by the successful performance of business or functional processes. (ii) A *goal* (or target) is the criterion by which the accomplishment of an objective is measured. Every objective must have a quantifiable goal. (iii) A *strategy* is a method or procedure to accomplish the related objective and the desired goal. (iv) A *performance measure* is an indicator built into a strategy that can measure progress towards satisfying the related strategy.

as the last phase. This phase turns the Web usage mining process into a *closed loop*, referred to as the WUA process. Following Chapman et al. [2000], the WUA process is an *iterative* process that is often traversed several times during concrete projects (with jumps back and forth between consecutive phases).

Kohavi and Provost [2001] propose the following five criteria that are relevant to successful data mining. Ideally, all of them should be satisfied, yet they are seldom present in real-world applications. But if the chance arises to alter and influence the boundary conditions of data mining projects, the list below should be accounted for within the constraints given.

- (1) **Data with rich descriptions.** For example, extensive customer records with many potentially useful attributes allow data mining algorithms to search beyond obvious correlations.
- (2) **Large volume of data.** The large model spaces corresponding to rich data demand many training instances to build reliable models.
- (3) **Controlled and reliable data collection.** Manual data entry and data integration from legacy systems are notoriously problematic. Fully automated data collection is considerably better and results in more consistent and more complete raw data.
- (4) **Ability to evaluate results and measure return on investment.** Evaluating changes and tracking their effects is hard, expensive, and takes a long time. But this process is necessary to prove the benefits of data mining capabilities and increase their acceptance. As mentioned before, this criterion can be accounted for more easily in WUA.
- (5) **Ease of integration with existing processes.** Although interesting insight is often discussed in data mining projects, concrete action is rarely taken, since legacy systems and inflexible domains make it hard to apply new knowledge and to improve existing processes.

Kohavi [2001] states that the above criteria can be met by the Web channel in an EC environment. In such an environment and with proper design of a Web site, large volumes of clickstream data can be collected efficiently for the purpose of data mining, and a large number of attributes can be made available (see section 3.2.2 on page 37). Ideally, a Web site's tracking capabilities are fully customizable and can be automated, reducing the noise inherent to the collected data – making further manual processing obsolete. Furthermore, EC is an actionable domain, that is, it is possible not only to conduct analyses but also to deploy the results in terms of *embedded data mining* models that can be integrated into a flexible EC architecture (compare item 4 on page 44). A Web site can be set up as an experimental laboratory that facilitates prompt success measures. Considering that many core EC processes are already online, they can be altered more easily compared to offline processes. The section below discusses *Web personalization*, which is the primary and most thoroughly researched practical application of WUA for ECRM in EC channels.

3.1.3 Web Personalization

Web personalization is defined as any action that adapts information or services provided by Web sites to the needs of particular users or user segments, taking advantage of the knowledge gained from users' navigational behavior and individual interests and making use of the content and structure information of Web sites [Eirinaki and Vazirgiannis, 2003]. According to Nasraoui [2005], the primary goals of Web personalization are (i) converting browsing Web site

users into customers, (ii) improving Web site design and usability, (iii) improving customer retention and loyalty, (iv) increasing cross-sales by recommending items related to the ones being considered, and (v) helping visitors to quickly find relevant information on Web sites and to make the results of information retrieval and search functions aware of the context and specific user interests. This means that Web personalization addresses the five essential CRM activities of section 2.2.2 on page 15. Mobasher et al. [2000] distinguish four types of Web personalization:

- (1) *Manual decision rules* allow personalization by manual intervention of Web site administrators and usually with the cooperation of users. Typically, static user models are obtained through user registration, and a number of rules are specified manually concerning Web contents that are provided to users with different static models [Pierrakos et al., 2003]. Decision rules are frequently created with *decision trees* and complemented by experts.
- (2) *Collaborative filtering* typically takes explicit information in the form of user ratings or user preferences and returns information that is predicted to closely match users' preferences through a correlation engine [Breese et al., 1998]. This approach is based on the assumption that users with similar behavior (for example, users that rate or browse similar objects) have analogous interests.
- (3) *Content-based filtering* relies on the similarity of contents of Web documents to personal profiles obtained explicitly or implicitly from users [Meteren and Someren, 2000]. It applies machine learning methods to Web contents, primarily texts, so as to discover personal user preferences.
- (4) *WUA* can reduce the need for obtaining subjective user ratings or personal preferences conveyed during registration to realize Web personalization.¹⁹ WUA tasks for Web personalization comprise the discovery of association rules, sequences, page view clusters, user clusters, or any other data mining method operating on Web usage data [Mobasher et al., 2000]. The most common type of Web personalization is to recommend a set of objects to users while they browse a Web site. Recommendations are based either on the users' current sessions (that is, recommendations are provided in real-time) or on the users' (or customers') browsing or purchasing histories (that is, recommendations can be prepared offline). Recommended objects refer to hyperlinks, advertisements, pieces of information, and products and services, each of which is tailored to the perceived preferences of users as determined by their usage patterns [Mobasher, 2004]. Real-time Web personalization can be accomplished by matching active user sessions (possibly in conjunction with previously stored user profiles) with usage patterns discovered with WUA.

3.2 Data Collection for Web Usage Analysis

The question arises now as to what data can be collected for WUA and what technical prerequisites must be fulfilled for this purpose. Data collection at EC Web sites includes *clickstreams* (page views and session information), *customer registration* attributes, and *order transactions* [Kohavi et al., 2004]. WUA research has so far concentrated on projects based on standard *Web server logs*. This can be attributed to the fact that Web server logs are relatively easy to access and obtain. A Web server such as the Apache Web server [APACHE] is easily accessible for

¹⁹Pierrakos et al. [2003] and Eirinaki and Vazirgiannis [2003] provide detailed overviews about the state-of-the-art of Web personalization.

most research institutions. However, Web server logs bring about a number of problems for WUA (see section 3.2.1). These problems must be addressed during the preprocessing phase and can largely be avoided if a more sophisticated Web application server environment that offers better tracking and logging capabilities is available (compare section 3.2.2 on page 37).

3.2.1 Web Server Logs

Web server logs are one possible form of server-side Web usage data collection for WUA. As depicted in figure 9, different users access a logical Web site simultaneously. The Web site is hosted on one or more physical Web (application) servers, each of which produces separate logs for each logical Web site (in case of Web servers). Alternatively, the cluster produces a consolidated log for each logical Web site (in case of Web application servers).

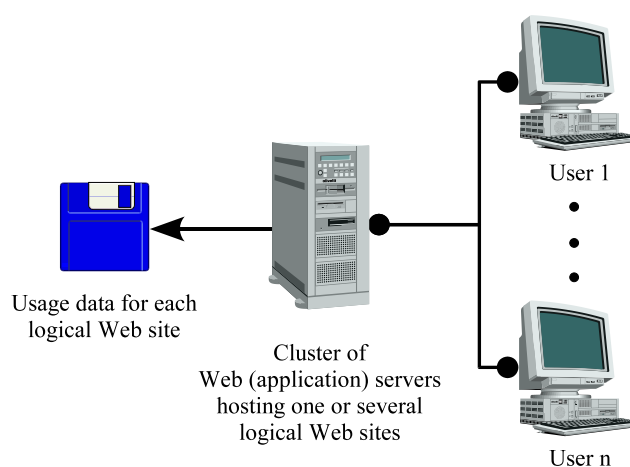


Figure 9: Server-side usage data collection for WUA.

The logging capability of mere Web servers is quite limited. Considering the Apache Web server [APACHE] as an example, a Web server offers a set of technical log variables (for instance, remote IP address, time of incoming request, and delivered bytes [see Bowen et al., 2002, chapter 24]) that can be flexibly combined by an administrator in order to create a proprietary Web server log or a Web server log conforming to one of the standardized Web server log formats mentioned in Sweiger et al. [2002, chapter 3]. All available logging variables are intended to support administrators in supervising a Web server's technical operation rather than in creating data for complex business analyses. Kohavi [2001] and Hu and Cercone [2004] specify a number of drawbacks of Web server logs in view of their capabilities for WUA:

- (1) *Web server logs do not identify sessions or users.* HTTP is by design a *stateless* protocol, which does not know sessions or users [Spiliopoulou, 2000].²⁰ In consequence, at least sessions must be reconstructed during preprocessing. This task has been researched in numerous contributions. Session creation heuristics and approaches to evaluate them are

²⁰The logical flow of an HTTP transaction sequence is as follows [compare Berghel, 2001]: (i) Establish a connection between a client and a Web server (in case of a Web browser, this typically amounts to a mouse click). (ii) Convey client's request to the Web server (for example, get data or execute a program). (iii) Web server fulfills client's request (if the client has sufficient permissions) and sends data back to the client. (iv) Immediately close the connection. The statelessness results from the last step, that is, as soon as a transaction cycle is complete, the connection between client and Web server is disconnected.

discussed in Berendt et al. [2001], Spiliopoulou et al. [2003], and Berendt et al. [2002]. Chen et al. [2002] address the problem of reconstructing *interval sessions* and *gap sessions*. Pitkow [1997] mentions the *caching problem*, which refers to the fact that proxy servers acting as an intermediary between users and the Internet in order to reduce network traffic buffer page invocations and lead to an incomplete mapping of partially cached sessions to Web server logs.²¹

- (2) *Web server logs lack critical business information.* Whatever business logic is implemented on (dynamic) Web sites (for instance, shopping carts or monetary transactions), it is not reflected in Web server logs (they lack especially content and user information and information provided by users in Web forms). Transactional data from back-end systems cannot be integrated into Web server logs and must be retrieved separately. Roughly speaking, information in Web server logs can be assigned to one of the following four categories [Bowen et al., 2002, chapter 24]: (i) *Address of the remote machine.* In the best case, a user's host name or IP address is logged. In the worst case, the host name or IP address of a proxy server accessed by multiple users is recorded. (ii) *Time of visit.* It is possible to include a timestamp that reflects the Web server's local time of incoming requests into each log record. (iii) *Resource requested.* This refers to log variables describing the requested resources such as the invoked URL and bytes transmitted. (iv) *Technical surveillance.* These are log variables that support Web server administrators in tracing technical errors, for instance, the HTTP status code that indicates whether requested resources have been delivered successfully.
- (3) *Web server logs must be consolidated.* If a Web site is distributed across a cluster of Web servers (possibly in different time zones), each Web server produces its own logs containing only those requests that were forwarded to it by a *load balancer*. In consequence, a session may be distributed across several physical Web servers and its traces spread across several logs with timestamps from different time zones. Much worse, Web server logs record every single resource requested. Since browsers translate one logical page request into a sequence of consecutive resource requests, Web server logs grow unnecessarily large. Both problems make Web server log consolidation an inevitable task for WUA.

The drawback item 1 on the previous page can be mitigated by deploying *cookies* [Sweiger et al., 2002, chapter 4]. A cookie is a key/value pair sent to a browser by a Web server to capture the current state of a session [Sit and Fu, 2001]. Browsers automatically include cookies in subsequent requests. From a technical point of view, a cookie is a piece of "transaction state" information left on the client machine before the HTTP transaction is completed. Cookies can authenticate users for multi-step transactions. EC Web sites such as Amazon [AMAZON] use cookies to associate the users with their shopping carts or sessions. Although many Web servers have the capability to track cookie information, users can annul the power of cookies by setting their browsers to generally refuse them or to limit their acceptance [Sweiger et al., 2002, chapter 4]. While the above disadvantage item 3 is a matter of how much effort one is willing to put into the problem, the shortcoming item 2 cannot be mitigated by Web server logs and must be resolved with a different logging approach (discussed in the next section).

²¹One of the reasons why there is so much research on Web server logs and sessionization is that Web server logs – as mentioned before – were designed to debug Web servers and not to provide useful information for data mining [Kohavi et al., 2004]. Hence, it is necessary to invest considerable effort into preprocessing Web server logs, a fertile ground for research activities.

Crucial business events can hardly be derived from mere URL invocations, even if all the page invocations involved are properly tracked. For dynamic Web sites, the same type of business event may result in a completely different sequence of URL invocations for different users. Since dynamic URLs encode numerous parameters meant to execute system-specific database queries or applications, it is extremely laborious to infer business events without detailed knowledge about the underlying Web site generation mechanism. However, if this knowledge is available, it is still less burdensome to leverage the internal logging capabilities of the underlying Web application server to track business events rather than to evaluate a multitude of parameters in numerous URL invocations.

3.2.2 Web Application Server Logs

Ansari et al. [2001] and Hu and Cercone [2004] emphasize the need for data collection at the application server level (instead of data collection at the Web server level) to mitigate the three problems mentioned in the previous section. Equally important, Ansari et al. [2001] mention the need to integrate all WUA activities into an EC architecture in order to significantly reduce the time spent for preprocessing. To fully understand the general architecture proposed by Ansari et al. [2001], which has been confirmed by Kohavi et al. [2004], it is helpful to demonstrate the basic principles of Web application servers. Their basic range of functionality addresses the three fundamental problems related to Web server logs from the previous section.

Definition 3.6 (Web Application Server [adapted from WHAT-IS]). A *Web application server* (referred to as an *application server*) is a server program on a computer in a distributed network that provides the *business logic* for application programs. It employs one or more Web servers as an interface to access and execute all implemented applications. A Web application server is frequently viewed as part of a *three-tier architecture* (see figure 10), consisting of an HTML browser (referred to as the *front-end* or the *client tier*), which operates as a graphical interface to the Web server, an application server (or business logic server, referred to as the *middle tier*), and a database and/or transaction server (referred to as the *back-end* or *database tier*).

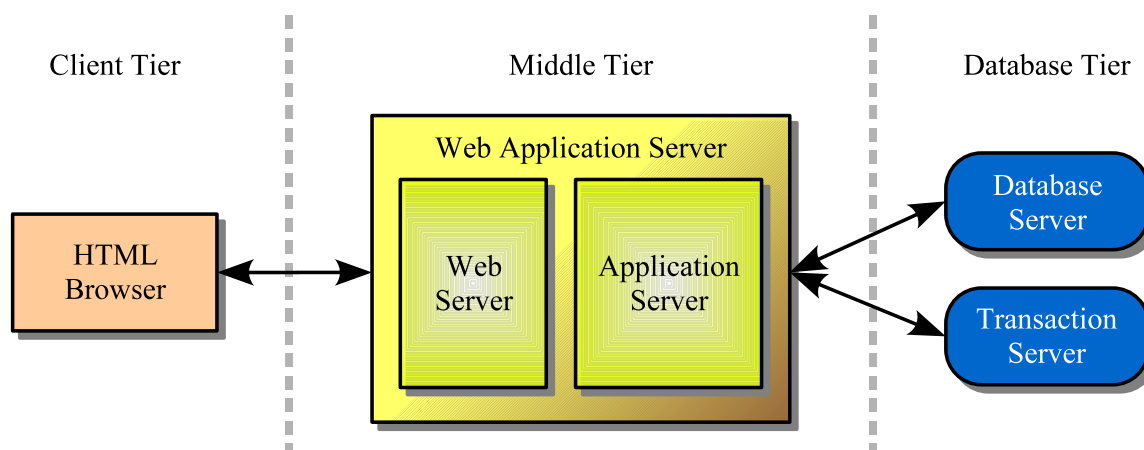


Figure 10: Web application server architecture [adapted from Mariucci, 2000].

Web browsers support an easy-to-create HTML-based front-end for users. Web servers provide different ways to route parameterized requests to application servers and to send back a modified or new Web page to users. These approaches include the Common Gateway Interface (CGI), Microsoft's Active Server Pages (ASP), and Sun's Java Server Pages (JSP),

each of which provides a mechanism for creating dynamic Web pages. The core technology of an application server is a standardized platform for component-based multi-tier enterprise application development. Currently, two basic concepts dominate the commercial market: (i) *Microsoft's .NET/ASP Web services and business application development framework* [MICROSOFT-.NET] and (ii) *Sun's J2EE/JSP Web services and enterprise application development framework* [SUN-J2EE].²² Taking J2EE/JSP as a paradigm of application server functionality, two fundamental features that are crucial to mitigate the three problems related to Web server logs, mentioned in the previous section, can be identified (compare Sweiger et al. [2002, chapter 3] and Kimball and Merz [2000, chapter 2]):

- (1) **Logging.** The JDK 1.4 and its successors implement the logging API [see Horstmann and Cornell, 2005a, chapter 11], which is a framework to implement flexible and powerful logging capabilities.²³ This logging framework enables logging of events in any Java application running on the application server. High-level business events, for example, “add product to shopping cart” or “check out”, are implemented within the application server framework and can be customized to issue a log entry upon transaction completion. This leads to clean, high-level logs of crucial business events occurring during sessions. The log format is proprietary and can be configured to be plain text, XML²⁴, or a database record.
- (2) **Session Tracking.** J2EE provides the *servlet session tracking API* [compare Hall and Brown, 2003, chapter 9] that handles session tracking. It combines two techniques:
 - (i) If users permit them, session tracking can be done with *cookies* [Sweiger et al., 2002, chapter 3]. Cookies are generally used for the following purposes [Hall and Brown, 2003, chapter 8]: (a) *Identify users during EC sessions.* Even returning users can be recognized. (b) *Avoid username and password* on Web sites where a login is required to conduct transactions. (c) *Customize Web sites.* This is, for instance, relevant to portal Web sites. Such Web sites offer various customizations that can be coded into a cookie concerning contents, layout, and services. (d) *Realize focused advertising.* This is primarily accomplished through shared cookies, leading to strong disputes about privacy [compare Chapman and Dhillon, 2002]. The purpose behind cookies is convenience for users and added value for Web site owners. The benefit of cookies for WUA is the fact that sessions can be associated to certain users as long as the cookies are stored on their machines.

²²Although the high-level architectures of both frameworks follow similar concepts, the details differ significantly and reflect two fundamentally different corporate philosophies (rather than strategies). Both frameworks are predominant in commercial business environments. SMEs or non-profit organizations that cannot afford a major investment in terms of money and technical excellence tend to realize dynamic Web sites based on WAMP or LAMP environments. That is, they set up a Web server that conforms in principle to figure 10 on the previous page but with reduced functionality in comparison to commercial application servers. A WAMP/LAMP environment consists of at least an Apache Web server [APACHE], the MySQL database [MYSQL], and the PHP hypertext preprocessor [PHP] to create dynamic pages, installed on a Windows or Linux environment. Recently, more sophisticated open source software products are being developed based on WAMP/LAMP environments, for example, Web content management systems such as TYPO3 or online shops such as OS-COMMERCE. Such off-the-shelf software makes it easy to create complex dynamic Web sites even for SMEs and individuals with a minimum of effort and costs. However, such packages do not offer business event logging by default. Nevertheless, PHP principally delivers the same functionality for logging and session tracking as its commercial counterparts [Hughes and Zmievski, 2001, chapter 12]. Hence, in principle, such open source environments can be customized to create logs with similar data quality as application server logs, but this involves large customization efforts.

²³LOG4J is an alternative logging approach for Java provided by the Apache Software Foundation.

²⁴For instance, LOGML [see Punin et al., 2002]

- (ii) If users decline cookies, the servlet session tracking API switches over to *URL rewriting*. With this approach, application servers automatically append an extra piece of information to every linked URL within each delivered resource (for example, a session ID). Upon clicking on the enriched hyperlinks, the client browsers implicitly send this piece of information back to the application servers, which then associate the received piece of information with data temporarily stored about sessions [Sweiger et al., 2002, chapter 3]. This method even works when browsers do not support cookies or when cookies are disabled.

It would be a tedious task to create complex dynamic Web sites based on application servers from scratch. For EC, typical tasks and processes (for instance, maintenance of product catalogs or shopping cart functionality) can be identified and implemented as a modular concept within an application server architecture. There are various commercial products that build on application server environments offering manifold modules to implement the business logic of EC Web sites. In addition, these products add standardized features to set up and administer large dynamic Web sites in a structured and efficient fashion and add capabilities in order to perform flexible, customized logging.²⁵

Apart from Ansari et al. [2001], Kohavi et al. [2004], and Hu and Cercone [2004], WUA research has not dealt with behavioral data stemming from application servers. The reason may be that application servers are not available and accessible to researchers offhand. Organizations operating application servers do not attach great importance to WUA²⁶ and are hesitant to grant research groups access to their usage data. However, the research activities mentioned above prove that these systems open up new vistas for WUA research.

3.2.3 Implications of the Preprocessing Phase and Open Issues

Obviously, application server logs have the potential to mitigate problem item 1 on page 35 to item 3 on page 36.²⁷ According to Kohavi et al. [2004], tracking critical business events, for example, successful and failed searches, shopping cart events, registration, initiation of checkout, order confirmation, and form field failures, have proven useful for WUA.²⁸ Moreover, as cited in item 1 on the preceding page, application servers can be configured to log events into a database, thereby additionally shortening the preprocessing phase [Kohavi et al., 2004].

Although application servers have the capability to avoid most preprocessing issues occurring with Web server logs, some open issues for the preprocessing phase that must be addressed still remain, even for application server logs:

- (1) **Session Timeout.** Session timeout duration is an important threshold for clickstream collection and sessionization. It determines the duration of inactivity after which a session would be considered timed out [Kohavi et al., 2004]. In an application server architecture, sessions automatically become inactive when the amount of time between consecutive client accesses exceeds a predefined threshold [Hall and Brown, 2003, chapter 9]. Accurate

²⁵It is beyond the scope of this thesis to provide an overview of the myriad of available commercial systems based on application servers. Sweiger et al. [2002, chapter 3] is a good jumping-off point for more information.

²⁶Deploying and operating an application server based system is a challenge on its own and itself consumes significant financial and human resources.

²⁷The *caching problem*, which has been mentioned in item 1 on page 35, is avoided by application servers since they produce dynamic URLs that cannot be cached by default (every user receives different dynamic URLs for the same type of business event). This effect is known as *cache busting* [Manjhi, 2000].

²⁸Furthermore, several performance measures related to the Web channel can be calculated only if statistics on crucial business events are available [Schonberg et al., 2000; Lee et al., 2001].

session tracking is a prerequisite for WUA. If long sessions are wrongly split, interrelated business events may end up in separate sessions. Generally speaking, session timeouts must be determined based on experiments for each concrete application domain.²⁹

- (2) **Detection of Web Robots.** *Web robots* are software programs that automatically traverse the hyperlink structure of the WWW in order to locate and retrieve information [Tan and Kumar, 2002]. Due to the volume and type of traffic they generate, Web robots can dramatically change clickstream patterns at Web sites, thereby skewing any clickstream statistics and data mining models [Kohavi et al., 2004].³⁰ Tan and Kumar [2002] and Sweiger et al. [2002, chapter 3] discuss a variety of approaches to detect non-camouflaging Web robots. Since some Web robots do not unveil their identity, Tan and Kumar [2002] propose a classification model that maps each session into one of two predefined classes: *Web robot* versus *human user*.
- (3) **Data Transformations.** According to Kohavi [2001] and Hu and Cercone [2004], there are two types of data transformations that need to take place during preprocessing: (i) data must be brought in from the operational systems to build a data warehouse, that is, data integration from multiple data sources (referred to as *ETL transformations*) and (ii) data may need to undergo additional transformations to answer specific business questions, for example, defining new attributes, binning, or aggregating data (referred to as *business intelligence transformations*). While the former transformations remain constant unless the tracked variables are altered, the latter transformations account for a great portion of the overall time spent during preprocessing.³¹ It is only natural to try to reduce this share by automating and simplifying all involved transformations. Kohavi et al. [2002] emphasize the need for these efforts, inasmuch as they have the potential to close the gap between business users' expectations towards user-friendliness of the complex analytic capabilities provided to them and their actual knowledge about technical details of the underlying data repository.

Remark. In addition to application server logging, other tracking mechanisms have been developed to address the three disadvantages of Web server logs mentioned on page 35. However, none of the alternatives has the capability to mitigate all these problems – or may even create new problems. For the sake of completeness, these alternatives are discussed in the following:

- (a) **Network Packet Sniffers.** A network packet sniffer monitors TCP/IP packets sent over the network to and from Web servers [DATANAUTICS]. Although they centrally gather network-level data (that is, low-level technical events rather than business events) that cannot be collected by any other data collection method (for example, “stop button activity”), and even though they eliminate the need to consolidate multiple logs, they have some serious drawbacks. First, they may not be able to track all packets in the event of a high load

²⁹Catledge and Pitkow [1995] propose the session timeout to be 1.5 standard deviations from the mean session duration (that is, about 25 minutes), whereas Kolari and Joshi [2004] recommend session timeouts to be no less than 60 minutes. Approaches to determine the end of user sessions in certain special cases are manifold. Sweiger et al. [2002, chapter 2] describe the method of *hyperlink redirection* for dynamic Web sites, which makes it possible to track users clicking on external hyperlinks that usually terminate sessions. Nonetheless, a session timeout is the most frequent means applied to determine when sessions actually end.

³⁰On high-volume EC Web sites, between 5% and 40% of the visits are caused by Web robots [Kohavi et al., 2004].

³¹Piatetsky-Shapiro et al. [1996] estimate the share of preprocessing to amount to 80% in industrial data mining projects.

on the Web server. Second, they cannot track encrypted connections at all [Kohavi, 2001]. This is a grave shortcoming in EC environments where large parts of the business logic are handled over encrypted connections.

- (b) **Page-Tagging.** A *page tag* is a small snippet of HTML code that is inserted into every single page delivered to clients [Henderson et al., 2002]. It consists of a JavaScript [compare Vincent, 2002] that is supposed to be executed by the clients' Web browsers. Embedded JavaScripts initiate a series of *one-pixel image requests* to a separate data collection server. To this end, they append various information as query strings to the one-pixel request URLs, which are then evaluated and logged by the data collection server. In principle, this server can produce clean logs that may contain whatever information can be accessed by JavaScripts on the client machine (for example, screen resolution and browser settings [see Sweiger et al., 2002, table 4.1]). Since JavaScripts are normally not executed by Web robots but by users' Web browsers even if the embedding page was loaded from a cache, the logs on the data collection server accurately track users' browsing behavior [Henderson et al., 2002]. The major drawbacks of the page-tagging approach are (i) its intrusive nature, which may cause serious privacy concerns [Sweiger et al., 2002, chapter 3], (ii) its sole dependency on JavaScript, which can be deactivated by users, thus resulting in very limited tracking capabilities, and (iii) its limitation to track page invocations only (that is, PDF requests or other non-page requests such as application invocations cannot be tracked).³²
- (c) **Reverse Proxy.** Pierrakos et al. [2003] discuss *reverse proxies* that can be configured to add session tracking capabilities to one or more Web sites without necessitating modification of these Web sites or reconfiguration of the underlying Web servers. This approach enables tracking across multiple Web servers with multiple logical Web sites and mitigates the sessionization problem of Web server logs. Nevertheless, reverse proxies are limited to logging URL invocations only. They are helpful if immediate improvement of tracking capabilities is called for or if it is too expensive to re-engineer the complete Web site [Kohavi, 2001].

3.3 A System for Effective Web Usage Analysis

As discussed in the previous section, an application server architecture significantly reduces the complexity of the preprocessing phase within the WUA process in comparison with other data collection techniques. This is due to the fact that the preprocessing phase profits from data with less noise and fewer inconsistencies. Nonetheless, even with application server logs, the preprocessing phase cannot be skipped completely. In section 3.2.3 on page 39, it was stated that there exist some challenging preprocessing problems regardless of the log type. A special emphasis was placed on data transformations for two purposes (see item 3 on the preceding page): (i) *ETL transformations* to populate a data warehouse with data from application servers and back-end systems and (ii) *business intelligence transformations* to transform data in order to answer specific business questions with WUA.

Both types of transformations should be configurable in a structured and straightforward manner in order to automate them and minimize manual interactions. Albeit this is basically

³²Shahabi and Banaei-Kashani [2001] propose a *remote agent* tracking framework that is similar to page-tagging but extends the tracking capabilities by sending a remote agent (that is, an executable) to the client machine. The remote agent is then fired on the client machine with explicit user consent and collects detailed client-side tracking data that is analogously sent to a data collection server. Clearly, privacy concerns and user bias represent the major drawbacks of this approach.

a common problem for data mining or business intelligence in general, it is notably a caustic problem for WUA, since this domain involves sheer amounts of data that make any manual interactions during the preprocessing phase tedious and complex [Anand et al., 2004]. Furthermore, when striving to close the loop of the WUA process on a regular basis by taking deployment seriously, automating preprocessing for both transformation types is compulsory.

This section centers a concrete WUA architecture that implements a new approach toward modeling both types of transformations based on a proprietary data warehouse. In section 3.3.1, the demands on a WUA architecture are summarized based on the discussions set out in chapter 2 and the previous sections. Afterwards, in section 3.3.2 on page 46, standards relevant to a WUA architecture are discussed, and it is shown how they contribute to concrete implementations and to closing the loop of the WUA process. Then, in section 3.3.4 on page 56, related research activities are summarized, and finally, in section 3.3.3 on page 51, WUSAN is introduced from a bird's eye view.³³

3.3.1 Prerequisites for a Web Usage Analysis Architecture

A WUA architecture should support the four phases of the WUA process, namely (1) the *preprocessing phase*, (2) the *pattern discovery phase*, (3) the *pattern analysis phase*, and (4) the *deployment phase* (compare figure 8 on page 27 and remark on page 32). Figure 11 on the next page depicts a general architecture for Web usage mining that covers all the relevant phases of the WUA process (except deployment) proposed by Cooley et al. [1997]. This general architecture, which does not allow for the details of how to actually implement its components, can be taken as a foundation for a concrete WUA architecture.

Although section 3.2 on page 34 clearly showed that data collection is a critical success factor for WUA, it is not considered part of the core WUA process. In practice, data collection is frequently driven by the technology that has been deployed to generate dynamic Web sites and its logging capabilities. Frequently, this technology cannot be altered or replaced by application server technology immediately, since it has been embedded into a complex environment of proprietary Web applications and back-end systems over years. Hence, a WUA architecture should be modeled in a way that creates no fundamental dependencies on certain log types to provide for changes in tracking over time.

Considering the four phases of the WUA process, the following prerequisites and requirements for a WUA architecture can be identified:

- (1) **Preprocessing Phase.** As mentioned in item 1 on page 27, the preprocessing phase primarily consists of *data cleaning* and *transaction identification*. Both tasks can be mitigated by a more sophisticated application server architecture as discussed in section 3.2 on page 34. In consequence, *data integration* can be considered the most important challenge to be addressed during the preprocessing phase, that is, ETL transformations. Furthermore, business intelligence transformations (referred to as “transformations” in figure 11 on the next page) must be modeled (compare item 3 on page 40).

To respond to the challenge of integrating large volumes of data from various data sources, deploying a data warehouse within the ECRM process is proposed by Pan and Lee [2003]. Although many authors agree on the benefits of using a data warehouse (i) to combine data mining and OLAP analyses [Han, 1997], (ii) to integrate various data sources and handle large volumes of data [Kimball and Merz, 2000], and (iii) to improve and leverage WUA [Joshi et al., 2003; Rahm and Stöhr, 2002; Zaïane et al., 1998], little is stated about how to

³³The discussion in this section has been in part adapted from Maier [2004].

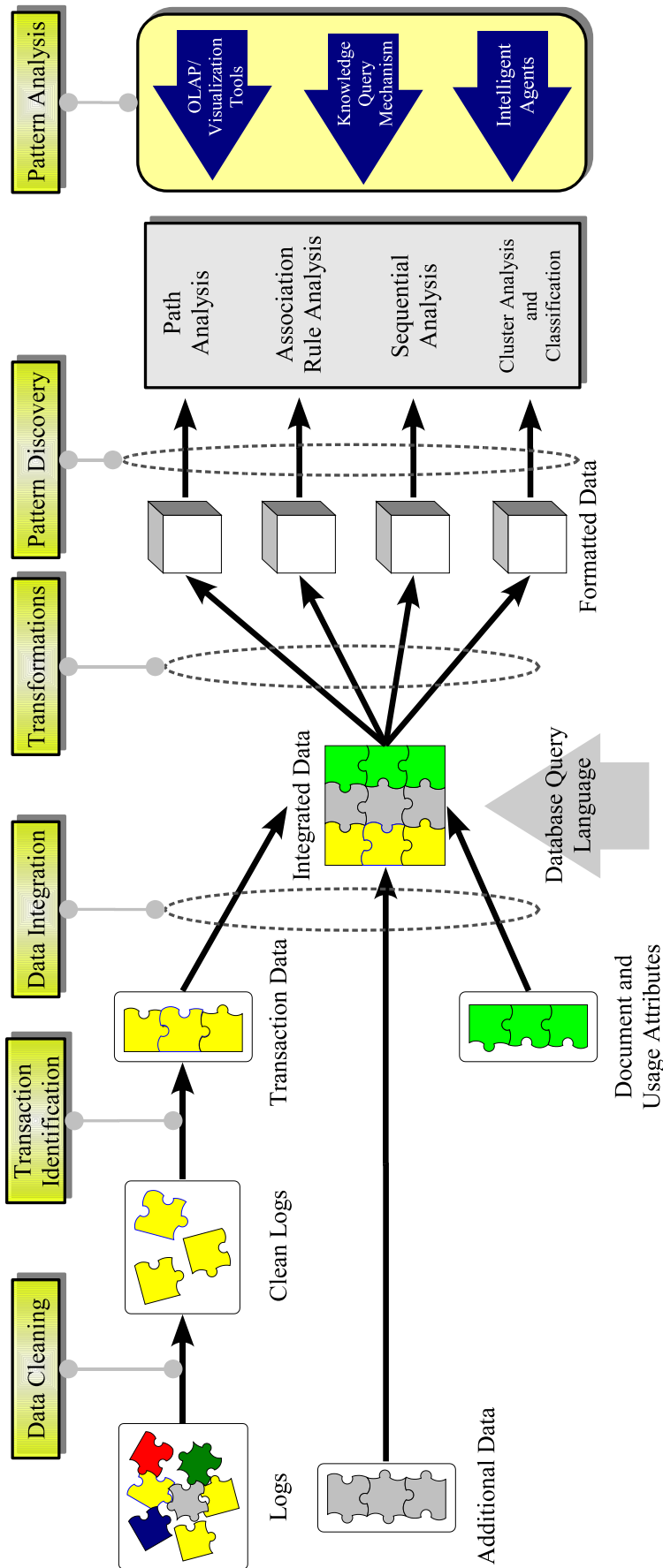


Figure 11: A general architecture for Web usage mining [adapted from Cooley et al., 1997].

populate and update a data warehouse by deploying a systematic conceptual model for the ETL process so as to reduce the time and costs needed for this task. According to Kimball and Caserta [2004, introduction], this can, in fact, account for up to 70% of the total data warehousing expenses.

For Web usage data in particular, which is continuously collected in large volumes, the ETL process must be deployed in a way that automates this task and minimizes manual interactions. Thus, it is necessary to provide a structured, practicable model of the ETL process that (i) supports the integration of multiple data sources, (ii) is robust in view of changing data sources, (iii) supports flexible transformations, and (iv) can be easily deployed.

In the preprocessing phase, it is also essential to access different kinds of data sources (for example, flat file logs, XML logs, or database records). These data sources must then be integrated into a data warehouse.³⁴

- (2) **Pattern Discovery Phase.** For the pattern discovery phase, it is mandatory to provide a common interface to access the data stored in the data warehouse. As this phase deals with frequently changing analytical questions, it is necessary to provide a means to run statistical and data mining algorithms on the data warehouse directly or to create relevant data sets with little effort. Furthermore, a WUA architecture should allow for the integration of third-party data mining algorithms. In view of definition 3.5 on page 32, OLAP is considered as an integral part of Web reporting (compare Kimball and Merz [2000, chapter 11] and Sweiger et al. [2002, chapter 9]), and the architecture should provide an interface to integrate an OLAP server.³⁵
- (3) **Pattern Analysis Phase.** In item 3 on page 31, it has been argued that this phase is next to impossible to quantify (since it is based on previous individual knowledge) and depends on evaluations by domain experts. As a consequence, a WUA architecture should provide resources that facilitate and support experts in interpreting the discovered patterns. Two techniques have been mentioned in item 3 on page 31: (i) *visualization techniques* and (ii) *querying techniques*, each of which should be supported in a WUA architecture.³⁶
- (4) **Deployment Phase.** For WUA, deployment primarily refers to the technical implementation of discovered rules and models, that is, making them actionable by embedding them into certain modules of an application server, for instance, a recommendation engine. Integrating data mining components into existing software systems is referred to as *embedded data mining* by Thess and Bolotnicov [2004, introduction and section 6.1.1.2] (also compare Witten and Frank [2005, chapter 14]). To facilitate embedding of models on a regular basis necessitates a precisely defined interface to embed models so as to reduce manual customizations.

³⁴Using a data warehousing approach for data integration and as a basis for all analytical activities is a *best practice* proposed by Kimball and Merz [2000] and Sweiger et al. [2002]. This practice is based on the insight that operational data and data for analytical purposes must be clearly separated [Hu and Cercone, 2004].

³⁵The main task of OLAP servers is to implement a multi-dimensional query language, for example, *multi-dimensional expressions* (MDX) [compare Spofford, 2001] and to provide a mechanism to query the data warehouse in terms of a *logical* multi-dimensional data model. They translate multi-dimensional queries into the query type of the underlying *physical* data model, for instance, SQL in case of ROLAP (see section 3.3.3.4 on page 54).

³⁶Along with visualization and querying techniques, *intelligent agents* are depicted in figure 11 on the previous page as a third alternative for the pattern analysis phase. This approach is based on the notion of *interestingness* of patterns, which involves the problems mentioned before.

First, the question arises as to which extent existing WUA tools conform to the prerequisites mentioned above, especially with respect to support for data warehousing and related data transformations for WUA. In a recent study, Maier and Reinartz [2004] took a closer look at state-of-the-art WUA tools, most of which do not completely cover all phases of the WUA process and their particular prerequisites.

Most tools are restricted to calculating detailed statistics from Web server logs, that is, they focus on Web reporting only. These tools parse Web server logs and produce static graphical reports showing activities by day and by time, top page accesses, least-accessed pages, server error code distribution, most commonly used browsers, and many more and focus on hits rather than users [Anand et al., 2004].³⁷ More sophisticated Web reporting tools assume that Web sites can be considered to be successful when their owners' objectives are satisfied, for example, conversion of visitors into (repeat) customers or increase in online sales of certain types of products and hence implement more complex business metrics [see Schonberg et al., 2000].

Baraglia and Palmerini [2002] discuss their WUA system "SUGGEST", which establishes a closed loop for Web personalization based on Web server logs. The system, which is implemented as a module of the Apache Web server [APACHE], comprises two components: (i) An *offline component* that aims at building the knowledge used for deployment by analyzing Web server logs. The main functions carried out by this component are preprocessing and pattern discovery. (ii) An *online component* that generates personalized content in real time based on the knowledge extracted in the offline component. The online component processes a request to the Web server by adding personalized content that can be expressed by hyperlinks, advertisements, or product information related to the current user. The system draws its knowledge from an incremental graph-partitioning algorithm that maintains clusters of related pages built according to users' navigation patterns. This approach is limited in view of handling dynamic Web pages and Web usage data other than Web server logs. Furthermore, all its components are restricted to addressing Web personalization with one proprietary approach only.

Povel and Giraud-Carrier [2004] propose their "SWISSANALYST" system, which addresses the issue of mapping the entire data mining process to a data mining tool. Although this system is not specific to WUA, it represents a step towards automating the overall WUA process. The current version of the system is built on WEKA, which is an attempt to support existing de facto standards for the data mining kernel but turns out to be a limitation in view of system performance: WEKA requires the data to fit in memory during analysis. The system centers around the CRISP-DM [see Chapman et al., 2000] and extends WEKA's functionality in view of supporting the goals of CRISP-DM by providing GUIs for each of its phases.³⁸

Clearly, Blue Martini's Web application server architecture [compare Ansari et al., 2001; Kohavi et al., 2004] covers the complete WUA process (deployment included) and integrates a proprietary data warehouse. However, this system is intended for complex EC Web sites and calls for significant investments in terms of money and technical knowledge (as most sophisticated WUA systems [Anand et al., 2004]). The system is therefore not an option for WUA projects with limited resources, especially academic research. Furthermore, the system accounts for data from its integrated Web application server only.

In consequence, a proprietary WUA solution is required to meet all of the discussed requirements for a WUA architecture. Since it would be too laborious to create a WUA architecture

³⁷ANALOG and AW-STATS are two exemplary open-source standard Web reporting tools.

³⁸Recently, the WEKA project has started activities to improve WEKA's ease of use by adding a new GUI that supports the process perspective of data mining. Furthermore, a few commercial data mining products also pursue the process perspective. All of these do not explicitly support WUA and hence only partially fulfill the prerequisites for a WUA architecture mentioned in section 3.3.1 on page 42 [compare Wilde and Hippner, 2002].

from scratch, as many components as possible should be realized with existing packages, in particular *open source software*.

The open source licensing model yields several advantages in comparison with commercial software products [compare Lerner and Tirole, 2002]: (i) *Customization and bug-fixing benefits*, since active open source projects automatically mature over time and can be customized or extended as needed due to the availability of all involved source code. (ii) *Manageable costs*, inasmuch as open source software is normally available at little expense, and support is provided by a community of developers and active users, often accessible at no charge. (iii) *No vendor lock-in and long-term project lifespan*, since even in case of the termination of active development of an open source project, the source code is still available and can be continued to be customized and extended for own account.

As a matter of principle, available standards should be utilized to design interfaces for a WUA architecture owing to their contribution to a reduction in the required integration efforts [Anand et al., 2004]. A common fundamental standard is, for example, XML, which is used in EC as an enabling technology that makes it possible for business documents, forms, and messages to be inter-operable and comprehensible [Meltzer and Glushko, 1998].³⁹

3.3.2 Standards Relevant to Web Usage Analysis

While there are no specific WUA standards, a variety of standards related to business intelligence systems based on a data warehouse and to data mining in general have been established (compare Leavitt [2002] and Barsegyan et al. [2004, chapter 8]). The former address the problem of modeling meta-data in a multi-vendor data warehousing environment (covered in section 3.3.2.1). The latter tackle the problem of integrating data mining and statistical models into other systems, for example, systems to support ECRM activities [Grossman et al., 2002] (discussed in section 3.3.2.2 on page 50). According to Grossman et al. [2002], standards for cleansing and transforming data – that is, standards related to the preprocessing phase of the WUA process – are only beginning to emerge.

3.3.2.1 The Common Warehouse Meta-Model

A typical data warehousing and business intelligence system is often described in terms of an *information supply chain* (ISC), a metaphor reflecting the fact that information in such a system flows from its sources through a sequence of refinements and data processing steps that are suitable for decision makers to support the decision-making process in organizations to its final state [Poole et al., 2002, chapter 2]. An ISC can be considered the equivalent of a supply chain for durable goods focusing on information as a “product”. It refers to the system depicted in figure 12 on the next page, which consists of the following components:

- (1) An *operational data store*, which is an information system that stores and integrates operational data (for example, transactions from EC Web sites).
- (2) An *ETL module* that supports all kinds of transformations of the operational data in order to process operational data in such a way that the outcome is more suitable for decision making.

³⁹According to Meltzer and Glushko [1998], the features of XML can be summarized as (i) a markup specification for creating self-descriptive data, (ii) a platform- and application-independent data format, (iii) a way to validate the structure of data, (iv) a syntax that can be understood by computers and humans, and (v) an incremental way to advance Web applications used for EC.

- (3) An *analysis module* that supports various analysis techniques on transformed operational data, including statistical analysis, reporting, visualization, and data mining.

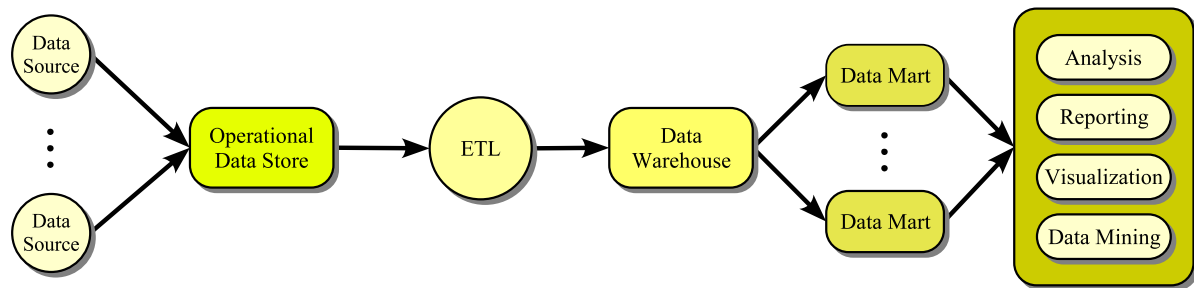


Figure 12: Information supply chain [adapted from Poole et al., 2002, chapter 2].

Remark. An ISC supports the transformation of raw data into strategic business information by means of a well-defined flow of data from initial data sources through a set of transformations. Special-purpose transformations provided by the ETL module enhance the transformed data's capability to serve as a strategic information base for decision makers. A strategic information base is usually realized with a data warehouse, the dimensional nature of which allows for sophisticated reporting along various business dimensions (compare footnote 34 on page 44).

Meta-data are the key to understanding what data actually mean: they are critical to all aspects of interoperability within an ISC environment, since they fuel the interfaces between its different components [Poole, 2001]. At present, an ISC primarily consists of a collection of software tools from various vendors, each of which has its own proprietary internal meta-data model, thus making mutual integration a difficult task. Meta-data integration within an ISC has so far been realized with two different approaches [compare Poole et al., 2002, chapter 2]:

- (i) *Point-to-point bridges.* This approach makes a specific software adapter necessary, whenever two components of the ISC interchange data (and hence require meta-data information). It may depend on a large number of adapters that mediate exclusively between two components on the meta-data level. As software tools within an ISC usually single out one aspect of ISC functionality, point-to-point bridges are a pragmatic but laborious response to the myriad of ISC supporting tools [Auth and von Maur, 2002].
- (ii) *Centralized meta-data repository.* This approach favors a central meta-data repository that stores meta-data in a proprietary format that can be accessed by all ISC components [compare Dinter et al., 1998]. Each component needs an adapter that translates its own proprietary meta-data model into the layout imposed by the central meta-data repository [Auth and von Maur, 2002].

Both approaches do not avoid strong dependencies on proprietary ISC components, as each provides its own meta-data model. Extending an ISC or exchanging single components is accompanied by programming several meta-data adapters from scratch.

The CWM [OMG-CWM] in figure 13 on the next page addresses the problem of incompatible meta-data through a coherent meta-data model (referred to as meta-model) developed by the *Object Management Group* [OMG]. It contrasts both meta-data integration approaches by proposing a product-independent external meta-model of resources, transformations, and data

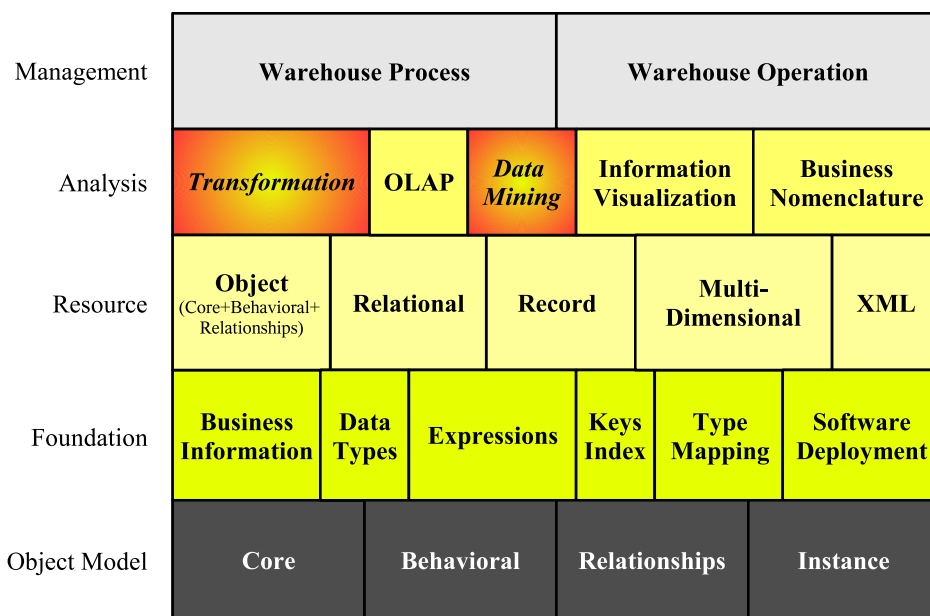


Figure 13: Layers of the CWM [OMG-CWM, chapter 5].

types that drives data access, data interchange, and type mappings among all relevant components of the ISC [Poole, 2001]. Each ISC component may realize its own proprietary internal meta-model but must conform to the CWM at all interfaces.

According to Poole [2001] and Poole et al. [2002, chapter 2], the CWM makes three contributions: (i) It provides a meta-model that defines shared meta-data for the data warehousing and business analysis domain. (ii) It provides a format to interchange and publish shared meta-data within an ISC. (iii) It models a UML API to access and discover shared meta-data within an ISC. Following Poole et al. [2003, chapter 3], the CWM conforms to three major design principles: (a) It uses *inheritance* to achieve reuse, that is, it employs *object-oriented design*, relying on the principles of inheritance [compare Horstmann and Cornell, 2005a, chapter 4]. (b) It ties meta-data definitions to physical data sources, that is, meta-data do not exist independently of concrete data sources. (c) It supports the creation of instance data objects for packages of the *resource layer* (see figure 13), that is, the CWM is a practicable meta-model that allows integration of programming languages into meta-data interchange.

The CWM in figure 13 is comprised of a number of constituent meta-models representing resources, analysis capabilities, data warehouse management, and foundational components of an ISC [Poole, 2001]. Figure 13 depicts the five layers of the CWM, each of which is comprised of a number of packages with constituent meta-models. Resources from the resource layer model relational databases, record-oriented databases, XML resources, and object-based data sources. The *analysis layer* defines meta-models for data transformations, OLAP, information visualization/reporting, business nomenclature, and data mining. The *warehouse management layer* consists of meta-models representing standard warehouse processes, activity tracking, and scheduling. Finally, the *foundation layer* provides common elements and services, for example, data types, keys and indexes, and expressions.

The CWM minimizes dependencies among packages of one layer [Poole et al., 2002, chapter 2]. Thus, each package of a certain layer depends only on packages from subordinate layers (aside from a few exceptions). Each package of the top three layers represents a constituent meta-model that corresponds to one important functional area of a typical ISC.

The real heart and purpose of an ISC is represented by meta-models comprising the analysis layer that covers business analysis concepts. This layer is relevant to ECRM activities. It describes services that operate on data sources and data targets described by packages from the resource layer. The following three packages from the analysis layer are adequately supported by open source ISC software tools or by open standards and provide a basis for deploying the WUA architecture discussed in section 3.3.3 on page 51:

- (1) **Transformation Package.** This package supports ETL and general object transformations. According to Thess [2004], it must be regarded as one of the most powerful assets of the CWM. Not only does it define modeling elements that can be used to specify source and target mappings and transformations between data resource models (that is, instances of the resource layer meta-models), it also defines source and target mappings and transformations between data resource models and any meta-model from the analysis layer [Poole et al., 2002, chapter 2].
- (2) **OLAP Package.** This package provides an OLAP model of essential OLAP concepts, analyzing data from the data warehouse in terms of cubes and dimensions [compare Han and Kamber, 2001, chapter 2]. Concrete instances are based on either the *relational package* or the *multi-dimensional package* from the resource layer and result in relational OLAP (ROLAP) or multi-dimensional OLAP (MOLAP), respectively. This package plays an important role even beyond the CWM: the *Java OLAP* (JOLAP) API [JSR-69] is an effort to develop a Java API for OLAP servers with JOLAP serving as a client API [Poole, 2001]. JOLAP makes use of the CWM OLAP meta-model to describe OLAP meta-data and ensures compatibility with the CWM.⁴⁰
- (3) **Data Mining Package.** Data mining tools are particularly effective in a data warehousing environment inasmuch as data warehouses provide for large quantities of cleansed business data suitable for data mining activities [OMG-CWM, chapter 15]. This package contains descriptions of the results of data mining activities by representing the models they discover and the attributes underlying the exploration [Poole et al., 2002, chapter 4].⁴¹ Similar to the OLAP package, the data mining package also plays an important role beyond the CWM. The *Java Data Mining* (JDM) API [JSR-73], for example, provides a Java API for business intelligence applications employing data mining techniques for knowledge discovery and analysis. It can be regarded as a reification of the CWM data mining meta-model that describes meta-data relevant to data mining activities, ensuring compatibility with the CWM [Poole, 2001].

Remark. It is beyond the scope of this thesis to discuss the CWM in detail. An overview from a conceptual perspective is provided by Poole et al. [2002], whereas Poole et al. [2003] discuss the CWM from a developer's perspective, going into the details of the individual CWM packages. OMG-CWM is the official documentation that describes all meta-models on the UML level.

⁴⁰JOLAP also defines query interfaces that support the formation and execution of OLAP queries along with the management and manipulation of multi-dimensional result sets [Poole, 2001].

⁴¹As XELOPES's data mining capabilities are based on the CWM data mining package, this package is a key prerequisite for all data mining activities within WUSAN.

3.3.2.2 The Predictive Model Markup Language

The *Predictive Model Markup Language* (PMML) is an XML standard developed by the *Data Mining Group* (DMG) [DMG-PMML] aiming at (i) describing and exchanging data mining and statistical models, (ii) describing meta-data for data sources that are used as input for models or transformations, and (iii) representing related preprocessing tasks that are required to transform data sources to conform to the prerequisites required by data mining and statistical models [Grossman et al., 2002]. PMML models build on XML and can hence be easily parsed and manipulated [Anand et al., 2004]. A typical application scenario for PMML in WUA is the offline creation of a classification model that is exported to PMML and imported into a recommendation engine of a Web site for real-time behavioral scoring. Both tools involved may have their own proprietary internal model representations but dispose of a conversion capability to map PMML to their proprietary models, and vice versa.

According to Grossman et al. [2002] and DMG-PMML, PMML comprises, amongst others, the following components (corresponding to XML tags):⁴²

- (i) **Data Dictionary.** Defines meta-data for attributes used in data mining and statistical models and specifies the types and value ranges.
- (ii) **Mining Schema.** Each model contains one mining schema that lists all the attributes used in that particular model. This is a subset of the attributes defined in the data dictionary. While a mining schema contains information that is specific to a certain model, the data dictionary contains data definitions that do not vary per model. The main purpose of a mining schema is to list the attributes needed in order to be able to apply the model.
- (iii) **Transformation Dictionary.** May contain one of the following elemental transformations: (a) *normalization*, that is, mapping continuous or discrete values to numbers, (b) *discretization*, that is, mapping continuous values to discrete values, (c) *value mapping*, that is, mapping discrete values to discrete values, and (d) *aggregation*, that is, summarizing or collecting groups of values, for example, sums and averages. This means that PMML 2.0 has only limited capabilities to describe data transformations. These are far from sufficient to describe the complex data transformations that can be modeled with the CWM transformation package.
- (iv) **Model Statistics.** Represent univariate statistics about the attributes contained in a model.
- (v) **Models.** Represent different data mining models: regression models, cluster analysis, decision trees, neural networks, Bayesian models, association rules, and sequential analysis.

The objective of PMML is to establish an open standard for modeling data mining results. Its models are independent of applications, platforms, and concrete data mining processes, which in turn facilitates the exchange of models among applications related to data mining.

Remark. In addition to the standards mentioned so far, there exist further established and emerging standards that may be useful for WUA, once commercial or open source tools fully support them. Grossman et al. [2002] mention (i) SQL/MM that comprises a part specifying an SQL interface to data mining applications and an API for data mining applications to access

⁴²The descriptions refer to PMML 2.0, since this PMML version is implemented in XELOPES 1.2.5, which is used for this thesis. The present version PMML 3.0 has been significantly extended [see DMG-PMML] and is currently about to be deployed in data mining tools [Thess and Bolotnicov, 2004, section 5.3.1].

data from RDBMSs compliant to SQL/MM [see Melton and Eisenberg, 2001] and (ii) Microsoft's OLE DB for data mining standard [MICROSOFT-OLE-DB] that defines an API for data mining for applications based on Microsoft's proprietary software development standards. Figure 14 depicts the relationships among the data mining standards that have been discussed above, in section 3.3.2.1 on page 46, and in section 3.3.2.2 on the facing page.⁴³

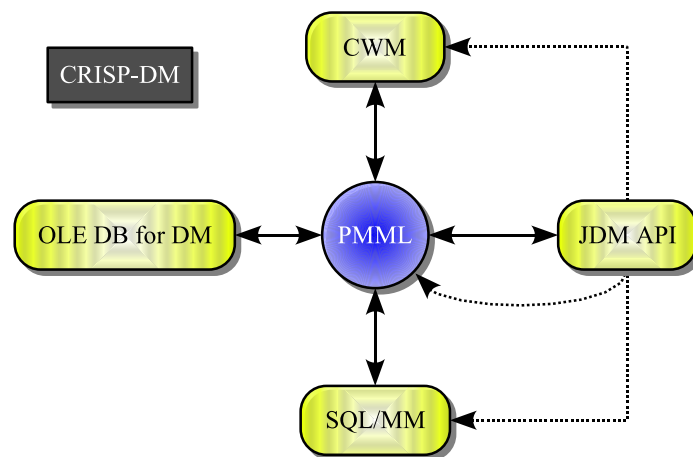


Figure 14: Relationships among the standards [Thess and Bolotnicov, 2004, section 5.3].

Furthermore, open source tools such as WEKA [WEKA] and XELOPES (compare Thess and Bolotnicov [2004] and Barsegyan et al. [2004, chapter 9]) with a high penetration in academic research can be regarded as de facto standards for data mining. The latter is based on the CWM and provides the foundation for all Java implementations for this thesis.

3.3.3 The Web Usage Analysis System (WUSAN) Architecture

Based on the prerequisites for a WUA architecture in section 3.3.1 on page 42 and drawing from the standards discussed in the previous section, it is now possible to outline WUSAN and delineate how its components are modeled. As mentioned, figure 11 on page 43 and figure 12 on page 47 both force the basic structure of WUSAN. This basic structure is reflected in figure 15 on page 53, which depicts its overall high-level architecture.

As illustrated in figure 15 on page 53, the system consists of four components, (i) the *data access* component, (ii) the *population (ETL)* component, (iii) the *data warehousing* component, and (iv) the *analysis* component, each of which is sketched in the following sections. All components fall back on the XELOPES Java data mining library [Thess and Bolotnicov, 2004], which is a data mining library based on the CWM (compare section 3.3.2.1 on page 46). As of version 1.2.5, which has been used for the WUSAN prototype, XELOPES realizes the two CWM packages that have been highlighted in figure 13 on page 48: (1) the *transformation* package and (2) the *data mining* package. Seeing that efficiently executing data mining algorithms and modeling data transformations for data mining are the two main tasks of XELOPES [Thess, 2004], the library is limited to implementing the transformation and data mining meta-models and skips all other CWM packages.⁴⁴ A central asset of the data mining package is the `MiningDataSpecification` class that models a collection of mining attributes

⁴³Dashed lines represent an “influenced by”-relationship.

⁴⁴Strictly speaking, dependent CWM packages from lower layers contributing to the transformation and data mining meta-models are modeled implicitly by XELOPES but are not relevant for end users since these packages do not constitute user interfaces for data mining tasks.

specifying how to interpret input data attributes [OMG-CWM, chapter 15].⁴⁵

Although this CWM class is intended to model input data for data mining algorithms, XELOPES makes use of it to describe data sources relevant to data mining that would – strictly speaking – have their own CWM meta-model. Since XELOPES focuses on data mining only (and not on the complete ISC as the CWM does), describing all data sources in terms of the data mining meta-model is a pragmatic simplification that reduces the library’s complexity for end users. This approach is simply passed on to WUSAN, as many standardized ISC resources (for instance, databases) do not yet conform to the CWM meta-models of the resource layer. In consequence, XELOPES’s variant of the `MiningDataSpecification` class can be considered to be a fundamental meta-model⁴⁶ to describe data throughout WUSAN, hence avoiding a meta-data repository or point-to-point meta-data bridges (compare item i on page 47 and item ii on page 47). Consequently, meta-data describing input data and transformed data can be exchanged without any further adaptations at any point in time over the `MiningDataSpecification` class.⁴⁷

3.3.3.1 The Data Access Component

Operational data and additional data sources can be accessed through *streams*. This notion refers to the XELOPES abstract `MiningInputStream` class (covered in detail in section 4.2.2 on page 68) that provides a mechanism for accessing data resources, the meta-data of which are modeled with the CWM `MiningDataSpecification` class [Barsegyan et al., 2004, section 9.6]. Streams implement a *cursor* and provide their data and meta-data through the `MiningInputStream` class record by record. A mixture of streams provided by XELOPES and streams developed especially for WUSAN provide access to a variety of data resources, for example, flat files conforming to various data formats, databases, and data residing in the memory (see section 4.2.2.2 on page 70).

3.3.3.2 The Population (ETL) Component

In order to manage the complex task of populating a data warehouse, various specialized ETL or data integration software packages that support the integration of heterogeneous data sources have been developed. Due to the previous lack of an overall model that integrates all kinds of middle-ware required for data warehousing, ETL and data integration tools have contributed to the myriad of poorly integrated systems in a data warehousing environment [Stonebraker, 2002].⁴⁸ According to Vassiliadis et al. [2001], the most prominent tasks of an ETL tool include (i) extraction of relevant information in data sources, (ii) integration of information from multiple sources into a common format, (iii) cleaning of the resulting data with regard to analysis purposes, and (iv) the propagation of data to the data warehouse.

It is this component which is realized with a new approach in WUSAN that is discussed in detail in section 4.3 on page 82. In section 3.3.4 on page 56, existing commercial ETL tools and research approaches toward modeling an ETL component are summarized and their drawbacks are clarified, working out the necessity to deepen research to address the challenge of modeling ETL for WUA.

⁴⁵This CWM class is discussed in more detail in section 4.2.1 on page 61.

⁴⁶The theory of the `MiningDataSpecification` class is discussed in section 4.2.1 on page 61.

⁴⁷This directly reflects the purpose of the CWM [compare Poole et al., 2002, chapter 2].

⁴⁸With the release of the CWM 1.0 [OMG-CWM] in 2001, this lack was addressed at least in theory. It will, however, take some time until the CWM is supported on a broad basis (or at least some portions of it).

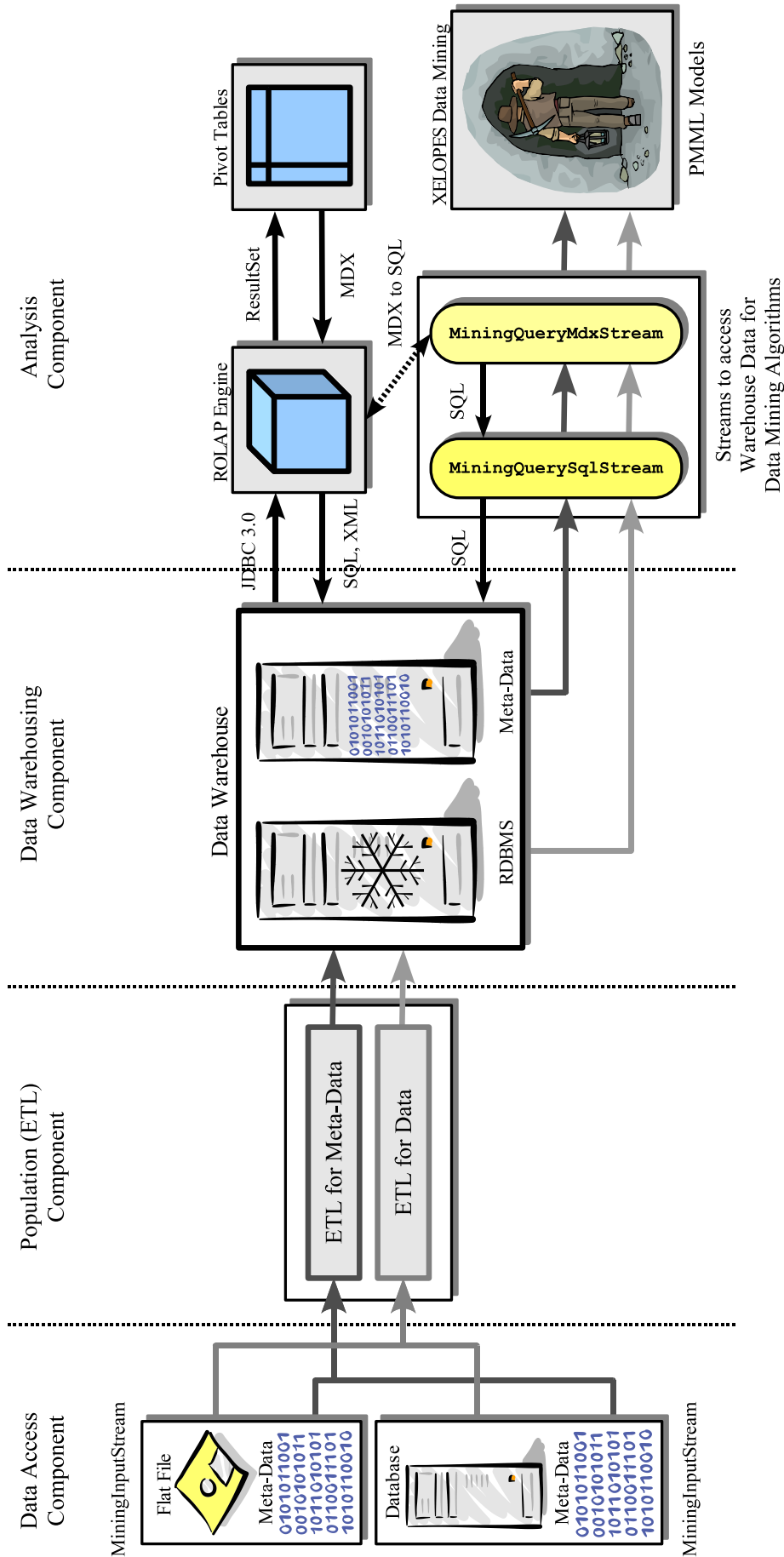


Figure 15: Overall architecture of the WUA system (WUSAN).

3.3.3.3 The Data Warehousing Component

The main task of the data warehousing component within WUSAN is to provide a flexible physical data model as a foundation for the analysis component (compare footnote 35 on page 44). WUSAN applies an off-the-shelf RDBMS, enforcing an ROLAP implementation for OLAP analysis within the analysis component in the next section. ROLAP provides a straightforward mapping of the multi-dimensional *logical* data model to the *physical* data model of a well-standardized RDBMS by using *star schemas* or related schemas [compare Martyn, 2004]. In contrast, mapping the logical OLAP data model to the underlying structures of a genuine multi-dimensional DBMS would be more complex, since, in that case, data is stored in multi-dimensional arrays (leading to enhanced performance compared to ROLAP if they can be stored completely in memory).⁴⁹ The decision in favor of ROLAP is confirmed by the study Gorla [2003], which concludes that – after considerable experience with OLAP – users prefer ROLAP systems owing to their flexibility and ability to handle complex queries in a business environment where analysis requirements change frequently.

It is not sufficient to simply integrate an RDBMS into WUSAN. In fact, the relational model of the integrated RDBMS must be extended to conform to the meta-model of the CWM data mining package. To this end, relational tables are modeled with the WUSAN-specific `MiningTableSqlStream` class (compare item 3 on page 71) that provides meta-data conforming to the CWM data mining package. Internally, this class stores stream meta-data that are modeled as a `MiningDataSpecification` in PMML format.⁵⁰

Remark. The actual design of relational schemas for ROLAP cannot be automated, for it involves semantic aspects and depends on available data sources. However, it can be systemized for specific application domains, for example, WUA. Kimball and Merz [2000] and Sweiger et al. [2002] both address this issue, which is the central topic in chapter 5 on page 95.

3.3.3.4 The Analysis Component

The analysis component comprises three constituents (compare figure 15 on the preceding page): (1) an *ROLAP engine*, (2) an *OLAP visualization tool*, and (3) the *XELOPES Java data mining library*, each of which is briefly discussed below.

- (1) **ROLAP Engine.** An ROLAP engine translates between the *logical multi-dimensional* data model and the *physical relational* data model of an RDBMS [Dinter et al., 1998]. Each incoming multi-dimensional query is translated into one or more relational queries that are optimized and run against the RDBMS. The ROLAP engine is a potential bottleneck in that OLAP SQL queries generally contain a large number of joins on large volumes of data and hence must be optimized and cached carefully so as to minimize response times.

⁴⁹According to Buzydlowski et al. [1998], a real standard for the implementation of MOLAP is lacking. As cited in item 2 on page 49, the CWM OLAP package only defines the general behavior of OLAP interfaces in terms of deliverable meta-data, neglecting implementation details due to its interface nature.

⁵⁰In the strict sense, the PMML `DataDictionary` tag is leveraged for this purpose (compare section 3.3.2.2 on page 50). This example of use shows that the PMML can be employed for various purposes, not only for model descriptions of data mining models [Grossman et al., 1999]. Basically, PMML and the CWM data mining package are two related, but independent, standards (compare figure 14 on page 51). Nevertheless, it is possible to map portions of the CWM data mining package to the PMML, and vice versa. As mentioned, the `MiningDataSpecification` class can be mapped to the PMML `DataDictionary` tag, and vice versa.

WUSAN applies the open-source Java OLAP server Mondrian [MONDRIAN-OLAP]. According to Grimes [2005], Mondrian is founded on an RDBMS and supports the multi-dimensional expressions (MDX) language for multi-dimensional queries [compare Spoford, 2001], as well as the JOLAP standard (see item 2 on page 49). Mondrian translates MDX queries into SQL queries and performs a query optimization prior to sending the SQL statements to the underlying RDBMS in order to create an OLAP result set.

- (2) **OLAP Visualization Tool.** JPivot [JPIVOT], an open-source companion project of Mondrian [MONDRIAN-OLAP], acts as a Mondrian client. It is a JSP custom tag library that renders OLAP tables and let users perform typical OLAP navigations such as *slice and dice*, *drill down*, and *roll up* [Grimes, 2005].⁵¹ Furthermore, JPivot operates on a Tomcat application server [JAKARTA-TOMCAT] that can be accessed by analysts through ordinary Web browsers.
- (3) **XELOPES Java Data Mining Library.** The XELOPES Java data mining library serves two purposes: (i) it provides access to the data stored in the data warehouse with the mechanism described in section 3.3.3.1 on page 52, that is, data can either be directly accessed by WUSAN's stream extensions or transferred into one of the more common streams discussed in section 4.2.2.2 on page 70⁵² and (ii) it provides the interface to integrate and run various data mining algorithms over the CWM data mining meta-model, for example, proprietary XELOPES algorithms or algorithms provided by WEKA [compare Thess and Bolotnicov, 2004, section 6.9.2]. In this context, XELOPES contributes to technical deployment (compare item 4 on page 44) by providing an interface for PMML import and export. As discussed in section 3.3.2.2 on page 50, PMML is a convenient language for importing and exporting data mining models between different systems of a decision support environment, especially data modeling and operational systems [Grossman et al., 1999].

In the case of WUSAN, this means that data mining models can be made operational by integrating their PMML representations into an application server, for example, in order to deploy a scoring model that rates the creditworthiness of new customers of an EC Web site.⁵³ This concept substantially simplifies closing the loop for the WUA process or any other instance of a data mining process.

In item 3 on page 31, it has been mentioned that visualization plays an important role during the pattern analysis phase. In item 2, OLAP visualization has been addressed by deploying an open-source package. Meanwhile, the problem of how to visualize data mining models remains. Due to the spreading of PMML, visualizers have evolved for PMML data mining models. The IBM DB2 Intelligent Miner Visualization [IM-VISUALIZATION], for example, presents the results of data mining algorithms and statistical functions. It comprises customized visualizers for cluster analysis, decision trees, and association rules. The input to the visualizers are PMML data mining models, that is, the visualizer can

⁵¹For typical OLAP operations compare Han and Kamber [2001, chapter 2].

⁵²For example, flat file streams or memory streams. Depending on the data mining task, it may be faster to run certain algorithms on flat file streams or memory streams instead of running them directly on database streams.

⁵³From a technical perspective, a deployment scenario could look like this: the XELOPES Java data mining library can be embedded into an application server based on J2EE (compare section 3.2.2 on page 37). Conforming to the CWM data mining meta-model, XELOPES implements an interface to apply data mining models that have been imported via PMML. Such data mining models can be computed with XELOPES or any other tool having the capability to export its data mining models into PMML. Then, any application deployed on the application server can make use of imported data mining models.

be invoked independently from different applications to present data mining models [IM-VISUALIZATION-HANDBOOK].

In summary, the WUSAN architecture in figure 15 on page 53 covers most of the prerequisites for a WUA systems of section 3.3.1 on page 42, provided that a structured, practicable ETL component is available. The latter aspect is investigated in the following section, which provides a detailed summary of research activities addressing the problem of modeling an ETL component.

3.3.4 Related Research Activities

ETL tools are pieces of software that manage the extraction of data from several data sources and their subsequent cleansing, customization, and insertion into a data warehouse [Vassiliadis et al., 2005]. According to Vassiliadis et al. [2001], the most prominent tasks of ETL tools include (a) identification of relevant information in data sources, (b) extraction of this information, (c) integration of information from multiple sources into a common format, (d) cleansing of the resulting data with regard to analysis purposes, and (e) the propagation of data into the data warehouse.

The commercial market for ETL and data integration tools offers a wide spectrum of solutions that have primarily the following things in common [compare Agosta, 2002; Kimball and Caserta, 2004; ASCENTIAL; INFORMATICA]: (i) they work with a proprietary meta-data model and meta-data management, (ii) they implement a proprietary transformation model, (iii) their configuration and set-up are very complex, (iv) they have complex hardware requirements and are thus not practicable for small- and medium-sized projects, (v) they do not have a standardized API, and (vi) they are autonomous and can only be combined with a restricted set of other systems in a data warehousing environment. Agosta [2002] states that ETL is unlikely to become an open, standardized technology solution and that proprietary approaches will remain dominant in the near future (in turn spurring researchers to make contributions toward finding a solution to standardizing and modeling ETL).

The most prominent commercial ETL tools are *Ascential DataStage* [ASCENTIAL], which also handles data quality assurance, *ETI Solution* [ETI], *Informatica PowerCenter* [INFORMATICA], and Microsoft's *Data Transformation Services* [MICROSOFT-DTS], which only integrate smoothly with Microsoft's own standards (ODBC, OLE DB) but which are nevertheless practicable for small projects.⁵⁴ The latter approach reflects the emerging strategy of database vendors to couple their DBMSs with proprietary generic ETL solutions (but which are limited in view of the complexity of the transformations they can model) [Agosta, 2002].

From the academic perspective, a sequence of papers [Vassiliadis et al., 2001, 2002, 2003, 2005] describes the system "Arktos"/"Arktos II", which is a tool that models and realizes the ETL process and ETL transformations. An ETL scenario can be modeled graphically by using a graphical specification language based on a formal conceptual and logical model describing data sources, data targets, and the transformations between the two. Although both "Arktos" projects cover the complete ETL process, their meta-models do not conform to emerging standards such as the CWM, thus, similar to the commercial tools mentioned, making it difficult to integrate the tool into existing heterogeneous data warehousing environments. The strength of "Arktos" lies in its template collection, which enables embedding of predefined ETL transformations into a new ETL scenario, enhancing reusability. The Arktos projects do not aim at WUA but at general data warehousing ETL requirements.

⁵⁴A comprehensive list of current ETL solutions can be found in Kimball and Caserta [2004, chapter 7].

Carreira and Galhardas [2004] present their data migration tool “Data Fusion”, which is designed to model complex transformations for general data migration tasks rather than ETL. The tool is concerned with the specification of migration transformations and supports related project development and management. The authors state that most commercial ETL tools are not expressive enough to model complex transformations, which are consequently frequently modeled outside the tools.⁵⁵ Conforming to the situation of WUA, “Data Fusion” targets projects dealing with large amounts of data that potentially involve a considerable number of transformations. In order to handle the complexity of transformations, the system provides a data transformation language for developing complex transformations compactly and concisely. The authors introduce the concept of a *mapper* that may enclose several *rules*, each of which encloses transformations with similar logics, providing a means of structuring complex transformations. Although “Data Fusion” does not explicitly target WUA, it seems to be appropriate to potentially meet the specific challenges for data transformations for WUA mentioned in the introduction of section 3.3 on page 41.

Appraising the ETL process as a key component in a data warehousing environment, Trujillo and Luján-Mora [2003] propose a UML-based approach for modeling the ETL process so as to ease its correct design by providing a reduced, yet powerful, set of ETL mechanisms: for example, aggregations, conversions, and filters. This set of mechanisms is in line with the ETL tasks discussed in Kimball and Caserta [2004]. Modeling the ETL processes by using the UML class diagram allows for conceptual aspects and avoids disclosing the actual implementation. The latter aspect is a drawback, inasmuch as implementing each conceptual step is not a straightforward matter, since the devil is in the details. The approach lacks a mapping from UML to concrete implementations that would significantly simplify ETL modeling.

Hu and Cercone [2004] present an abstract framework for WUA, the basic structure of which is similar to figure 15 on page 53, that is, the system is based on a clickstream data warehouse fueled with data from application servers. They describe in detail the principal requirements that each component must fulfill, but they do not go into concrete implementation aspects to prove that their system is feasible. Although the authors explicitly discuss the design of ROLAP star schemas for WUA, they do not address how to model and implement the ETL process in order to populate the schemas proposed.

Finally, Kimball and Caserta [2004] investigate the ETL process from two perspectives: (i) the *project management perspective* and (ii) the *implementation perspective*. While the latter perspective also draws the conclusion that ETL is not yet an open, standardized technology, the authors provide various hints and tips from a practitioner’s perspective concerning the concrete implementation of the ETL process. Many of these considerations are deployed in WUSAN’s ETL component in chapter 4 on page 59.

3.4 Summary

Tying in with chapter 2, in section 3.1 on page 25, this chapter introduced WUA as a primary instrument to deploy ECRM in the Web channel, currently the only true EC channel. In section 3.2 on page 34, the kind of data required for WUA were investigated: the conclusion was drawn that data collection on the application server level best serves the specific requirements for WUA. Then, in section 3.3 on page 41, prerequisites for a system for effective WUA were inferred, and the overall architecture of WUSAN was introduced. Furthermore, it was pointed

⁵⁵According to Kimball and Caserta [2004, chapter 1], almost all ETL tools allow *escapes* to standard programming languages to model complex transformations that cannot be modeled with the tool’s standard approach.

out that most components of WUSAN can be modeled with existing packages conforming to various data mining standards. Modeling the ETL component has been identified as a central open issue to complete the system.

The CWM was advocated as the perfect launching pad for ETL modeling. This view is also emphasized by Agosta [2002], who states that the CWM is arguably approved and complete enough to allow ETL vendors to implement it.⁵⁶ Due to the fact that both commercial tools and academic approaches currently do not provide a means of structured modeling, and, at the same time, implementing ETL, it is necessary to do more research on this issue being a crucial prerequisite for automating the WUA process and closing the loop in real projects.

The following chapter provides the theoretical background of the CWM data mining meta-model and introduces formal notations to infer the *logical object-oriented relational data storage model* (LOORDSM), which not only models the physical data storage of WUSAN's data warehouse but also provides a framework for the implementation of associated ETL transformations, thereby meeting the requirements mentioned in item 1 on page 42 (in addition, conforming to the CWM). The LOORDSM, which accounts for the specific situation of WUA, is then modeled as a UML class diagram and deployed in Java.⁵⁷

⁵⁶Agosta [2002] adds that they are currently refraining from implementing it, as they fear that this step will undercut their proprietary solutions and planned technology lock-in.

⁵⁷Kimball and Caserta [2004, chapter 1] mention that hand-coded ETL systems greatly profit from *unit testing* [compare Olan, 2003], which is not available in commercial ETL systems. Unit testing has also been employed for the Java implementations in the context of this thesis to keep all implemented classes accurate and maintainable.

Chapter IV

MODELING ETL FOR WEB USAGE ANALYSIS

This chapter explores how to model and automate the ETL process during the preprocessing phase of the WUA process. As discussed in the previous chapter, modeling data transformations is the key issue for the preprocessing phase, and it was concluded that only a proprietary architecture for WUA can fulfill these requirements. The missing building block for WUSAN in figure 15 on page 53 is the ETL component that handles WUA-specific ETL transformations.

In order to realize the missing building block, a model that systemizes and simplifies the ETL process, a sub-process of the WUA process, is sought. A clearly structured, formalized model provides for a precise, systematic user interface when the model is deployed in practise. Furthermore, a structured interface can be realized with XML amounting to a standardized, user-friendly interface.¹

A central asset of the LOORDSM, which is introduced in this chapter to bridge the gap caused by the missing ETL building block, is its coherence with the CWM². The LOORDSM inherits the structured transformation modeling of the CWM transformation package³ and compatibility to generic data pools conforming to the meta-model of the CWM data mining package^{4,5}

WUSAN implements the LOORDSM, which facilitates modeling and automating ETL transformations. Although the LOORDSM is not restricted to WUA, it is of special importance for this domain, as mentioned in the introduction of section 3.3 on page 41.

During the course of this chapter, many formalized notations are introduced in order to avoid tedious verbal circumscriptions and to provide a concise discussion of the LOORDSM. However, parts of the discussion are divided into a non-formal and a formal part to facilitate the comprehensibility of the reasoning for readers that are not familiar with formal notations.

4.1 Subsumption of the LOORDSM

Figure 16 on the next page depicts how the LOORDSM can be subsumed within WUSAN. This illustration represents an alternative view of WUSAN's data management capabilities, which comprise four layers: (1) the *Data Storage Layer*, (2) the *Data Mining Layer*, (3) the *ETL Layer*, and (4) the *OLAP Layer*, each of which is briefly discussed next.

(1) **Data Storage Layer.** This layer hosts the physical data storage in an RDBMS, which is employed to access the data in a structured and standardized manner. It was stated in

¹An XML interface allows for both automated and manual processing and can be expanded with graphical capabilities that further improve user-friendliness.

²Recall section 3.3.2.1 on page 46.

³Compare item 1 on page 49.

⁴Compare item 3 on page 49.

⁵Recall section 3.3.2.1 on page 46. Compatibility to the meta-model of the CWM data mining package amounts to compatibility with other CWM packages in figure 13 on page 48, as the meta-data from different packages can be transferred into each other through the CWM [Poole et al., 2002, chapter 2, especially figure 2.9]. Hence, it is sufficient to work with only the meta-model of the data mining package and employ this model as a standardized meta-model throughout this thesis.

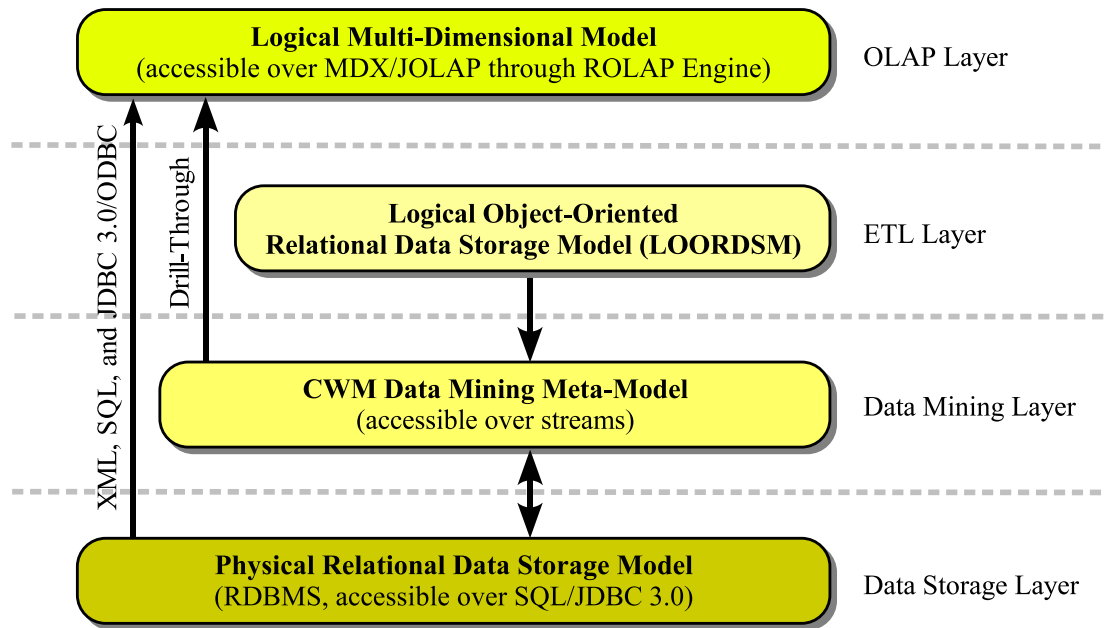


Figure 16: WUSAN's four-layer architecture.

section 3.3.3.3 on page 54 that commercial RDBMSs do not yet support the CWM relational meta-model. Consequently, connecting the proprietary meta-model of an RDBMS to the CWM data mining meta-model can be achieved only through the database streams of the data mining layer, the meta-data of which are compatible with the CWM data mining meta-model (compare section 3.3.3.3 on page 54).⁶

- (2) **Data Mining Layer.** As stated in the introduction of section 3.3.3 on page 51, the `MiningDataSpecification` class from the CWM data mining meta-model is employed as a standard meta-model in WUSAN. It is a fundamental prerequisite for WUSAN that needs to be thoroughly introduced and formalized. This is accomplished in section 4.2 on the next page, which also introduces transformation modeling with XELOPES (based on the CWM transformation meta-model). All data mining activities of WUSAN's analysis component in figure 15 on page 53 take place in the data mining layer.
- (3) **ETL Layer.** This layer hosts the core ETL process. Its main purpose is to populate the *physical* database tables employed within the data mining and OLAP layers. WUSAN's ETL component is deployed with the LOORDSM, which utilizes the CWM transformation meta-model as its core component. Prior to a detailed discussion of the LOORDSM in section 4.3 on page 82, transformation modeling is examined in section 4.2 on the facing page.
- (4) **OLAP Layer.** This layer accesses the data of the data storage layer, which is populated through the ETL layer, with a logical multi-dimensional model provided by the OLAP engine (previously discussed in section 3.3.3.4 on page 54). The mapping between the OLAP engine's meta-model and the meta-model of the RDBMS is accomplished over SQL and configured in a *schema file* through XML.⁷

⁶The actual mapping between the proprietary meta-data model of the RDBMS and the CWM data mining meta-model is accomplished with JDBC 3.0.

⁷Compare section G.4 on page 172.

4.2 Modeling Complex Transformations for Preprocessing

Preprocessing is essentially the execution of a sequence of data transformations. Thus, automating the preprocessing phase amounts to automating data transformations. The purpose of this section is to present the fundamentals for the LOORDSM in section 4.3 on page 82 and to devise an XML interface for convenient transformation modeling. These goals are tackled in three successive steps:

- (1) The CWM's meta-model of the data mining package strictly separates data and meta-data. Strictly speaking, data mining algorithms coherent with this meta-model operate on real-valued data only. The meaning of the data is obtained through its meta-data. Whenever users feed WUSAN with data, they must provide assigned meta-data, which characterize the data provided. Although this can be done through a PMML interface (or a user-friendly GUI that assembles required XML models), it is compulsory to understand meta-data modeling to entirely comprehend the LOORDSM. Section 4.2.1 for the first time presents a concise mathematical model capturing the complexity of CWM's data mining meta-model – the first step towards the section's goals.
- (2) The second step towards the section's goals is modeling data matrices or so called streams that must be configured by users whenever they access data and their meta-data reading or writing. In WUSAN, streams are basically inherited from XELOPES. However, the LOORDSM makes use of a variety of novel stream classes. Although users employing the LOORDSM do not have to understand all subtleties of these streams, a general understanding of them is indispensable to fully comprehend the LOORDSM. Section 4.2.2 on page 68 covers data matrices and streams from a bird's eye view whereas the details of the implemented stream classes are discussed in the appendix chapter D on page 131.
- (3) The third and final step towards the section's goals discusses in section 4.2.3 on page 72 transformation modeling based on XELOPES's infrastructure for transformation modeling conforming to the CWM transformation package. It is this step that is crucial for understanding the WusanML XML interface for transformation modeling. Whenever users model a concrete ETL scenario in practical projects, they must fall back on the WusanML interface and entirely comprehend its rationale.⁸

4.2.1 Modeling Meta-Data

4.2.1.1 Outline

Three basic terms, which are related in a bottom-up manner, are defined in the following section: (i) A *mining attribute*, the smallest meta-data unit, is defined. A mining attribute (referred to as an *attribute*) is defined as a real-valued variable to which a real-valued mapping into a user-defined domain is assigned. The mapping is referred to as the attribute's meta-data and determines the attribute's type. Four attribute types complying to the types usually employed for data mining can be defined routinely: *numeric* attributes, *discrete* attributes, *categorical* attributes, and *ordinal* attributes. The meta-data can be conceived as an "interpretation" of

⁸Chapter 5 on page 95 and the related appendix chapter G on page 149 exemplify how to leverage the WusanML XML interface in practical projects. The remark on page 105 points out that a GUI can be employed to support users in assembling WusanML models and observing its syntax and rationale. As such a GUI has not been realized yet in the current WUSAN prototype, users have no choice but to manipulate a WusanML file directly, which demands a thorough understanding of its syntax and semantics.

the attribute's real value. (ii) A (*mining*) *vector* is a collection of attributes to which, again, meta-data are assigned. The vector's mapping can be obtained in a straightforward manner by merging the meta-data of each attribute. A vector is the smallest unit on which data mining algorithms operate. (iii) A collection of vectors with the same meta-data is referred to as a *data matrix*. Each line of a data matrix corresponds to a vector, that is, a real-valued vector the interpretation of which can be obtained by applying the assigned meta-data to it.

Once the above mentioned terms are introduced, it must be confirmed that the definitions comply to the CWM's data mining package meta-model depicted in figure 17 on page 65, namely the `MiningDataSpecification` class. The formal definitions in the subsequent section integrate handling of missing values. For the sake of clarity, the formal discussion in section 4.2.1.2 is concluded with a detailed example in section 4.2.1.3 on page 66.

4.2.1.2 Mathematical Model

Definition 4.1 (Attribute). An *attribute* A is a variable that takes values in its domain $\mathbb{R} \cup \{\vartheta\}$ (*missing values* are represented by ϑ).

Definition 4.2 (Meta-Data of an Attribute). Let A be an attribute, let $M(A)$ be a user-defined domain, and let $\Theta(A)$ be a set of values that represent missing values. Any *onto mapping*

$$M_A : \mathbb{R} \cup \{\vartheta\} \rightarrow M(A) \cup \Theta(A) \quad (4.1)$$

that fulfills the following conditions

- (i) $M_A|_{\mathbb{R} \setminus M_A^{-1}(\Theta(A))}$ is a one-to-one mapping⁹,
- (ii) $M_A(\vartheta) \in \Theta(A)$,
- (iii) if $M(A) \not\subseteq \mathbb{R}$, there exists $n \in \mathbb{N}_0$ such that

$$M_A(\mathbb{R} \setminus \{0, \dots, n\}) \subseteq \Theta(A)$$

and

$$M_A(\{0, \dots, n\}) = M(A)$$

hold with $M(A)$ countable, and

- (iv) if $M(A) \subseteq \mathbb{R}$, $M_A = I_S$ holds where $S \subseteq \mathbb{R}$ with $I_S(a) = a$ if $a \in S$ and $I_S(\mathbb{R} \setminus S) = \Theta(A) := \{\vartheta\}$

is called *meta-data* appendant to A .

Definition 4.2 requires elucidation by a number of comments. First of all, it must be emphasized that the term meta-data actually refers to a *surjective mapping*, that is, whatever real value is provided, the mapping is defined and results in a valid value. However, equation (4.1) must be constrained with four conditions to be meaningful:

- ad (i): The mapping must be unique except for the set of values that are mapped to the set $\Theta(A)$ of missing values.

⁹ $M_A^{-1}(\Theta(A)) := \{a \in \mathbb{R} \cup \{\vartheta\} : M_A(a) \in \Theta(A)\}$ denotes the *preimage* of $\Theta(A)$ under M_A .

- ad (ii): The constant ϑ , representing a missing value in the preimage space, must be mapped to the set $\Theta(A)$ of missing values. However, yet other real values may be mapped to the set of missing values.
- ad (iii): This constraint anticipates definition 4.3. Categorical attributes may only take the values $0, \dots, n$ in the image space, that is, 0 represents the first discrete value, 1 the second and so forth. Of course, any other discrete mapping would be feasible for categorical attributes, but would rather lead to confusion and is obviated by this constraint. In short, this constraint *standardizes* discrete attributes.
- ad (iv): This constraint also anticipates definition 4.3. For numeric attributes, equation (4.1) on the preceding page must be defined as an identical mapping except for those values that are mapped to the set of missing values $\Theta(A)$. This means that no pathological real-valued mappings are permitted. This constraint is rather trivial, since mappings other than the identical mapping are not meaningful for numeric attributes. If the range of a numeric attribute is to be restricted, its meta-data can be modified by adjusting the set S , that is, undesired values are mapped to $\Theta(A)$.

Next, the attribute types mentioned just now and outlined in the previous section are defined.

Definition 4.3 (Mining Attribute). Let A be an attribute and let M_A be meta-data appendant to A . Then, the tuple (A, M_A) denotes a *mining attribute* that can be further classified according to its meta-data:

- (i) If $M(A) \subseteq \mathbb{R}$, (A, M_A) is denoted as a *numeric attribute*.
- (ii) If $M(A)$ is countable, (A, M_A) is characterized as a *discrete attribute*.
- (iii) If $M(A) \not\subseteq \mathbb{R}$, (A, M_A) is referred to as a *categorical attribute*.
- (iv) Let (A, M_A) be a categorical attribute. If there exists a total order¹⁰ on $M(A)$ such that M_A preserves the natural order on the preimage space, (A, M_A) is called an *ordinal attribute*.

Definition 4.3 is straightforward. Given that an attribute's meta-data M_A comply to definition 4.2 on the preceding page, its user-defined domain $M(A)$ is real-valued for a numeric attribute, contains a finite number of values for a discrete attribute, or may not be real-valued for a categorical attribute. The latter condition implies that categorical attributes with numeric categories should be modeled as discrete numeric attributes.¹¹ Finally, for ordinal attributes, there must exist a meaningful order on M_A and the meta-data must preserve the natural order on $0, \dots, n$.

Remark. (a) In order to simplify the notation, a mining attribute (A, M_A) in definition 4.3 is simply referred to as an attribute A that implicitly provides meta-data.

(b) It is important to note that in definition 4.2 on the preceding page, $M(A) \cap \Theta(A) \neq \emptyset$ may hold for a categorical attribute. This means that by adding a category of $M(A)$ to $\Theta(A)$, occurrences of this category must be interpreted as *missing values*.

¹⁰A *binary order relation* R on a set fulfills *reflexivity*, *anti-symmetry*, and *transitivity*. R is a total order relation, if two arbitrary elements of the set are comparable.

¹¹The Quibbler's remark: numbers in terms of *strings* may of course be modeled as categories for categorical attributes.

The following definition describes what has been referred to as “merging” the meta-data of a collection of attributes in the outline in section 4.2.1 on page 61.

Definition 4.4 (Meta-Data of a Set of Attributes). Given attributes A_1, \dots, A_m , $m \in \mathbb{N}$. A mapping

$$M_{A_1, \dots, A_m} : (\mathbb{R} \cup \{\vartheta\})^m \rightarrow M(A_1) \cup \Theta(A_1) \times \dots \times M(A_m) \cup \Theta(A_m)$$

that fulfills $M_{A_1, \dots, A_m}(\mathbf{a}) := (M_{A_1}(a_1), \dots, M_{A_m}(a_m))$ with $\mathbf{a} = (a_1, \dots, a_m) \in (\mathbb{R} \cup \{\vartheta\})^m$ is called *meta-data appendant* to A_1, \dots, A_m .

Definition 4.4 states that the meta-data of a collection of attributes can be retrieved by simply combining the meta-data of each attribute. Definition 4.5 assembles a collection of attributes to a mining vector in a bottom-up approach.

Definition 4.5 (Mining Vector). Let A_1, \dots, A_m , $m \in \mathbb{N}$ be attributes, let $\mathbf{a} = (a_1, \dots, a_m) \in (\mathbb{R} \cup \{\vartheta\})^m$, and let M_{A_1, \dots, A_m} be meta-data appendant to A_1, \dots, A_m . Then, the tuple

$$(\mathbf{a}, M_{A_1, \dots, A_m})$$

denotes a *mining vector*.

Remark. To simplify the notation, a mining vector $(\mathbf{a}, M_{A_1, \dots, A_m})$ in definition 4.5 is referred to as a *vector* \mathbf{a} that implicitly provides meta-data.

Finally, definition 4.6 assembles a collection of vectors with the *same* meta-data to a data matrix continuing the bottom-up approach.

Definition 4.6 (Data Matrix). Let $\mathbf{a}_1, \dots, \mathbf{a}_n$, $n \in \mathbb{N}$ be vectors with meta-data M_{A_1, \dots, A_m} , $m \in \mathbb{N}$. Then, the matrix

$$\mathbf{A} := \begin{pmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_n \end{pmatrix} = \begin{pmatrix} a_1^1 & \cdots & a_1^\ell & \cdots & a_1^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_i^1 & \cdots & a_i^\ell & \cdots & a_i^m \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_n^1 & \cdots & a_n^\ell & \cdots & a_n^m \end{pmatrix}$$

is called a *data matrix*.

Remark. A data matrix can be accessed vector by vector following the matrix notations of R-PROJECT and Verzani [2005], that is, $\mathbf{A}[i, \cdot] := \mathbf{a}_i$, $i = 1, \dots, n$. Furthermore, a vector is a $1 \times m$ data matrix.

Definition 4.7 (Meta-Data of a Data Matrix). Let \mathbf{A} be a data matrix. $M_{\mathbf{A}} := M_{A_1, \dots, A_m}$ is called *meta-data appendant* to \mathbf{A} .

Remark. A data matrix \mathbf{A} implicitly provides meta-data. Note that, in practice, in the majority of the cases $M_{\mathbf{A}}(\mathbf{A})$ is available only, and \mathbf{A} and $M_{\mathbf{A}}$ must be derived (see the example in section 4.2.1.3 on page 66).

Basically, a data matrix contains real-valued row vectors that are “interpreted” by its assigned meta-data. The data mining algorithms provided by XELOPES operate on a data matrix. However, as section 4.2.2 on page 68 reveals, a data matrix is not the same as a *data source*, for instance, a flat file or a database table.

In practice, data matrices can be derived from a variety of data sources, for example, flat files that contain Web logs, database tables in an RDBMS, and multi-dimensional arrays in higher programming languages. As mentioned in section 3.3.2.1 on page 46, the CWM’s resource layer defines meta-models for all of the aforementioned types of data sources. According to the introduction of section 3.3.3 on page 51, due to the lack of software products or programming libraries that cover the CWM entirely or at least significant parts of it when used in a data warehousing environment, the `MiningDataSpecification` class from the CWM data mining meta-model makes up for this weakness and acts as a universal meta-model for modeling data matrices within XELOPES (and hence within WUSAN as well).

In order to show that definition 4.4 on the preceding page is compatible with the `MiningDataSpecification` class, it is necessary to discuss some of the details of the corresponding CWM meta-model, which is depicted in figure 17 as a UML class diagram. An instance of the `MiningDataSpecification` class consists of one or more instances of the `MiningAttribute` class.¹² A `MiningAttribute` accepts one of the following types: (a) `NumericAttribute`, (b) `CategoricalAttribute`, and (c) `OrdinalAttribute`, each of which corresponds to an attribute type in definition 4.3 on page 63. Clearly, a `CategoricalAttribute` and item iii on page 63 are equivalent due to the fact that a `CategoricalAttribute` stores its associated categories in an ordered list that maps each category uniquely to a category number. A `CategoricalAttribute` provides access to this mapping in both directions. All other equivalences between the attribute types of definition 4.3 on page 63 and figure 17 are apparent.

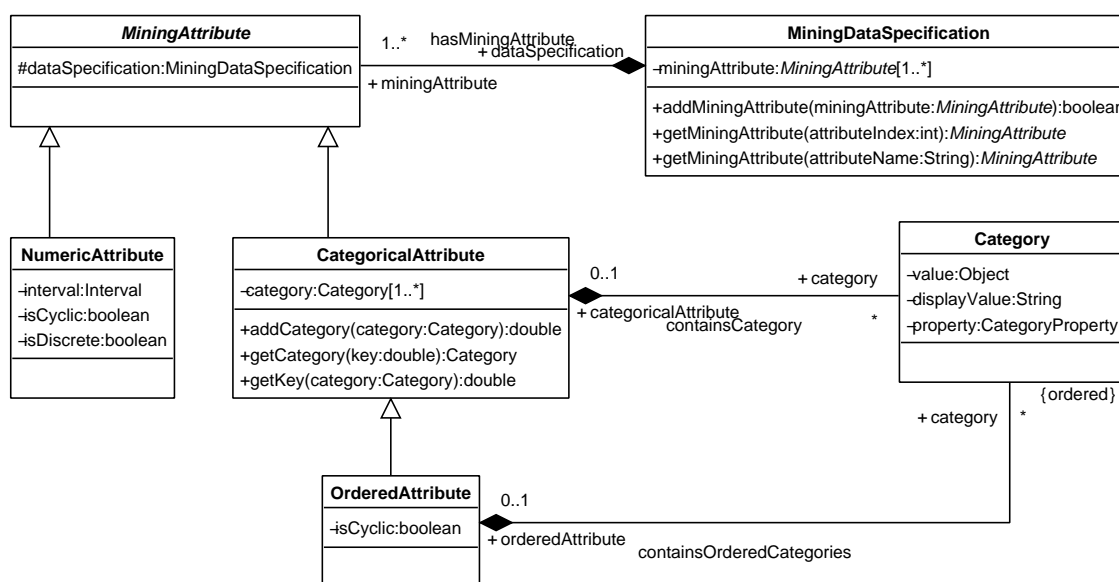


Figure 17: Meta-data modeling with the CWM data mining package (UML class diagram).

¹²Strictly speaking, there is a significant difference between a *class* and *instances* of a class. UML describes the classes and interrelations between the classes. Such interrelations actually exist only among instances of the classes. However, discussing UML class diagrams is unnecessarily protracted, if this distinction is explicitly made for every occurrence of a class name. Therefore, from now on, both classes and instances are uniformly typeset in the typewriter style.

4.2.1.3 Example

This example examines three aspects of a data matrix and its assigned meta-data.

Interpreting a Data Matrix Employing its Assigned Meta-Data Given is the following real-valued 4×2 data matrix

$$\mathbf{A} = \begin{pmatrix} 0.5 & 1 \\ 1 & 0 \\ 2 & 0.5 \\ 0.1 & \vartheta \end{pmatrix}. \quad (4.2)$$

A_1 is a numeric attribute implying $\Theta(A_1) = \{\vartheta\}$, and A_2 is a categorical attribute with $\Theta(A_2) = \{\text{"null"}\}$. Let the meta-data of A_1 correspond to the mapping

$$M_{A_1}(a) = I_{[0,1]}(a) = \begin{cases} a, & \text{if } a \in [0, 1] \\ \vartheta, & \text{if } a \in (\mathbb{R} \setminus [0, 1]) \cup \{\vartheta\}, \end{cases} \quad (4.3)$$

that is, A_1 is restricted to the interval $[0, 1]$ and any values outside this interval are interpreted as missing values by A_1 's meta-data. Let the meta-data of A_2 correspond to

$$M_{A_2}(a) = \begin{cases} \text{"category 1"}, & \text{if } a = 0 \\ \text{"category 2"}, & \text{if } a = 1 \\ \text{"category 3"}, & \text{if } a = 2 \\ \text{"null"}, & \text{if } a \in (\mathbb{R} \setminus \{0, 1, 2\}) \cup \{\vartheta\}, \end{cases} \quad (4.4)$$

that is, A_2 is defined as a categorical attribute with three categories. It can be easily verified that the meta-data in equation (4.3) and equation (4.4) conform to definition 4.2 on page 62 and definition 4.3 on page 63.

Let $\mathbf{a} \in (\mathbb{R} \cup \{\vartheta\})^2$ be a vector according to definition 4.5 on page 64. Then, $M_{\mathbf{A}}(\mathbf{a}) = (M_{A_1}(a_1), M_{A_2}(a_2))$ holds, that is, its meta-data is obtained by combining the meta-data of each attribute. Applying $M_{\mathbf{A}}$ to the entire data matrix in equation (4.2) yields the following *transformed data matrix*, an interpretation of \mathbf{A} :

$$M_{\mathbf{A}}(\mathbf{A}) = \begin{pmatrix} 0.5 & \text{"category 2"} \\ 1 & \text{"category 1"} \\ \vartheta & \text{"null"} \\ 0.1 & \text{"null"} \end{pmatrix}. \quad (4.5)$$

Equation (4.5) is called a *stream*. It is obvious that this notion corresponds to the streams mentioned in section 3.3.3.1 on page 52. The notion of a stream is covered in more detail in section 4.2.2 on page 68.

Deriving Meta-Data from a Stream In practice, the reverse approach is taken, that is, a stream is given and the data matrix is derived implicitly by canonically creating $M_{\mathbf{A}}$.¹³ Alternatively, $M_{\mathbf{A}}$ can be explicitly specified and provided to the stream. It is important to note that the mining algorithms operate on the data matrices that are derived from the streams, not on the streams directly. This means, for example, that if $\mathbf{A}[1,]$ is required by a mining algorithm,

¹³ $M_{\mathbf{A}}$ is generally not unique.

it is computed *dynamically* as $\mathbf{A}[1,] = (0.5, 1)$. Therefore, it is necessary to infer $a \in \mathbb{R}$ for given $M_A(a)$. This can be achieved with the following mapping:

$$\overline{M}_A(m) := \begin{cases} M_A^{-1}(m), & \text{if } m \in M(A) \setminus \Theta(A) \\ \vartheta, & \text{otherwise.} \end{cases} \quad (4.6)$$

$M_A^{-1}(m)$ is the *inverse mapping* of M_A , which obviously exists due to definition 4.2 on page 62. Although equation (4.6) is not the inverse mapping of M_A in the strict sense, it is referred to as the *inverse meta-data* of A .

Remark. Adding “category 2” to $\Theta(A)$ in above example yields $\mathbf{A}[1,] = (0.5, \vartheta)$ due to equation (4.6). This reflects remark item b on page 63.

Presetting Meta-Data for a Stream M_A can be regarded as an “interpretation” of its assigned attribute A . Changing the meta-data mapping M_A leads to a different interpretation of A . If the meta-data mapping of A_1 equation (4.3) on the facing page is altered by using $I_{[0,\infty)}$ instead of $I_{[0,1]}$ and if the meta-data mapping of A_2 equation (4.4) on the preceding page is replaced by

$$M_{A_2}(a_2) = \begin{cases} \text{“red”}, & \text{if } a_2 = 0 \\ \text{“null”}, & \text{if } a_2 \in (\mathbb{R} \setminus \{0\}) \cup \{\vartheta\}, \end{cases} \quad (4.7)$$

the following stream, given the same initial data matrix equation (4.2) on the facing page, is obtained:

$$M_{\mathbf{A}}(\mathbf{A}) = \begin{pmatrix} 0.5 & \text{“null”} \\ 1 & \text{“red”} \\ 2 & \text{“null”} \\ 0.1 & \text{“null”} \end{pmatrix}. \quad (4.8)$$

Conversely, if the stream in equation (4.8) is given, the meta-data for A_2 can be canonically inferred as equation (4.7). Another option is presetting the meta-data for a given stream. Suppose that the following stream

$$M_{\mathbf{A}}(\mathbf{A}) = \begin{pmatrix} 0.5 & \text{“blue”} \\ 1 & \text{“red”} \\ 2 & \text{“null”} \\ 0.1 & \text{“null”} \end{pmatrix}$$

is given and its meta-data for A_2 are preset as equation (4.7). If a data mining algorithm seeks $\mathbf{A}[1,]$, \overline{M}_{A_2} can be leveraged to infer that $\mathbf{A}[1,] = (0.5, \vartheta)$ holds, since “blue” is not a valid category.¹⁴

However, if a stream is configured to alter its meta-data dynamically, M_{A_2} can be modified by adding “blue” to $M(A_2)$ as a valid category, yielding $\mathbf{A}[1,] = (0.5, 1)$, since category “blue” is assigned to the next available integer, that is, 1. It is important to bear in mind that there is, in fact, a significant difference between a data matrix \mathbf{A} and a stream $M_{\mathbf{A}}(\mathbf{A})$. As streams are a fundamental prerequisite for WUSAN, they are discussed in greater detail in section 4.2.2 on the following page.

¹⁴Strictly speaking, it would be better to introduce *invalid values* besides missing values, since “blue” in the above stream is actually not a missing value. However, this would unnecessarily increase the complexity of the mathematical model. In practice, the CWM data mining meta-model supports invalid values along with missing values. The concept of invalid values can be added to the mathematical model similarly to the concept of missing values.

4.2.1.4 Summary

This section briefly summarizes the benefits of the meta-data concept that has been introduced as a mathematical model in the previous sections.

- (1) The model provides a standardized and clearly defined meta-data model, which is compatible with the CWM data mining package meta-model and hence, with any generic data pools conforming to the CWM. This feature supersedes any meta-data transformations during the WUA process.¹⁵ The model is applicable as shown at the end of section 4.2.1.2 on page 62, since it has been realized in XELOPES, which deploys figure 17 on page 65.
- (2) In the example in the previous section, it has been demonstrated that meta-data can be derived from any stream through inverse meta-data. That is, for any data source provided as a stream, meta-data are readily available. Alternatively, they can be explicitly preset for streams. In appendix chapter G on page 149, it is demonstrated how meta-data can be modeled with PMML, which represents a standardized XML interface for users.
- (3) By altering meta-data that are assigned to a data matrix, it is possible to create a different interpretation of the same data set. This can be beneficial during the pattern discovery phase when a highly flexible data pool is required. Furthermore, some of the transformations discussed in section 4.2.3 on page 72 amount to altering meta-data – a very efficient way of implementing a stream transformation.
- (4) The handling of missing values can be dealt with by administering the missing value set $\Theta(A)$ for each attribute A . This feature is discussed in the following section.
- (5) Storing a data matrix and its assigned meta-data in a data source instead of the inferred stream, generally helps to reduce the amount of data to be stored.
- (6) The transformations that are introduced in section 4.2.3 on page 72 can be modeled in a clearly structured manner. Furthermore elemental transformations that are coded in Java can be implemented and tested more efficiently, since they also separate the transformation of data and their meta-data (also compare next item).
- (7) The mathematical model supports a unified view of data mining (see appendix chapter C on page 129). Admittedly, this is rather significant from a theoretical perspective.

4.2.2 Modeling Streams

Streams have been mentioned twice so far: first, in section 3.3.3.1 on page 52, it was stated that streams are employed to access data within WUSAN; second, in section 4.2.1.3 on page 66, it was mentioned that, in practice, a stream $M_{\mathbf{A}}(\mathbf{A})$ is given, not the data matrix \mathbf{A} itself.

This section discusses the stream classes of XELOPES and WUSAN, which provide access to manifold kinds of data sources. The classes relevant to the LOORDSM are illustrated as UML class diagrams throughout this section.¹⁶

¹⁵Compare section 3.3.2.1 on page 46.

¹⁶In order to simplify the UML class diagram illustrations, the following approach is pursued: a UML class diagram only depicts methods and variables that are introduced or overridden by this class. Methods implementing an interface are not depicted, since it is clear that an implementing class must realize them. Some of the original XELOPES stream classes have been adapted for WUSAN, whereas others have been implemented from scratch. Adaptations have been made in view of consistent behavior, performance, and additional functionality.

4.2.2.1 Stream Properties

As yet, streams have been discussed from a theoretical perspective. It has been determined that data matrices and streams can be mapped into each other through the assigned meta-data or inverse meta-data, respectively. Furthermore, it was stated that streams are employed to model various data sources in XELOPES and that they are passed on to WUSAN. Hence, users must be knowledgeable about the basic features of stream classes, that is, Java classes that model streams.¹⁷ Thus, it is necessary to switch to a nuts and bolts perspective to discuss streams. A programmer's nuts and bolts perspective is UML, and its sub-model of class diagrams is well suited to discuss streams and their various features.¹⁸

The abstract class `MiningInputStream` is the prototype of a stream. Its UML class diagram is depicted in the appendix section D.1 on page 131 and can be skipped at this point. The class offers methods for three functional areas: (1) *general stream properties*, especially a stream's meta-data, (2) *cursor methods* that implement the access to the rows of its assigned data matrix, and (3) *handling of missing values*, that is, methods that administer the sets of missing values.¹⁹

The following definition links the notion of a stream to a concrete Java class implementing the principal features just mentioned.

Definition 4.8 (Stream). The image of a data matrix $M_A(\mathbf{A})$ is called a *stream*. It is modeled by the abstract class `MiningInputStream` in figure 43 on page 131, which provides access to the inferred data matrix \mathbf{A} .

In detail, a stream comprises the following features:

- (1) **General Stream Properties.** Once a stream's constructor has been invoked, the stream must provide meta-data that are accessible with the `getMetaData` method. The meta-data may originate from three different sources: (i) they may be *explicitly* provided as an argument of the stream's constructor, (ii) they may be *implicitly* provided by an additional data source, for example, a flat file or a database table providing meta-data in a standardized format, for instance, the PMML `DataDictionary` tag²⁰, or (iii) they may be *auto-detected* by iterating through the complete stream or portions of it and canonically creating its meta-data.²¹
- (2) **Cursor Methods.** Prior to accessing it, a stream must be initialized with the `open` method. Nonrecurring stream initializations, that is, initializations that have to be made only once for a stream (usually time-expensive operations), are placed in the `open` method. Recurring stream initializations, that is, initializations that must be executed when the stream is reset (generally less time-expensive operations), are placed in the `reset` method. Both methods position the cursor in front of the first vector of the data matrix.²²

¹⁷Lacking ease of use in terms of GUI support for users is a definite deficiency of the current WUSAN prototype. However, this thesis rather focuses on research results than usability. That is why a low-level technical discussion could not be entirely omitted.

¹⁸The detailed UML diagrams can be found in the appendix chapter D on page 131.

¹⁹Moreover, the `MiningInputStream` class implements various getter methods that access its properties.

²⁰Compare section 3.3.2.2 on page 50.

²¹The first two items refer to presetting the meta-data for streams and the last item refers to deriving the meta-data through the stream's inverse meta-data. Both alternatives have been brought up in the example of section 4.2.1.3 on page 66.

²²The `reset` method can be invoked, only if the stream is opened, that is, if the nonrecurring stream initializations have been completed.

Once the stream is initialized, its cursor can be moved with the `next` and `move` methods. While the former moves the cursor from its position to the next row, the latter positions the cursor directly at a certain position within the stream.²³ The `read` method retrieves $\mathbf{A}[i,]$, if the cursor is placed at position i . It is this method that actually provides access to the inferred data matrix as mentioned in definition 4.8 on the previous page.

- (3) **Handling of Missing Values.** As mentioned in definition 4.1 on page 62, missing values are represented by ϑ in a data matrix \mathbf{A} , whereas missing values in a stream $M_{\mathbf{A}}(\mathbf{A})$ are represented by the elements of the set of missing values $\Theta(A)$ for attribute A . A `MiningInputStream` implements the `addMissingValue` method, which adds strings to $\Theta(\mathbf{A}) := \Theta(A_i)$ for all categorical attributes $A_i \in \{A_1, \dots, A_m\}$. This means that there exists only *one* set of missing values for a stream shared by all categorical attributes. The `isMissingValue` method checks whether a given string is contained in $\Theta(\mathbf{A})$. As $\Theta(A_i) := \{\vartheta\}$ for all numerical attributes $A_i \in \{A_1, \dots, A_m\}$ (compare item iv on page 62), the methods for handling missing values in a stream have an effect on categorical attributes only.

Depending on the data source accessed by a stream, it is possible to provide not only *reading* access but also *writing* access. Writing access is modeled by the `UpdatableStream` interface depicted in the UML class diagram in figure 44 on page 132. The interface offers methods for appending single vectors or complete streams on condition that the meta-data match.

4.2.2.2 Stream Classes

The `MiningInputStream` class has been extended by various sub-classes, providing for concrete stream implementations that access certain physical data sources or that fulfill certain data transformation tasks. The list below gives a brief overview of the streams available in XELOPES and WUSAN:

- (1) **Memory Streams.** Refers to the streams that reside in the computer's main memory, that is, they allow for fast stream operations and data access but are limited in the amount of data they can handle. The `MiningArrayStream` class is based on an array of fixed length that cannot be altered at runtime; the `MiningCollectionStream` class is updatable, since it is based on Java collections²⁴. The UML class diagrams of both streams can be found in the appendix section D.2 on page 132.
- (2) **Flat File Streams.** Refers to streams that access data in flat files. The prototype of a flat file stream is modeled by the `MiningFileStream` class. Its sub-classes parse the file structure either through a Java `StreamTokenizer` (realized by the abstract `MiningTokenizerStream` class) or through regular expressions [compare Friedl, 2002] (realized by the `MiningCsvStream` and the `LogFileStream` classes). The following types of file streams are available:
- (i) The `MiningArffStream` class models streams with an embedded flat file conforming to the ARFF file format [see Paynter et al., 2002]. This file format is proposed by Witten and Frank [2005, section 2.4].
 - (ii) The `MiningCsvStream` class models streams with an embedded CSV flat file, that is, its values are separated by a delimiter and optionally enclosed in quote chars.

²³Only if this feature is supported by the stream; this can be checked with the `isMovable` method.

²⁴An introduction to Java collections can be found in Horstmann and Cornell [2005b, chapter 2].

- (iii) The `MiningC50Stream` class models streams with an embedded flat file conforming to the C5.0 file format [compare `RULEQUEST`].²⁵ This file format is actually comprised of two files: (a) a *data* file in CSV format and (b) a *names* file that stores meta-data in a proprietary format. This stream transforms the meta-data originating from the names file into a `MiningDataSpecification`.
- (iv) The `LogFileStream` class models streams with an embedded flat file conforming to one of various log formats such as the W3C extended log format or the NCSA log formats [see Sweiger et al., 2002, chapter 2].

The corresponding UML class diagrams covering the flat file streams mentioned can be found in the appendix section D.3 on page 133.

- (3) **Database Streams.** Refers to streams that access data in an RDBMS. Databases play an important role when it comes to managing and storing large volumes of data. Figure 48 on page 135 depicts WUSAN's database streams. The `MiningSqlStream` class provides access to the *result set* of an SQL query. Its meta-data is auto-detected through the result set meta-data provided by the RDBMS – in combination with a (partial) iteration through the result set. This approach has one major drawback: meta-data operations in an RDBMS are expensive in terms of program execution performance, especially if they occur repeatedly, and so they should generally be avoided or at least alleviated by sending smart queries that minimize the costs of retrieving database meta-data to the RDBMS [`DATADIRECT`]. The following two sub-classes address this particular problem:

- (i) The `MiningTableSqlStream` class accesses data in a *single* table of an RDBMS. Such a table must conform to a special format that can be created with WUSAN's `MiningUpdatableSqlSource` class (see appendix section D.6 on page 133). This table format provides for meta-data that is compatible with the `MiningDataSpecification` class, stored in PMML format. In order to accelerate reading and writing stream accesses, this stream implements internal insert and select caches. It is a fundamental prerequisite for deploying the LOORDSM.
- (ii) The `MiningQuerySqlStream` class models access to a result set that is created by an SQL query, with the involved tables conforming to the internal structure required for the `MiningTableSqlStream` class.

- (4) **Filter Streams.** The abstract class `MiningFilterStream` models a stream that embeds another stream. The embedded stream is transformed by its wrapper class through various stream transformations. Common transformations are, for instance, a *row filter*, that is, certain rows of the embedded stream are filtered if they match a filter criterion, or a *column filter*, that is, certain columns from the individual vectors of the embedded stream are skipped. The sub-class `MiningVectorFilterStream` models a row filter based on regular expressions for each attribute.²⁶ The sub-class `MiningTransformationStream` applies a *regular transformation* to every vector of the embedded stream.²⁷ This feature ensures that this sub-class is a powerful instrument for realizing complex stream transformations. A column filter is only one of many applications of this stream.

²⁵This stream class has been developed for WUSAN due to the fact that the KDD Cup 2000 data employed in chapter 5 on page 95 are stored in C5.0 format.

²⁶An introduction to regular expressions is given in Friedl [2002].

²⁷Regular transformations are discussed in section 4.2.3 on the following page.

Finally, the `MultidimensionalStream` class models simple memory-based, OLAP-like select and order operations. Its practical employment is described in Thess and Bolotnicov [2004, section 6.4.7].

The corresponding UML class diagrams for filter streams are illustrated in the appendix section D.5 on page 133. Furthermore, vector filters, which are required for the `Mining-VectorFilterStream` class, are discussed in more detail in section D.7 on page 137.

4.2.2.3 Summary

The various types of streams discussed in the previous section serve three purposes within WUSAN: (i) They are employed within the data access component in figure 15 on page 53 in order to tap various physical data sources, that is, they model the source streams for the ETL process. (ii) Database streams are employed within the data warehousing component in figure 15 on page 53 as target streams.²⁸ (iii) They are employed for tapping the data warehouse for data mining within the analysis component in figure 15 on page 53.

Now, the preparatory work for transformation modeling is completed, that is, the notion of meta-data has been introduced and streams that leverage the meta-data are disposable. This means that the most complex milestone towards the LOORDSM can be tackled next: transformation modeling within the ETL component in figure 15 on page 53.

4.2.3 Transformation Modeling

4.2.3.1 Outline

As mentioned in the introduction of section 4.2 on page 61, preprocessing can be regarded as the application of a sequence of complex transformations to a stream, that is, preprocessing basically transforms a stream into another stream. As stated before, the third step towards an XML interface for transformation modeling is the design of a mathematical model for ETL transformations, which must be convertible into a structured XML user interface.

WUSAN's architecture in figure 15 on page 53 shows that the data flow during ETL transformations is split into two parts: first, the transformation of the data, that is, a sequence of *real-valued transformations*, and second, the transformation of the meta-data, that is, a sequence of *meta-data transformations*.

The first part of the mathematical model deals with the different types of transformations that may occur for ETL modeling. It is inferred that for ETL modeling, certain types of transformations are essential, referred to as *regular transformations*. Regular transformations can be executed vector by vector, that is, every single vector of the source stream is mapped to a distinct vector of the target stream. It is the regular transformations or rather the *vector transformation* related to them that are the backbone of ETL modeling.

Consequently, the question arises as to how vector transformations can be modeled in view of (i) conveniently coding vector transformations to compile a library of standardized transformations for the WUA domain, (ii) concatenating and nesting vector transformations to facilitate the modeling of complex vector transformations, and (iii) providing a convenient and user-friendly XML interface for transformation modeling.

²⁸Target streams are defined with the `WusanML TableStream` tag. An example for employing this tag can be found in listing G.15 on page 166.

4.2.3.2 Mathematical Model

Basics of Transformation Modeling It has been mentioned before that ETL is about transforming streams into one another. The following definition characterizes the most general type of an ETL transformation.

Definition 4.9 (Transformation). Let A_1, \dots, A_m be attributes and let M_{A_1, \dots, A_m} be meta-data appendant to A_1, \dots, A_m . Let $\mathbb{M}(A_1, \dots, A_m)$ be the set of data matrices with m columns that provide meta-data M_{A_1, \dots, A_m} . Then, a mapping

$$T_{A_1, \dots, A_m} : \mathbb{M}(A_1, \dots, A_m) \rightarrow \mathbb{M}(A'_1, \dots, A'_{m'}) \quad (4.9)$$

is called a *transformation*, if for any preimage data matrix $\mathbf{A} \in \mathbb{M}(A_1, \dots, A_m)$ its image $T_{A_1, \dots, A_m}(\mathbf{A})$ has equal lines as or fewer lines than its preimage \mathbf{A} .

Remark. Equation (4.9) can be shortened to

$$T_{A_1, \dots, A_m} : A_1, \dots, A_m \rightarrow A'_1, \dots, A'_{m'},$$

since it is implicitly clear on what kind of domains the transformations operate.

This definition basically states that transformations operate on data matrices and that the transformed data matrix cannot have more lines than the original data matrix, that is, a transformation can only map available vectors (each vector separately or all vectors as a whole), but not “create” additional vectors. Normally, the original and the transformed data matrix have the same number of lines. A good example of a transformation where the transformed data matrix has less lines than the original data matrix is a filter that separates out certain vectors of the original data matrix.

If in definition 4.9 both the original data matrix and the transformed data matrix consist of one vector only, T_{A_1, \dots, A_m} is referred to as a *vector transformation*. As mentioned before, a vector transformation can be split into the following two transformations:

- (i) a *real-valued transformation* $t_{A_1, \dots, A_m} : \mathbb{R}^m \rightarrow \mathbb{R}^{m'}$ and
- (ii) a *meta-data transformation* $T_{M_{A_1, \dots, A_m}} : M_{A_1, \dots, A_m} \rightarrow M_{A'_1, \dots, A'_{m'}}$.

This reflects the characteristics of vectors, defined in definition 4.5 on page 64. Vectors consist of a real-valued vector and meta-data to interpret this vector.

The following definition characterizes the most important transformation type in ETL modeling. A *regular transformation* links a transformation according to definition 4.9 and a vector transformation and allows to reduce the discussion of transformation modeling to vector transformations, significantly simplifying subsequent disquisitions.

Definition 4.10 (Regular Transformation). Let T be a transformation and let \mathbf{A} be an $n \times m$ data matrix. If there exists a vector transformation T' such that

$$T(\mathbf{A}) = \begin{pmatrix} T'(\mathbf{A}[1,]) \\ \vdots \\ T'(\mathbf{A}[n,]) \end{pmatrix},$$

T is called a *regular transformation*.

A regular transformation T can be regarded as a vector transformation that is applied to every single vector of a data matrix. Consequently, for regular transformations, it is sufficient to analyze their assigned vector transformations. For the sake of simplification, a vector transformation T' that is assigned to a regular transformation T is also referred to as T , though, strictly speaking, the two transformations are not the same. To complete the theory, a transformation T that is not regular is referred to as a *special transformation*.

Regular transformations are fully supported by the CWM transformation package. Most preprocessing transformations occurring in practice are modeled as regular transformations. In general parlance, if a preprocessing transformation does not require information that can be calculated or retrieved only from more than one line of the data matrix or from the whole data matrix (for example, an empirical expected value or an empirical variance), it can be modeled as a regular transformation. Otherwise, it must be modeled as a special transformation. Special transformations represent a proprietary transformation type that is not supported by the CWM transformation package [compare Thess and Bolotnicov, 2004, section 6.6.4].

XELOPES implements two interfaces that can be employed for representing the two transformation types. Regular transformations can be modeled with the `MiningTransformer` interface in figure 18, whereas special transformations can be modeled with the `MiningStreamTransformer` interface in figure 19. Yet, at the same time, they may not implement the `MiningTransformer` interface. It is important to note that the `MiningTransformer` interface principally implies the `MiningStreamTransformer` interface due to the fact that a vector transformation can be regarded as a regular transformation. Although both interfaces are independent of the CWM, XELOPES realizes the `MiningTransformer` interface with the CWM transformation package's infrastructure if the interface is implemented by transformation classes. Special transformations, however, always require a proprietary modeling and implementation [Thess and Bolotnicov, 2004, section 6.6.4].

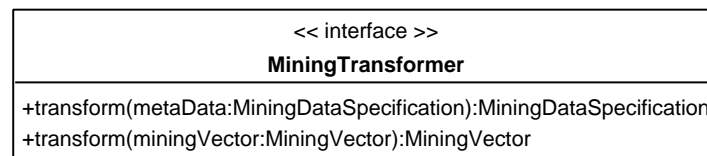


Figure 18: The XELOPES `MiningTransformer` interface (UML class diagram).

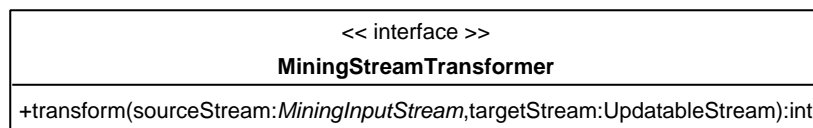


Figure 19: The XELOPES `MiningStreamTransformer` interface (UML class diagram).

A vector transformation T is given. The question arises how the smallest transformation units, the “atoms” of a vector transformation, can be characterized. It is obvious that for a modular transformation design allowing for the composition of complex transformations from several simple transformations, the atoms must be known.

On the one hand, vector transformations can be concatenated given that the attributes match. This means that two vector transformations T_1 and T_2 with matching attributes can be executed *consecutively* to create a new vector transformation, for instance, $T = T_2 \circ T_1$ meaning that T_2 is applied to the results of T_1 . In other words, T can be decomposed *vertically* into T_1 and T_2 .

On the other hand, a vector transformation T may be composed of two or more transformations that are executed *in parallel*. Figure 20 depicts an exemplary vector transformation that can be decomposed *horizontally*. A vector transformation is referred to as *indecomposable*, if it can neither be decomposed horizontally nor vertically. An indecomposable vector transformation can be regarded as an atom.

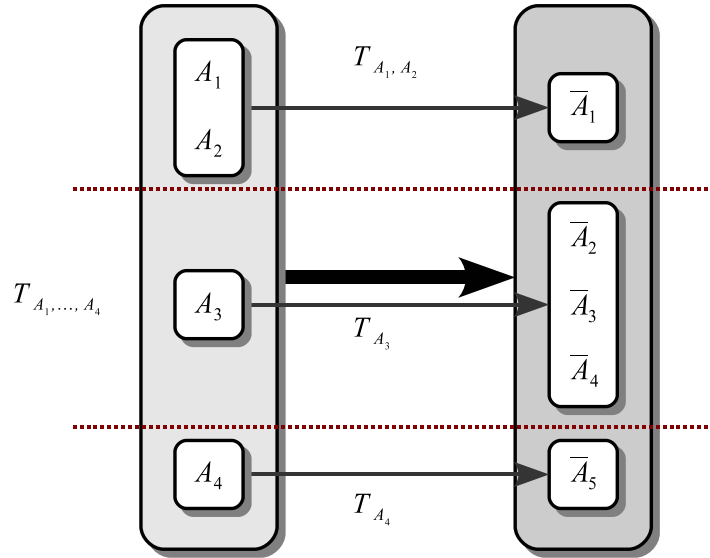


Figure 20: Horizontal decomposition of a vector transformation.

Example (Describing a Horizontal Composition as a Vertical Composition). Given are attributes A_1, \dots, A_6 and two vector transformations

$$T_{A_1, A_2, A_3} : A_1, A_2, A_3 \rightarrow A'_1, \dots, A'_4$$

and

$$T_{A_4, A_5} : A_4, A_5 \rightarrow \bar{A}_1.$$

Obviously, both vector transformations can be composed horizontally. However, from a computational perspective, it may not be desirable or feasible that both vector transformations are executed in parallel. Instead, it may be advisable to execute them consecutively. This can be achieved by adding identical vector transformations as follows:

$$T_{A_1, \dots, A_6} = \begin{pmatrix} T_{A_4, A_5} \\ I_{A'_1, \dots, A'_4, A_6} \end{pmatrix} \circ \begin{pmatrix} T_{A_1, A_2, A_3} \\ I_{A_4, A_5, A_6} \end{pmatrix}.$$

In less precise notation, the assembled vector transformation can be written as an ordinary vertical composition

$$T_{A_1, \dots, A_6} = T_{A_4, A_5} \circ T_{A_1, A_2, A_3},$$

bearing in mind that identical vector transformations are implicitly added. It is this concept that is referred to as a *vertical composition* in the remaining discussion of this chapter. So, the overall vector transformation T_{A_1, \dots, A_6} is executed as follows:

$$A_1, \dots, A_6 \xrightarrow{T_{A_1, A_2, A_3}} A'_1, \dots, A'_4, A_4, A_5, A_6 \xrightarrow{T_{A_4, A_5}} A'_1, \dots, A'_4, \bar{A}_1, A_6.$$

Modeling Vector Transformations From now on, the term transformation refers to a vector transformation. XELOPES's model of handling transformations is based on the CWM transformation package. The XELOPES transformation classes are sub-classes of the various CWM transformation classes described in detail in Poole et al. [2003, chapter 10]. Basically, transformations can be composed of atoms. Compound transformations can again be grouped into transformations of higher complexity on different abstraction levels.

This paragraph discusses the CWM transformation model based on the corresponding XELOPES sub-classes [compare Thess and Bolotnicov, 2004, section 5.6]. An in-depth understanding of them is essential for the foundations of the LOORDSM. Following the structure of the CWM transformation package, XELOPES implements three classes representing atoms, that is, indecomposable transformations, referred to as *mappings*: (1) the `OneToOneMapping` class, which, in CWM terms, represents a *feature map*, (2) the `OneToMultipleMapping` class, which, in CWM terms, represents a *classifier feature map*, and (3) the `MultipleToMultipleMapping` class, which, in CWM terms, represents a *classifier map*.

- (1) **One-To-One Mappings.** A one-to-one mapping transforms exactly one attribute into another attribute as shown in figure 21. One-to-one mappings are modeled with the `OneToOneMapping` class. This class serves three purposes: (i) its sub-classes implement the actual one-to-one-mappings, (ii) it implements the methods required to configure and to fire a mapping, and (iii) it implements an XML serialization of instances of the class to WusanML, the XML user interface of the LOORDSM. WusanML is defined as an XML DTD, which is depicted in the appendix chapter F on page 147. Implementational details of the `OneToOneMapping` class are discussed in the appendix section E.1 on page 141 based on a UML class diagram.

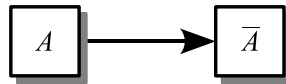


Figure 21: One-to-one mapping.

- (2) **One-To-Multiple Mappings and Multiple-To-One Mappings.** A one-to-multiple mapping transforms one attribute into two or more attributes as shown in figure 22. A multiple-to-one mapping, on the other hand, is the counterpart that maps two or more attributes into one attribute as shown in figure 23 on the next page.

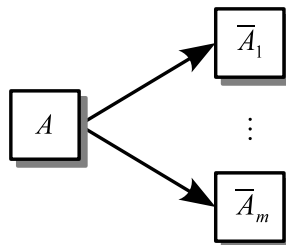


Figure 22: One-to-multiple mapping.

Both mappings are modeled with the `OneToMultipleMapping` class, the details of which are discussed in the appendix section E.2 on page 142. This class serves the same three purposes as discussed above for the `OneToOneMapping` class. Implementational details of the `OneToMultipleMapping` class are discussed in the appendix section E.2 on page 142 based on a UML class diagram.

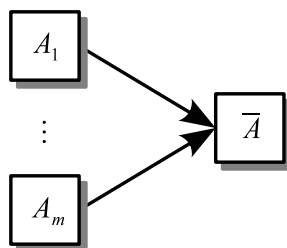


Figure 23: Multiple-to-one mapping.

- (3) **Multiple-To-Multiple Mappings.** A multiple-to-multiple mapping transforms two or more attributes into two or more attributes as shown in figure 24. Multiple-to-multiple mappings are modeled with the `MultipleToMultipleMapping` class. This class serves the same three purposes as discussed above for the `OneToOneMapping` class.

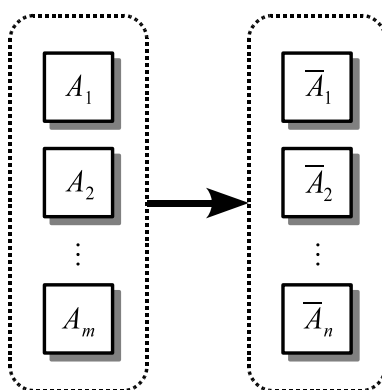


Figure 24: Multiple-to-multiple mapping.

Unlike the first two mapping classes, this mapping class is not restricted to realizing atoms in the strict sense. While figure 24 depicts an atom in the strict sense, that is, the mapping is indecomposable, in practice, it may be desirable to regard certain multiple-to-multiple mappings as “semantic atoms” even if they are actually decomposable. Figure 20 on page 75 demonstrates the latter case of a “decomposable multiple-to-multiple mapping” as opposed to an indecomposable multiple-to-multiple mapping. From a theoretical perspective, the term decomposable multiple-to-multiple mapping is inconsistent, since mappings have been defined as being indecomposable. However, from an implementational perspective, the term is meaningful if atoms are regarded as smallest semantic transformation units. This fuzziness only occurs for multiple-to-multiple mappings as the other mappings are indecomposable at any rate.

Consequently, the `MultipleToMultipleMapping` class can be employed just like the previous two classes, but additionally, the two mapping classes `OneToOneMapping` and `OneToMultipleMapping` can be embedded into a sub-class of the `MultipleToMultipleMapping` class, to contribute to a multiple-to-multiple mapping. However, the `MultipleToMultipleMapping` class cannot be employed to compose two or more multiple-to-multiple mappings. This must be done within the concatenation framework discussed in the following paragraph.

Nesting and Concatenating Transformations On page 74 it was mentioned that transformations in XELOPES are modeled with the `MiningTransformer` interface. This interface

provides two `transform` methods, which are required to fire the real-valued transformation and the meta-data transformation.²⁹ Only classes that actually implement this interface can be employed to execute transformations. However nested or concatenated a transformation is, its execution interface is always the same.

Furthermore, the CWM transformation package implies that one-to-one, one-to-multiple, and multiple-to-one mappings must always be embedded in a multiple-to-multiple mapping class.³⁰ This is due to the fact that, strictly speaking, the CWM transformation package knows only one mapping type, the multiple-to-multiple mapping class, which may comprise the other two mapping types.

Figure 25 illustrates how mappings can be composed to more complex transformations. As discussed before, the `MultipleToMultipleMapping` class features horizontal compositions of the other mapping types, indicated by the horizontal lines in figure 25.³¹ Horizontal composition matches horizontal decomposition, which is illustrated in figure 20 on page 75. All composed mappings are executed *in parallel* on the same meta-data. One or more multiple-to-multiple mappings can then be again horizontally composed with a `MiningTransformationStep` class.

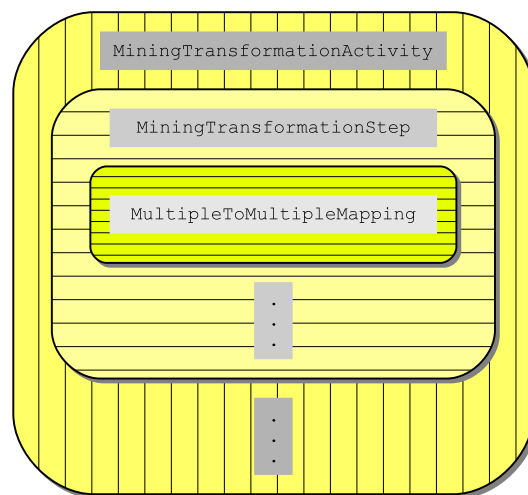


Figure 25: Composing complex transformations.

It is important to note that the multiple-to-multiple mappings included in a `MiningTransformationStep` class may not interfere, that is, only such mappings that can be executed smoothly in parallel may be included. Here, the word “interfere” refers to implementational details.³²

²⁹Compare item i on page 73 and item ii on page 73.

³⁰Compare Poole et al. [2003, figure 6.46].

³¹The execution order of the `MultipleToMultipleMapping` class is discussed in item vi on page 143.

³²With horizontal composition, the same meta-data are employed for all horizontally composed mappings. In XELOPES’s implementation, an attribute can be employed by several of the horizontally composed mappings. When meta-data are changed dynamically within a multiple-to-multiple mapping embedded in a transformation step, strange effects may occur when another multiple-to-multiple mapping of the transformation step makes uses of the modified meta-data, due to the fact that the original and the modified meta-data are linked through pointers where programmers do not expect such a link. This problem primarily results from the fact that XELOPES does not provide methods for duplicating meta-data other than shallow copies, that is, the original and copied meta-data partially share objects. This problem can be avoided by wrapping only one multiple-to-multiple mapping into each transformation step. This loophole would mean that multiple-to-multiple mappings are composed only through vertical compositions. Apart from performance issues, this loophole has no other drawbacks but adds to making the user interface more robust.

Finally, different `MiningTransformationSteps` can be composed vertically with the `MiningTransformationActivity` class. A vertical composition corresponds to the approach followed in the example on page 75. The basic principle of gradually nesting and concatenating transformations and mappings on different abstraction levels directly reflects the CWM approach to assemble complex transformations [compare OMG-CWM, chapter 13]. A detailed example for this fundamental principle is discussed next, as a clear understanding is required for the concrete modeling of ETL transformations. The UML class diagrams of the `MiningTransformationStep` class and the `MiningTransformationActivity` class are depicted in the appendix section E.4 on page 144.

4.2.3.3 Example

This example examines how complex transformations can be assembled. Given the scenario that a statistical or data mining analysis on customer data must be accomplished, an attribute $A_{\text{duration_in_months}}$, which describes the duration in months between a customer's first and last transaction, is required. The source data matrix consists of the attributes shown in table 1. A

<i>Attribute</i>	<i>Type</i>	<i>Meaning</i>
$A_1 = A_{\text{customer_id}}$	numeric	unique customer ID
$A_2 = A_{\text{customer_status}}$	categorical	differentiation between profitable and non-profitable customers
$A_3 = A_{\text{first_timestamp}}$	categorical	timestamp of first completed transaction
$A_4 = A_{\text{last_timestamp}}$	categorical	timestamp of last completed transaction
$A_5 = A_{\text{zip_code}}$	numeric	customer's zip code

Table 1: Source attributes for the exemplary transformation.

transformation that takes these attributes as input and delivers $A_{\text{customer_id}}$ and $A_{\text{duration_in_months}}$ as output is sought. To this end, the overall transformation task must be decomposed into (semantic) atoms. XELOPES offers a wide spectrum of predefined mappings that can be employed as building blocks for transformation modeling.³³ The following transformations must be modeled for this example:

step (1): Remove all the attributes that are not needed for any of the transformation steps or for the final transformed data matrix, namely $A_{\text{customer_status}}$ and $A_{\text{zip_code}}$,

step (2): Derive $A_{\text{duration_in_months}}$ from $A_{\text{first_timestamp}}$ and $A_{\text{last_timestamp}}$, and

step (3): Preserve $A_{\text{customer_id}}$ for the final data matrix.

With WUSAN, the concrete transformation modeling can be accomplished by creating a description of the sought transformation in WusanML (which intuitively follows the nesting and concatenation approach just discussed). The resulting WusanML description is shown in listing 4.1 on the next page. This XML description can then be imported into a `MiningTransformationActivity` class by employing the methods provided by the `PmmlPresentable` interface.³⁴ Alternatively, the transformation can be directly modeled with the Java classes. This is more complex and requires in-depth knowledge about the actual class structure.

³³The mappings of XELOPES are discussed in Thess and Bolotnicov [2004, section 6.6.1-section 6.6.3].

³⁴Compare section E.4 on page 144. Unlike the original XELOPES classes, WUSAN's `MiningTransformationActivity` and the `MiningTransformationStep` classes implement a serialization to WusanML.

Listing 4.1: Modeling the exemplary transformation with WusanML.

```

1 <Transformation>
  <Step>
3     <Mapping className="com.prudsys.pdm.Transform.MultipleToMultiple.\
      →RemoveAttributes" removeSourceAttributes="true">
        <AttributeList />
5         <AttributeList>
          <Name>customer_status</Name>
7           <Name>zip_code</Name>
        </AttributeList>
9     </Mapping>
  </Step>
11 <Step>
    <Mapping className="com.prudsys.pdm.Transform.\
      →MultipleToMultipleMapping" removeSourceAttributes="true">
13       <AttributeList />
      <AttributeList />
15       <Mapping className="wusan.pdm.Transform.OneToMultiple.\
        →DurationInMonths" removeSourceAttributes="true">
          <AttributeList>
17             <Name>first_timestamp</Name>
              <Name>last_timestamp</Name>
19           </AttributeList>
          <AttributeList>
21             <Name>duration_in_months</Name>
          </AttributeList>
23       </Mapping>
    </Mapping>
25 </Step>
  <Step>
27     <Mapping className="com.prudsys.pdm.Transform.MultipleToMultiple.\
      →ChangeAttributeOrder" removeSourceAttributes="true">
        <AttributeList />
29       <AttributeList>
          <Name>customer_id</Name>
31           <Name>duration_in_months</Name>
        </AttributeList>
33     </Mapping>
  </Step>
35 </Transformation>

```

step (1): Create a transformation step $T_{\text{remove}} : * \rightarrow \neg(A_{\text{customer_status}}, A_{\text{zip_code}})$, which removes all the attributes that do not belong to the final transformed data matrix or that are not required during subsequent transformation steps, that is, $A_{\text{customer_status}}$ and $A_{\text{zip_code}}$ are removed by T_{remove} from the meta-data $*$.³⁵ The transformation step encapsulates the multiple-to-multiple mapping class `RemoveAttributes`.

step (2): Create another transformation step that computes $A_{\text{duration_in_months}}$, that is,

$$T_{\text{calculate_duration}} : A_{\text{first_timestamp}}, A_{\text{last_timestamp}} \rightarrow A_{\text{duration_in_months}},$$

which embeds the proprietary mapping class `DurationInMonths`.³⁶

³⁵This less precise notation means that all available attributes are taken as source attributes.

³⁶According to the discussion on page 78, the latter mapping must be embedded in a `MultipleToMultipleMapping` class.

step (3): No transformation is necessary, since the transformations are composed vertically. Following the example on page 75, identical mappings are filled in where necessary. In order to keep the desired sequence of target attributes, a mapping that produces the designated attribute order is added as a last transformation step.

Putting it all together leads to the following transformation (the attribute realignment mapping is not considered):

$$T_{\text{calculate_duration}} \circ T_{\text{remove}} : * \rightarrow A_{\text{customer_id}}, A_{\text{duration_in_months}}.$$

This looks quite trivial due to its compact notation. Yet, disassembling the overall transformation leads to clarity about exactly what happens during its execution:

(1) The transformation starts with the projection

$$T_{\text{remove}} : A_1, \dots, A_5 \rightarrow A_1, A_3, A_4,$$

which can be divided into a real-valued transformation

$$t_{A_1, \dots, A_5}^{\text{remove}} : (\mathbb{R} \cup \{\vartheta\})^5 \rightarrow (\mathbb{R} \cup \{\vartheta\})^3$$

and a meta-data transformation

$$T_{M_{A_1, \dots, A_5}}^{\text{remove}} : M_{A_1, \dots, A_5} \rightarrow M_{A_1, A_3, A_4}.$$

(2) Executing $T_{\text{calculate_duration}}$ on the image of T_{remove} implies the application of the meta-data

$$M_{A_1, A_3, A_4} : (\mathbb{R} \cup \{\vartheta\})^3 \rightarrow (M(A_1) \cup \{\vartheta\}) \times (M(A_3) \cup \{\vartheta\}) \times (M(A_4) \cup \{\vartheta\})$$

as a first step. This, in turn, interprets the output of T_{remove} taken as input for

$$t_{A_1, A_3, A_4}^{\text{calculate_duration}} : (\mathbb{R} \cup \{\vartheta\})^2 \rightarrow \mathbb{R} \cup \{\vartheta\},$$

which is accompanied by the meta-data transformation

$$T_{M_{A_1, A_3, A_4}}^{\text{calculate_duration}} : M_{A_3, A_4} \rightarrow M_{A_{\text{duration_in_months}}}.$$

As mentioned before, a vertical composition involves the execution of the identical transformation $I_{A_1} : A_1 \rightarrow A_1$ in parallel to $T_{\text{calculate_duration}}$, since otherwise the meta-data would not match.

The modular conception for transformation modeling calls for a collection of ready-to-use mappings and transformations, that is, building blocks that can be employed as constituents for complex transformations. To this end, WUSAN provides a variety of WUA-specific mappings that can be leveraged for modeling ETL transformations for this domain. However, if a required mapping is not available, it must be implemented from scratch and made available in WUSAN. As WusanML employs the names of the Java mapping classes for transformation modeling, a new mapping class can be used without any further adaptations on WusanML.

All mapping types are modeled with the `Mapping` tag in WusanML. The different mapping types are distinguished by the number of `Name` tags enclosed in the source and target `AttributeList` tags.³⁷

³⁷Compare the WusanML DTD in listing F.1 on page 147.

4.2.3.4 Summary

This section presented the last milestone towards the LOORDSM: transformation modeling. Initially, a general understanding of transformations was introduced. It was concluded that for ETL transformation modeling, vector transformations are crucial and must be further investigated. The smallest transformation units have been identified as (semantic) atoms modeled with one-to-one, one-to-multiple, multiple-to-one, and multiple-to-multiple mappings. While their implementational details have been postponed to the appendix chapter E on page 141, their employment and compositions have been discussed, allowing users to create complex vector transformations with WusanML.

It is this XML interface that lays the foundations for the LOORDSM's road capability in practical projects. This aspect is renewed in chapter 5 when a showcase example is discussed. The LOORDSM, which is discussed next, assembles all building blocks derived so far to the overall model, capturing the entire ETL process and discharging into the WusanML interface in chapter F on page 147.

4.3 The LOORDSM

In the previous section, the three prerequisites for the LOORDSM have been staged: a meta-data model, streams, and transformation modeling. At the end of chapter 3, it was concluded that a data storage model is sought: this should not only model the physical data storage of WUSAN's data warehouse but also provide a framework for the implementation of associated ETL transformations such that it meets the requirements mentioned in item 1 on page 42 conforming to the CWM. The LOORDSM brings together all building blocks prepared so far and assembles them into a higher-ranking ETL transformation and data storage model.

The LOORDSM is introduced in this section in two steps. First, in section 4.3.1 on the next page, the formal model that describes the basic ideas of the LOORDSM is presented. Second, in section 4.3.2 on page 86 its concrete realization in Java is discussed in detail on the basis of a comprehensive UML class diagram, the nuts and bolts perspective that goes into some implementational details relevant for employing the WusanML XML user interface.

Remark. This section assumes that the reader is familiar with the foundations of *dimensional modeling*. Kimball and Merz [2000, p. 364] define dimensional modeling as

“A methodology for modeling data that starts from a set of base measurement events and constructs a table called the fact table, generally with one record for each discrete measurement. This fact table is then surrounded by a set of dimension tables, describing precisely what is known in the context of each measurement record. Because of the characteristic structure of a dimensional model, it is often called a star schema. [...]”

Dimensional modeling is the standard approach to realizing ROLAP. As such, its employment for WUSAN was motivated in section 3.3.3.3 on page 54. A comprehensive introduction to dimensional modeling can be found in Kimball and Ross [2002].

4.3.1 The LOORDSM from a Theoretical Perspective

4.3.1.1 Outline

In section 4.1 on page 59, it has been discussed that the LOORDSM belongs to WUSAN's ETL layer.³⁸ The LOORDSM creates a data flow between a source stream and a target stream as shown in figure 26. The source stream can be modeled with any of the streams discussed in section 4.2.2.2 on page 70 and accesses application server logs or related data sources for the WUA domain. The target stream is modeled with a database stream in which the underlying database table is part of a superordinate data model, for instance, a data mart.

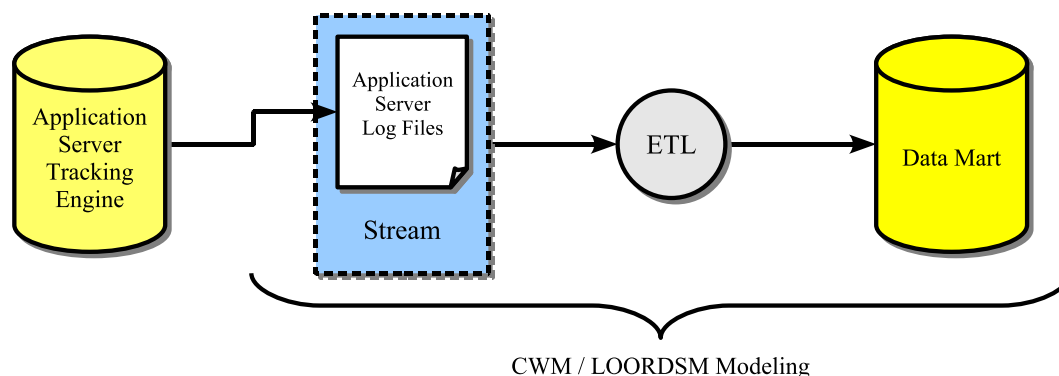


Figure 26: Data flow for populating a data mart.

The link between the source stream and the target stream is created with the LOORDSM within WUSAN's ETL layer. Generally, *one* source stream is employed to populate *two or more* target streams. In the degenerated case, only *one* target stream is populated.

The LOORDSM comprises four constituents: (i) *star schemas*, which describe the tables of a data mart to be populated simultaneously from the same source stream, (ii) *dimensions*, which are contained in one or more star schemas, (iii) *fact tables*, each of which is assigned to exactly one star schema being its core, and (iv) *transformations*, each of which is assigned to a dimension. Star schemas can be regarded as the containers for the ETL framework, that is, they contain information about target streams and ETL transformations.³⁹

4.3.1.2 Mathematical Model

The first definition of the mathematical model characterizes the mapping that is required to generate primary keys when vectors are inserted into dimensions. Although not obvious at this point, the primary key mapping is an important constituent of the LOORDSM, which influences a dimension's behavior when vectors are inserted. Its role becomes clearer in section 4.3.2 on page 86 where the LOORDSM is discussed from a more practical UML perspective.

Definition 4.11 (Primary Key Mapping). Given attributes A_1, \dots, A_m and appendant meta-data M_{A_1, \dots, A_m} , let

$$p : M(A_1) \cup \Theta(A_1) \times \dots \times M(A_m) \cup \Theta(A_m) \rightarrow \mathcal{P}$$

³⁸Compare figure 16 on page 60.

³⁹It is important to mention dimensions are the only objects that actual host ETL transformations. Coming to the point, a dimension knows which target stream is assigned to it and what data is required to populate that stream. This knowledge is contained in its assigned ETL transformation.

be a one-to-one and onto mapping into a user-defined finite-dimensional primary key domain \mathcal{P} . The mapping $p \circ M_{A_1, \dots, A_m}$ maps any vector $\mathbf{a} \in (\mathbb{R} \cup \{\emptyset\})^m$ to a unique primary key. p is called a *primary key mapping*.

The first constituent of the LOORDSM is characterized in the following definition. A *dimension* comprises a stream (extended by a primary key), its assigned meta-data, and a primary key mapping, which generates the primary keys for vector insertions.

Definition 4.12 (Dimension). Let \mathbf{A} be an $n \times m$ data matrix, let p be a compatible primary key mapping, and let $\mathbf{p} := p(M_{A_1, \dots, A_m}(\mathbf{A}))$ be the corresponding *primary key vector/matrix*.⁴⁰ Then,

$$D := \left((\mathbf{p} | M_{A_1, \dots, A_m}(\mathbf{A})), p \right) \quad (4.10)$$

is called a *dimension*. $(\mathbf{p} | M_{A_1, \dots, A_m}(\mathbf{A}))$ denotes a stream extended by the primary key.

Similarly to dimensional modeling, two types of dimensions can be identified, each of which is implemented in a separate class of the UML class diagram to be discussed in section 4.3.2 on page 86. Once again, it must be emphasized that all definitions in this section refer to the ETL level in figure 16 on page 60, not the data storage layer.

If D consists of two or more attributes none of which is a foreign key attribute⁴¹ and if the primary key mapping produces one-dimensional primary keys, D is referred to as a *regular dimension*.⁴² Further, if D consists of m attributes with the primary key mapping being the identical mapping on these attributes, D is referred to as a *degenerate dimension*. The third alternative dimension type is covered in the next definition. *Fact tables* cover the case when foreign key attributes occur in a dimension.

Definition 4.13 (Fact Table). Let F be a dimension with $m > 1$. If A_1, \dots, A_m represent foreign key attributes, that is, $M(A_{i_1}) \times \dots \times M(A_{i_m}) \subseteq \mathcal{P}_i$ with \mathcal{P}_i being the primary key domain of dimension D_i , $i = 1, \dots, \ell$, then F is called a *fact table*.

This definition states that a fact table comprises two or more foreign key attributes, each of which is part of a foreign key referencing a dimension. Any dimension type is acceptable, that is, regular dimensions, degenerate dimensions, or fact tables. The involved dimensions are referenced through one-dimensional foreign keys except for degenerate dimensions: they are obviously completely embedded into the fact table due to the fact that their primary key comprises the entire dimension vector.

Finally, the definition coming next characterizes the container for dimensions and fact tables, a *star schema*. At this point, transformations are ignored in the mathematical model as they can be better dealt with in the UML model discussed later.

Definition 4.14 (Star Schema). Let D_1, \dots, D_ℓ be dimensions and let F be a fact table such that dimension D_i is assigned to one or more attributes of F by a foreign key reference. Then, the tuple $\mathcal{S} := (F, D_1, \dots, D_\ell)$ denotes a *star schema*.

Regular dimensions and fact tables are assigned by one attribute (the foreign key attribute), while degenerate dimensions are assigned by one or more attributes, since they are entirely embedded into the fact table.

⁴⁰Normally, \mathbf{p} is a vector.

⁴¹A foreign key attribute is part of a foreign key consisting of one or more attributes.

⁴²A regular dimension with only one attribute is not meaningful, because it can be more easily modeled as a degenerate dimension.

The above definitions reflect the common understanding of star schemas in dimensional modeling [compare, for example, Martyn, 2004]. However, as mentioned before, there a significant difference between these definitions and the LOORDSM: while the common understanding of a star schema refers to a relational data storage model, that is, the mere database tables in an RDBMS, definition 4.14 on the preceding page and all dependent definitions refer to the classes of the LOORDSM that operate on the *physical database tables* in order to realize the ETL process.

During the ETL process, two tasks must be accomplished for a star schema: (i) populate the target streams, that is, the physical database tables on which the dimensions operate and (ii) populate the stream that is assigned to the fact table, also a database stream. Both tasks can be performed *sequentially*, *in parallel*, or as a *combination* of both approaches. In any case, whenever a vector is inserted into the fact table, it must contain valid foreign key references, that is, valid references to the vectors stored in the dimensions. So as to populate a dimension with meta-data M_{A_1, \dots, A_m} , a source stream $M_{\tilde{A}_1, \dots, \tilde{A}_\ell}(\mathbf{A})$ and a transformation

$$T_{\tilde{A}_1, \dots, \tilde{A}_\ell} : \tilde{A}_1, \dots, \tilde{A}_\ell \rightarrow A_1, \dots, A_m, \quad (4.11)$$

referred to as an *ETL transformation*, are required. In general, $T_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ is composed of the following two components:

(1) A *raw ETL transformation*

$$T_{\hat{A}_1, \dots, \hat{A}_\ell} : \hat{A}_1, \dots, \hat{A}_\ell \rightarrow A_1, \dots, A_m \quad (4.12)$$

with $\{\hat{A}_1, \dots, \hat{A}_\ell\} \subseteq \{\tilde{A}_1, \dots, \tilde{A}_\ell\}$, that is, the raw ETL transformation comprises equal or less attributes than the overall transformation in equation (4.11).

(2) A *projection* Π_{A_1, \dots, A_m} that removes all attributes not required to populate the dimension.

Consequently, equation (4.11) can be rephrased as a vertical composition of a raw ETL transformation and a projection:

$$T_{\tilde{A}_1, \dots, \tilde{A}_\ell} = \Pi_{A_1, \dots, A_m} \circ T_{\hat{A}_1, \dots, \hat{A}_\ell}. \quad (4.13)$$

Above equation might puzzle attentive readers, since the meta-data of $T_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ and $T_{\hat{A}_1, \dots, \hat{A}_\ell}$ do not match in general. The answer to this problem is rephrasing equation (4.13) to

$$T_{\tilde{A}_1, \dots, \tilde{A}_\ell} = \Pi_{A_1, \dots, A_m} \circ \left(T_{\hat{A}_1, \dots, \hat{A}_\ell} \circ I_{\tilde{A}_1, \dots, \tilde{A}_\ell} \right).$$

Here, the first concatenation $T_{\hat{A}_1, \dots, \hat{A}_\ell} \circ I_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ is a vertical composition following the example on page 75, while the second is an ordinary vertical composition with matching meta-data.

The question arises as to how the splitting of an ETL transformation into a raw ETL transformation and a projection in equation (4.13) is beneficial for the actual ETL modeling. If the ETL transformation equation (4.11), which depends on the meta-data $M_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ of the source stream, was directly modeled, in general, it could not be applied to a source stream with meta-data different from $M_{\tilde{A}_1, \dots, \tilde{A}_\ell}$. However, if the raw transformation $T_{\hat{A}_1, \dots, \hat{A}_\ell}$ is modeled, only the projection Π_{A_1, \dots, A_m} must be adapted to the meta-data of the source stream. The classes of the LOORDSM allow the projection to be created automatically: modeling the raw transformations makes the ETL process more robust in view of changing source streams, since the actual ETL transformations can be created semi-automatically from a static raw transformation and a dynamic projection.⁴³

⁴³In section E.1 on page 141, it is mentioned that attribute selections are realized with attribute names. Therefore, if attribute names of the source stream change, it may be necessary to adjust the names of the source attributes

4.3.1.3 Summary

The theoretical perspective of the LOORDSM pointed out two issues relevant for the further discussion in the following section. First, it clarified the coherences among its constituents. These coherences are directly reflected in the design of the LOORDSM's UML classes. Second, it pointed out how ETL transformations can be split into a static part and a dynamic part. This makes transformation modeling more robust in view of changing source streams, as the UML classes realize this feature allowing users to restrict themselves to modeling the static parts only. This leads to increased user friendliness, since it is not necessary for users to bother with modeling tedious attribute eliminations for every transformation.⁴⁴

4.3.2 The LOORDSM from an Implementational Perspective

This section switches to a programmer's perspective by discussing WUSAN's central UML class diagram in figure 27 on the facing page. Its main purpose is to model star schemas and their associated ETL processes in the ETL layer of WUSAN.⁴⁵ All classes support a WusanML serialization over the `PmmlPresentable` interface.⁴⁶ This is required for two purposes: (i) Star schemas including their entire ETL process can be modeled completely or partially in WusanML. Their XML descriptions can then be imported into WUSAN and translated into concrete Java classes for deployment. As mentioned before, WusanML is currently the primary user interface for ETL modeling for the WUSAN prototype. (ii) Conversely, star schemas and their complete ETL process can be exported to WusanML and are thereby made persistent.⁴⁷

Potential users of WUSAN do not have to comprehend this section in every detail. However, this section is of great help when it comes to structuring a complex ETL process in terms of splitting it up into several less complex sub-processes. Users do not have to deal with the classes illustrated in figure 27 on the next page. However, employing WusanML for ETL modeling means employing these classes *indirectly*, since the instances required to execute the ETL process are generated automatically from WusanML models. Some of the subtleties of WusanML directly reflect the subtleties of those Java classes. Therefore, it is highly recommended to work through this section prior to employing WUSAN's LOORDSM in practice.

4.3.2.1 Dimensions

All dimensions in terms of definition 4.12 on page 84 implement the `AbstractDimension` interface, which constitutes the basic functionality of a dimension. It provides the `createEtlTransformation` method, which creates the ETL transformation according to the remark above. The `etl` method applies this transformation to a vector from the source stream. The `getRoleMetaData` method delivers the attributes employed to model the foreign key

of the raw ETL transformation. This can be accomplished by altering its WusanML serialization. This does not affect the actual (static) structure of the raw ETL transformation.

⁴⁴The reader should keep this feature in mind and recall it in chapter 5 on page 95 and appendix chapter G on page 149, since all elaborated WusanML transformation modeling examples realize the static part only, conforming to the theoretical findings.

⁴⁵Recall figure 16 on page 60.

⁴⁶At this point, the LOORDSM ties in with the WusanML serialization for transformations discussed in section 4.2.3 on page 72. As the WusanML DTD makes use of the `PMML DataField` tag (which models attributes) and all dependent tags, the `PmmlPresentable` interface is also employed for serialization to WusanML, since it allows XELOPES's PMML serialization of the `MiningDataSpecification` class to be partially reused. The WusanML DTD is illustrated in the appendix chapter F on page 147.

⁴⁷This mechanism is explained in more detail in the appendix section D.6 on page 133.

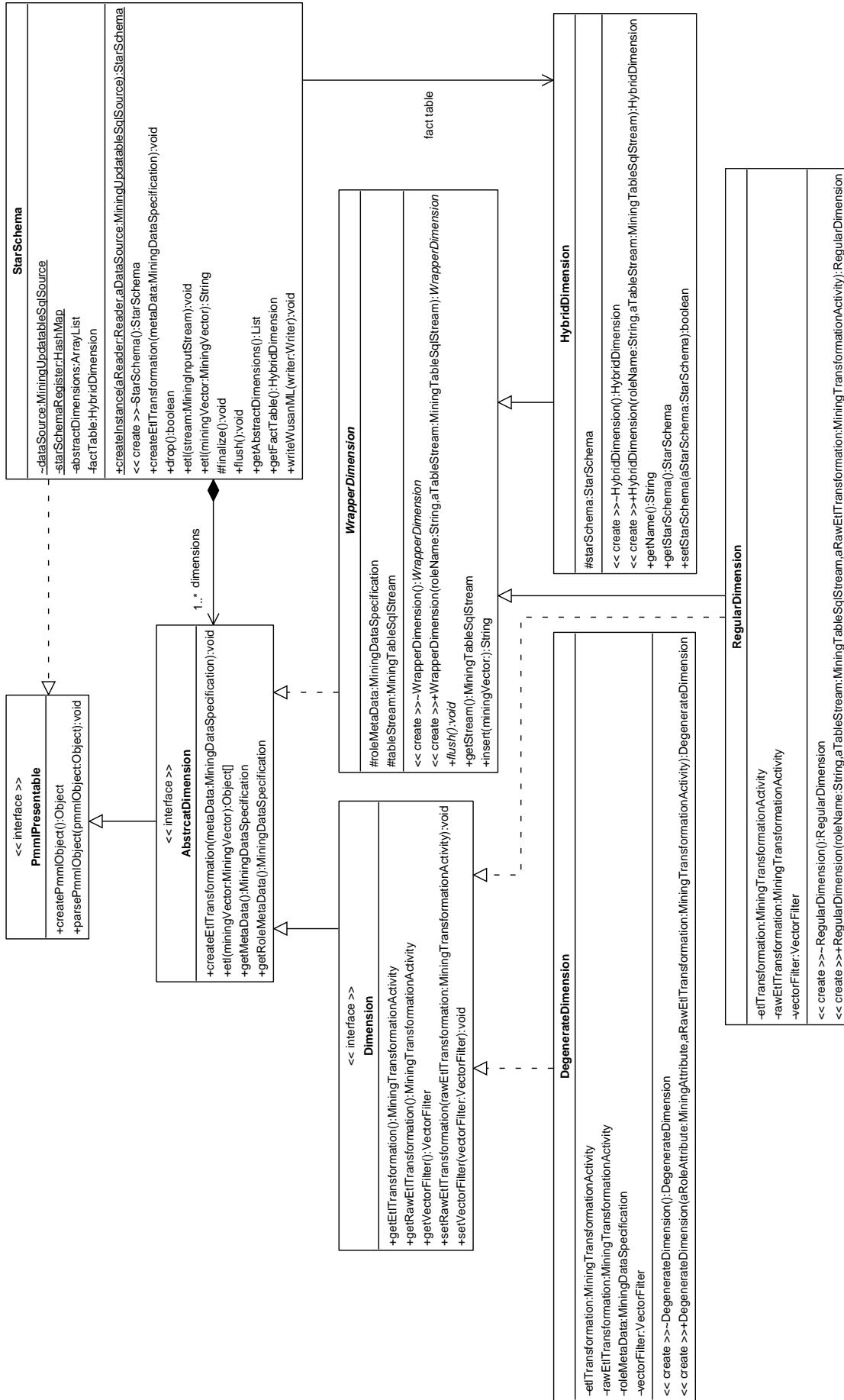


Figure 27: The LOORDSM illustrated as a UML class diagram.

of the fact table pointing to this dimension. Figure 28 depicts how the meta-data of a fact table are composed of the attributes of the role meta-data of its assigned dimensions.⁴⁸ The `getMetaData` method returns the meta-data of the dimension.

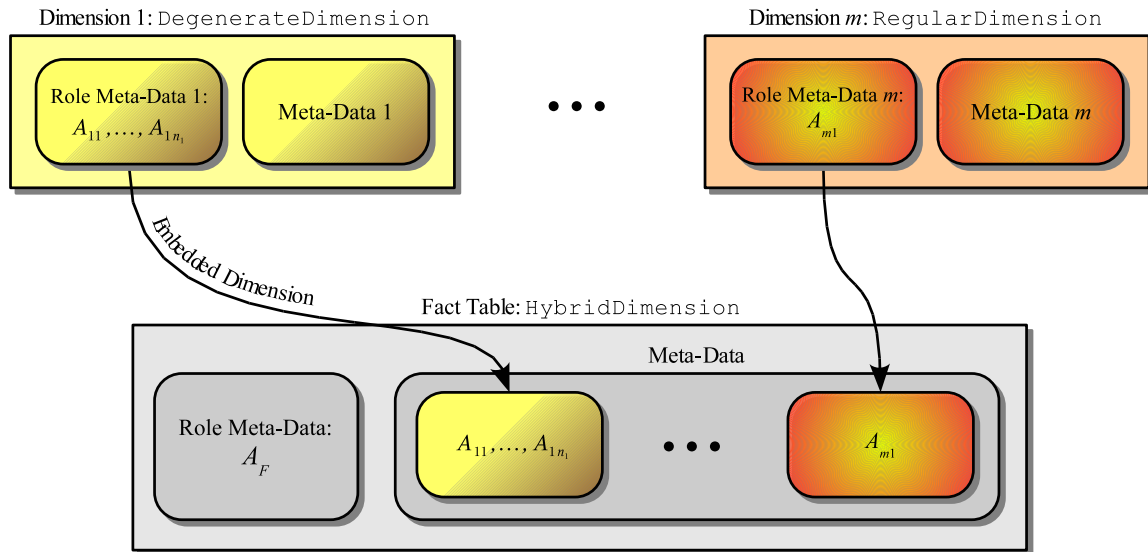


Figure 28: Role meta-data comprise the meta-data of a fact table in a star schema.

If a dimension is not a fact table, it must implement the `Dimension` interface, which provides administrative methods for the ETL transformations and for the assigned vector filter.⁴⁹ According to figure 27 on the preceding page, the `RegularDimension` and the `DegenerateDimension` classes implement this interface. They correspond to the conforming notions defined previously. Both classes contain the `rawEtlTransformation` and the `etlTransformation` variables, which also correspond to the conforming notions defined in section 4.3.1.2 on page 83.

(1) **Regular Dimensions.** Following figure 27 on the preceding page, the `RegularDimension` class is a sub-class of the `WrapperDimension` class, which embeds a `MiningTableSqlStream`. So as to improve the performance of the `MiningTableSqlStream` class⁵⁰ in view of the two typical ETL operations *vector selections* and *vector insertions*, the class implements two caches that accelerate both operations. In order to avoid concurrent caches for the same physical database table (ultimately leading to an abnormal behavior of the resulting streams), only *one* instance of the `MiningTableSqlStream` class can be instantiated for each physical database table. However, each instance of the `MiningTableSqlStream` class can be embedded in more than one instance of a `RegularDimension`.

The `MiningTableSqlStream` class provides several methods that are relevant to the ETL process. The `getKey` method takes a vector as a parameter and returns its primary

⁴⁸The denomination “role meta-data” stems from the fact that the name of the attributes of the role meta-data describe the role the dimension plays in a star schema. According to figure 27 on the preceding page, a fact table also provides role meta-data, since a complete star schema can be employed as a dimension in another star schema. The role meta-data consist of one attribute only for regular dimensions and fact tables.

⁴⁹The `VectorFilter` class can be employed to prevent a vector from being inserted into a dimension if it matches the filter criterion configured in an instance of this class (compare appendix section D.7 on page 137). A fact table disposes of neither an ETL transformation nor a vector filter, since it contains foreign keys only.

⁵⁰Compare item 3 on page 71 and section D.4 on page 133.

key, if it is contained in the stream. The `insert` method computes a primary key for the parameter vector. Then, the parameter vector is inserted only if the computed primary key does not yet exist in the stream. The `PrimaryKeyGenerator` class implements the primary key mapping p , which was introduced in definition 4.11 on page 83. The behavior of the `insert` method is influenced by p . For instance, if p calculates a digest with the MD5 [compare Rivest, 1992], a certain vector can be inserted into the stream only once.⁵¹ This is the normal behavior for dimensions.

- (2) **Degenerate Dimensions.** A `DegenerateDimension` in figure 27 on page 87 consists of one or several attributes, referred to as *role attributes* and pooled in the *role meta-data*. As opposed to the `RegularDimension` class, the `DegenerateDimension` class does not embed a stream that is referenced by a fact table, since the foreign keys, which are embedded in the fact table, contain all relevant information.⁵²

4.3.2.2 Fact Tables

According to definition 4.13 on page 84, a fact table is a dimension that consists of foreign key attributes only. It is modeled with the `HybridDimension` class in figure 27 on page 87, a sub-class of the `WrapperDimension` class. A fact table must be *uniquely* assigned to a star schema. Due to the fact that a `HybridDimension` implements the `AbstractDimension` interface, it can be employed as a dimension in another star schema. This means that a `HybridDimension` or rather its assigned star schema acts as a dimension.⁵³ The principle of nesting star schemas with this approach is illustrated in figure 29 on the next page.

4.3.2.3 Star Schemas

Every single dimension can be regarded as a building block that contributes one aspect to the overall ETL process of a star schema. The `StarSchema` class in figure 27 on page 87 hosts many such building blocks and forges the overall ETL transformation from all the building blocks it contains. According to definition 4.14 on page 84, a star schema comprises a fact table and at least one dimension.⁵⁴ The `StarSchema` class contains one fact table and an ordered list of dimensions, each of which is modeled as described in the previous sections. Every dimension comprises a raw ETL transformation that is modeled with the `Mining-TransformationActivity` class.⁵⁵

⁵¹The MD5 is a cryptographic hash function with a 128-bit hash value. The idea is to calculate a unique digest for every combination of characters of arbitrary length. Only recently, Wang et al. [2004] managed to detect a collision for the MD5, that is, this hash function is actually broken. Since there are “only” 2^{128} MD5 outputs but an infinite number of inputs, it is obvious that such collisions exist; yet, it had been previously believed to be impractically difficult to find a collision. In consequence, two different vectors might result in the same digest and, hence, only one of them could be inserted into the stream – an undesirable behavior for dimensions. However, a random collision for the MD5 is extremely improbable [compare Menezes et al., 1996, chapter 9] and so it is extremely unlikely that a vector cannot be inserted into a dimension due to a collision.

⁵²The primary keys for regular dimensions are abstract or at best summarize the vector provided as an argument of the primary key mapping. In contrast, the primary keys of degenerate dimensions do not summarize the vectors (compare the discussion on page 84). Furthermore, the meta-data of a degenerate dimension are the same as its role meta-data, that is, the `getMetaData` and `getRoleMetaData` methods yield the same meta-data.

⁵³A normalized dimension is a dimension that is decomposed into a star schema structure, denoted as a *snow flake* dimension by Kimball and Merz [2000, p. 381].

⁵⁴However, this dimension may be degenerate meaning that the star schema comprises *one* physical database table only, that is, the star schema is *degenerate*. An example is discussed in section G.1.4 on page 158.

⁵⁵Compare the paragraph about nesting and concatenating transformations on page 77.

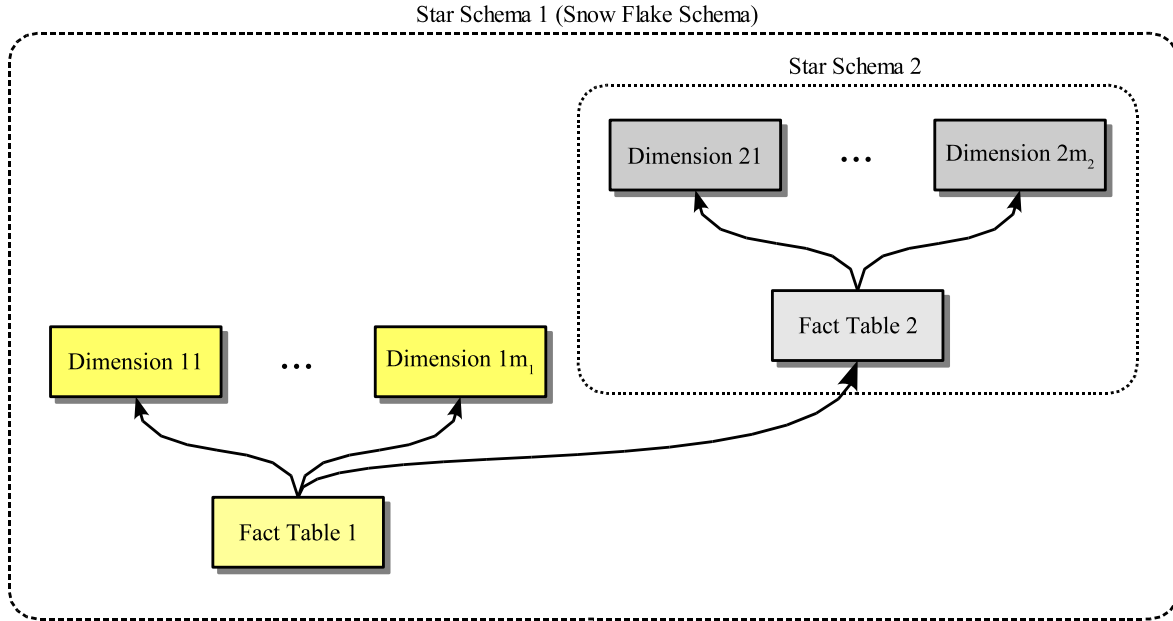


Figure 29: Nesting of star schemas to model a more complex snowflake schema.

A star schema can be created only with the static `createInstance` method, which requires a WusanML description of the star schema as input. So as to keep the XML description of a star schema concise, the dimensions modeled with WusanML do not contain information about the meta-data of the underlying physical database table, since it is assumed that the required tables for regular dimensions exist when the star schema is created.⁵⁶ In this case, their meta-data can be retrieved through the `MiningUpdatableSqlSource` class, which is discussed in the appendix section D.6 on page 133. In order to create a database table for a `RegularDimension`, the following two steps must be taken: (i) create the meta-data for the dimension with WusanML and convert it into a `MiningDataSpecification` and (ii) invoke the `createTable` method for the created `MiningDataSpecification`, in order to generate the physical database table for the regular dimension.⁵⁷

Given is a star schema $\mathcal{S}_1 = (F_1, D_{11}, \dots, D_{1m_1}, F_2)$ consisting of D_{11}, \dots, D_{1m_1} , each of which is a dimension but not a fact table, and the hybrid dimension F_2 . F_2 is the fact table of the star schema $\mathcal{S}_2 = (F_2, D_{21}, \dots, D_{2m_2})$ with D_{21}, \dots, D_{2m_2} also being dimensions but not fact tables. This represents the scenario depicted in figure 29. Furthermore, let $M_{A_1, \dots, A_\ell}(\mathbf{A})$ be the source stream for the ETL process. It is necessary to ensure that A_1, \dots, A_ℓ contain all the relevant information to populate \mathcal{S}_1 and \mathcal{S}_2 , which is explicitly linked by \mathcal{S}_1 .

There exist two alternative approaches for executing the ETL process that populates all the physical database tables embedded in the dimensions $D_{11}, \dots, D_{1m_1}, F_1, D_{21}, \dots, D_{2m_2}$, and F_2 : (1) the *instantaneous ETL approach* and (2) the *gradual ETL approach*, each of which is briefly discussed below.

Remark. As shown in figure 26 on page 83, the entire purpose of the ETL process is to populate a set of relational database tables, which are the basis for ROLAP and data mining analyses. The LOORDSM in figure 27 on page 87 is merely a “toolbox” to model and to deploy the ETL process and to populate the physical database tables. The classes of the LOORDSM are not

⁵⁶The fact table is created automatically, if it does not exist.

⁵⁷Examples are discussed in section G.1.1 on page 149 and subsequent sections.

directly involved in the analytical activities of the data mining and OLAP layers⁵⁸, since these activities operate on streams, that is, the data mining layer (as in the case of data mining) or the physical database tables, that is, the OLAP layer (as in the case of OLAP).

- (1) **Instantaneous ETL Approach.** In this approach, the star schemas \mathcal{S}_1 and \mathcal{S}_2 are directly linked by adding F_2 as a dimension to \mathcal{S}_1 as depicted in figure 29 on the preceding page. Consequently, the ETL process can be executed *instantaneously* for \mathcal{S}_1 and \mathcal{S}_2 . To initialize the ETL transformations, the `createEtlTransformation` method of \mathcal{S}_1 must be invoked for the meta-data of the source stream. This method invokes the corresponding methods of D_{11}, \dots, D_{1m_1} with the same parameter and initializes each dimension. Then, \mathcal{S}_2 is retrieved by invoking the `getStarSchema` method of F_2 , and the `createEtlTransformation` method of \mathcal{S}_2 is called to start the *recursive initializations* for \mathcal{S}_2 . The execution of the ETL transformation operates analogously. If the `etl` method of the star schema is invoked passing a vector from the source stream, the vector is dispatched to the `etl` methods of D_{11}, \dots, D_{1m_1} , each of which executes its ETL transformation for the forwarded vector. The resulting vector (if not yet contained) is then inserted into the target stream, which is embedded in the dimension, and its primary key is returned. The star schema collects all delivered primary keys in order to assemble the vector of foreign keys to be inserted into the fact table.⁵⁹ When the `etl` method of F_2 is invoked, it calls the `etl` method of \mathcal{S}_2 , which analogously executes the `etl` transformations for D_{21}, \dots, D_{2m_2} .
- (2) **Gradual ETL Approach.** In this approach, the star schemas \mathcal{S}_1 and \mathcal{S}_2 are not explicitly linked as before. Figure 30 illustrates the alternative modeling: instead of adding F_2 as a dimension, a one-dimensional degenerate dimension $D_{1(m_1+1)}$ that contains a foreign key reference to F_2 is added. This slight change enables the following approach. First, the ETL process for \mathcal{S}_2 is executed based on a source stream $M_{\tilde{A}_1, \dots, \tilde{A}_\ell}(\mathbf{A})$, that is, the `createEtlTransformation` method of \mathcal{S}_2 must be invoked for the source stream's meta-data $M_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ before the `etl` method of \mathcal{S}_2 can be invoked for vectors of the source stream. Second, the ETL process of \mathcal{S}_1 is invoked for another source stream $M_{A_1, \dots, A_\ell}(\mathbf{A})$ in a similar way. The meta-data M_{A_1, \dots, A_ℓ} of the second source stream must meet the following prerequisite: one of the attributes A_1, \dots, A_ℓ must contain the foreign key information required for the link to \mathcal{S}_2 via the degenerate dimension $D_{1(m_1+1)}$, which is extracted by the corresponding ETL transformation.

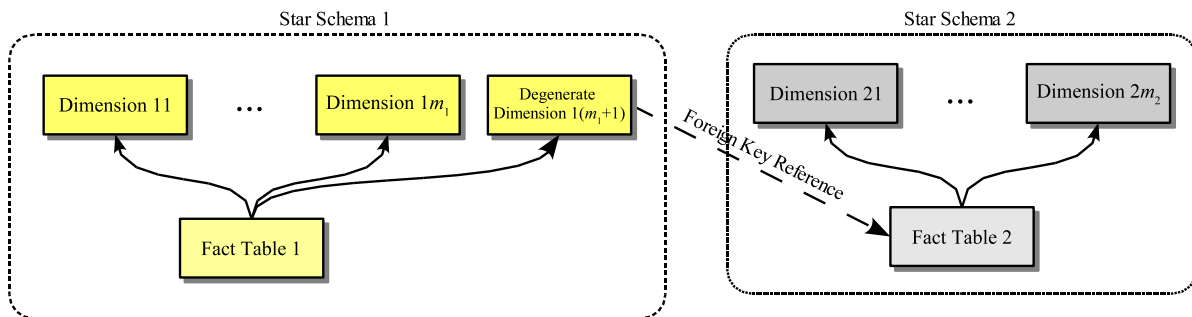


Figure 30: Alternative modeling for the gradual ETL approach.

⁵⁸Recall figure 16 on page 60.

⁵⁹Compare figure 28 on page 88.

The two discussed ETL approaches are revisited in the following chapter. Section 5.2.3.1 on page 104 and section 5.2.3.2 on page 106 both discuss data marts relying on both ETL approaches when they are populated. In view of these examples, it must be emphasized that in figure 30 on the preceding page the second star schema may as well be degenerate.⁶⁰ This means that the the second star schema, which is referenced by the first, comprises only *one* physical database table.

Remark. (1) Both ETL approaches eventually lead to the same set of *physical* database tables.

The instantaneous approach can be employed if all the information required to populate the nested star schemas is available in a *single* source stream. The gradual approach must be employed if different source streams are required to populate the database tables or if the ETL process should be decomposed so as to reduce its complexity.

- (2) It is compulsory to ensure that referential integrity holds after concluding all ETL transformations. If the degenerate dimension $D_{1(m_1+1)}$ in figure 30 on the preceding page contains any NULL values, they must be substituted with a foreign key to a vector of F_2 that again references NULL vectors in *every* assigned dimension. This is due to the fact that the Mondrian OLAP server mentioned in section 3.3.3.4 on page 54 must be able to perform a join operation for *every* row of a fact table.
- (3) Although the discussion in this section has been restricted to two star schemas, the discussed details can be transferred to *arbitrarily nested* star schemas.
- (4) The notion of a *measure* is a fundamental concept for fact tables in dimensional modeling.⁶¹ The LOORDSM models measures within degenerate dimensions.⁶² If a measure can be computed during the ETL process, that is, it is contained in the source stream or it can be computed from a single vector of the source stream, the resulting transformation can be included in the raw ETL transformation of a degenerate dimension and executed on the fly. However, if the measure cannot be computed from a single vector of the ETL source stream, it must be calculated in a proprietary manner *after* the ETL process.⁶³ Nevertheless, a transformation that creates a NULL value for any input can be added to the raw ETL transformation of a degenerate dimension. This results in an empty column in the fact table, which can be filled manually later on.⁶⁴

4.4 Summary

This chapter delved into the details of the LOORDSM, which bridges the gap of ETL modeling identified in the previous chapter. It covers the ETL layer in figure 16 on page 60 and provides the missing building block for WUSAN in figure 15 on page 53, the ETL component. In the introduction of section 3.3 on page 41, it was argued that automating ETL transformations is particularly important for WUA, so as to close the loop of the WUA process. To this end, the LOORDSM contributes the following benefits:

⁶⁰Compare footnote 42 on page 84.

⁶¹Compare the references mentioned in the introduction of section 4.3 on page 82.

⁶²This is the reason why the role meta-data of a degenerate dimension may contain *numeric* attributes as well.

⁶³This means that the measure cannot be computed with a vector transformation (that is, a regular transformation) but requires a special transformation.

⁶⁴This approach ensures that the meta-data of the fact table remain consistent. Should a column for a complex measure be added to the database table with an SQL query, its meta-data stored in WUSAN's meta-data repository would not match the structure of the altered database table any longer and would have to be adjusted manually.

- (1) The LOORDSM with its clearly structured formal mathematical data and transformation model conveys its structured approach to WUSAN's WusanML interface providing users the standardized structured user interface, the definition of which is discussed in the appendix chapter F on page 147. This XML interface reflects the qualities of the mathematical and the UML model of the LOORDSM, which have been discussed in this chapter.
- (2) Clear and structured transformation modeling due to the integration of the CWM transformation package. This approach assumes that a comprehensive library of WUA-specific transformations for transformation modeling is available.
- (3) The LOORDSM is fully compatible to the CWM, thereby allowing for integrating any CWM compatible generic data pools into the ETL process.
- (4) The LOORDSM creates the prospect of automating the preprocessing phase of the WUA process, the very phase which is particularly challenging for the WUA domain.

As mentioned above, so as to fully capitalize on the LOORDSM for WUA, a manifold set of transformations relevant to WUA is required. Since the transformations provided by XELOPES are not sufficient for WUA, additional transformations have been made available in WUSAN in order to establish a solid basis for practical WUA projects.⁶⁵ Any missing transformations can be coded, tested, and easily added to WUSAN due to its structured transformation model.⁶⁶

The following chapter completes this thesis by demonstrating how a concrete WUA-related analytical challenge can be tackled with WUSAN. Additionally, its deployment is demonstrated and the importance of the LOORDSM's WusanML user interface for ETL modeling is highlighted. Hence, chapter 5 returns to the issue raised in chapter 2, namely, how to concretely deploy ECRM as a response to an abruptly changing competitive environment in EC.

⁶⁵For more details see WUSAN's API documentation.

⁶⁶Compare section 4.2.1.4 on page 68.

Chapter V

DEPLOYING THE LOORDSM AND OPEN ISSUES IN WUSAN

This chapter demonstrates the approach to realize a fictitious, yet realistic WUA project with WUSAN aiming at simple real-time personalizations on an EC Web site. As no EC Web site conforming to the application server structure discussed in section 3.2.2 on page 37 was available for research experiments during this thesis, the application scenario from the KDD Cup 2000¹ has been employed as a guideline to develop a businesslike showcase as a thread for this chapter. Section 5.1 characterizes the challenge employed as a showcase in this chapter: the realization of a simple real-time recommender system with WUSAN. To this end, the relevant steps for its deployment are derived and discussed in detail, that is, subsequent sections investigate each step as to WUSAN's capabilities to support the realization of the respective step within the WUA process².

Then, in section 5.2 on page 101, relevant mappings for a closed loop recommendation engine are discussed and it is demonstrated what parts of the underlying WUA process can be automated with WUSAN. Here, the LOORDSM is deployed concretely and appendix chapter G on page 149 completes the details of WUSAN's XML interface, which makes the formal model presented in the previous chapter operational. Finally, in section 5.4 on page 111, open issues and practical problems of WUSAN being subject to future research and development activities are identified.

5.1 Showcase Description

This section develops the challenge of providing real-time recommendations on a fictitious EC Web site. The boundary conditions are set as realistic as possible, to point out the contribution of the LOORDSM towards tackling the challenge of real-time personalization, a paradigm application scenario for WUA, and to detect links to future research necessities.

5.1.1 KDD Cup 2000 Data

The KDD Cup 2000 data employed in this chapter have been gathered using Blue Martini's Web application server, which has been mentioned in section 3.3.1 on page 42. It was stated that this system complies with the application server architecture discussed in section 3.2.2 on page 37 and features all the benefits for data collection cited. It was added, however, that the system is proprietary, very complex, and not available and applicable offhand for academic WUA research projects.

Due to the fact that obtaining real Web usage data from EC Web sites for WUA raises many difficulties in practice, for this thesis – as for most research activities in data mining – the only way out of this dilemma is employing freely disposable data sets. Witten and Frank [2005, preface] mention that organizations generally consider their data repositories as an invaluable

¹The KDD Cup 2000 [Kohavi et al., 2000] is part of the series of yearly data mining competitions started in 1997 and held in conjunction with the ACM KDD conferences.

²Recall section 3.3 on page 41 and the remark on page 32.

corporate asset that is not made available – if accessible at all – without extensive restrictions. Furthermore, if customer data are involved, the *purpose specification principle*³ and the *use limitation principle*⁴, two integral constituents of most privacy policies and legislations, impede any analyses for purposes other than those stated at the time of data collection. This is a severe obstacle for all personalized analyses within WUA research projects.

Consequently, the KDD Cup 2000 data are a good choice for WUA research [Kohavi et al., 2000]: (i) They are available at no cost for research activities. (ii) They comprise anonymous customer information, which includes customer ID, registration information, and registration form questionnaires. (iii) They contain order information at two levels of granularity: (a) *order header*, which includes date/time, discount, tax, total amount, payment, shipping status, and session ID, and (b) *order line*, which includes quantity, price, product, date/time, assortment, and status.⁵ (iv) Furthermore, they contain clickstream information at two levels of granularity: (a) the *session* level, which includes starting and ending date/time, cookie, user agent, referrer, and hit count, and (b) the *page view* level, which includes date/time, sequence number, URL, processing time, product, and assortment. (v) They are partially preprocessed and consolidated, that is, (a) sessions caused by a server monitor have been removed, (b) test users and all their transactions have been deleted, (c) returned and uncompleted orders were removed, (d) some aggregations have been made in view of the KDD Cup 2000 questions, (e) the data have been transformed into C5.0 format⁶, and (f) simple descriptive statistics for every attribute of each file have been computed and made available in HTML reports.⁷

5.1.2 Recommender Systems

According to Schafer et al. [2001], *recommender systems* employ product knowledge, which is either available as hand-coded rules (that is, rules generated by domain experts) or learned from behavioral data through data mining, to guide (potential) customers of EC Web sites through the multitude of offers by recommending products and services the customers will probably like. That is, recommender systems suggest products and services to (potential) customers of EC Web sites to help them to decide which products to purchase.

The authors specify three main objectives as to how recommender systems enhance EC sales (compare section 3.1.3 on page 33):

- (1) **Converting Browsers into Buyers.** That is, users that browse an EC Web site are incited to purchase products and services through appropriate Web personalization⁸ and personalized recommendations.
- (2) **Increasing Cross-Sales.** By suggesting additional products and services to customers, the average order size can be increased if the recommendations are precisely targeted.

³Compare item 2 on page 118.

⁴Compare item 3 on page 118.

⁵Each order consists of one or multiple order lines, each of which is a purchase record of one particular product in a quantity of one or more.

⁶Compare item 2iii on page 71.

⁷According to the KDD Cup 2000 organizer's report by Kohavi et al. [2000], this preparatory work is compulsory, as the participants of the competition did not have access to domain experts to discuss open issues in view of the raw data. All preprocessing activities add up to innumerable hours spent for only a portion of the relevant preprocessing activities. Although the data warehouse should principally contain all available attributes and every single piece of information that may be useful for the pattern discovery phase of the WUA process, some of the available attributes were excluded from ETL modeling, if their available statistics show that they contain a vast majority of missing values or if they take the same value for almost all records.

⁸Compare section 3.1.3 on page 33.

- (3) **Building Loyalty.** As competitors are only a click away in EC, gaining customer loyalty is of central strategic importance in this domain. Recommender systems help to operationalize knowledge about customers and to create real relationships through personalized communications – a central aspect of ECRM⁹.

All three objectives are in line with the findings of section 2.2.2 on page 15, that is, recommender systems directly realize and support the mentioned five CRM activities, which in turn materialize a differentiation strategy for EC.¹⁰

Furthermore, recommender systems represent one feasible realization of the concept of mass customization, which has been identified as an important driver of EC in item 3 on page 10. Hence, the realization of a simple recommender system as a showcase of a typical WUA application for this chapter is an adequate choice to demonstrate WUSAN's capabilities and to identify open issues.

5.1.3 The Challenge of Real-Time Personalization

An EC Web site realized with an application server as discussed in section 3.2 on page 34 with sophisticated data collection capabilities is given. Naturally, application servers create every page delivered dynamically. To keep the page layout consistent, application servers employ a *template*, that is, a guideline that ascertains a consistent page design to a certain extent. The realization of an application server's templates depends highly on the concrete software product.¹¹

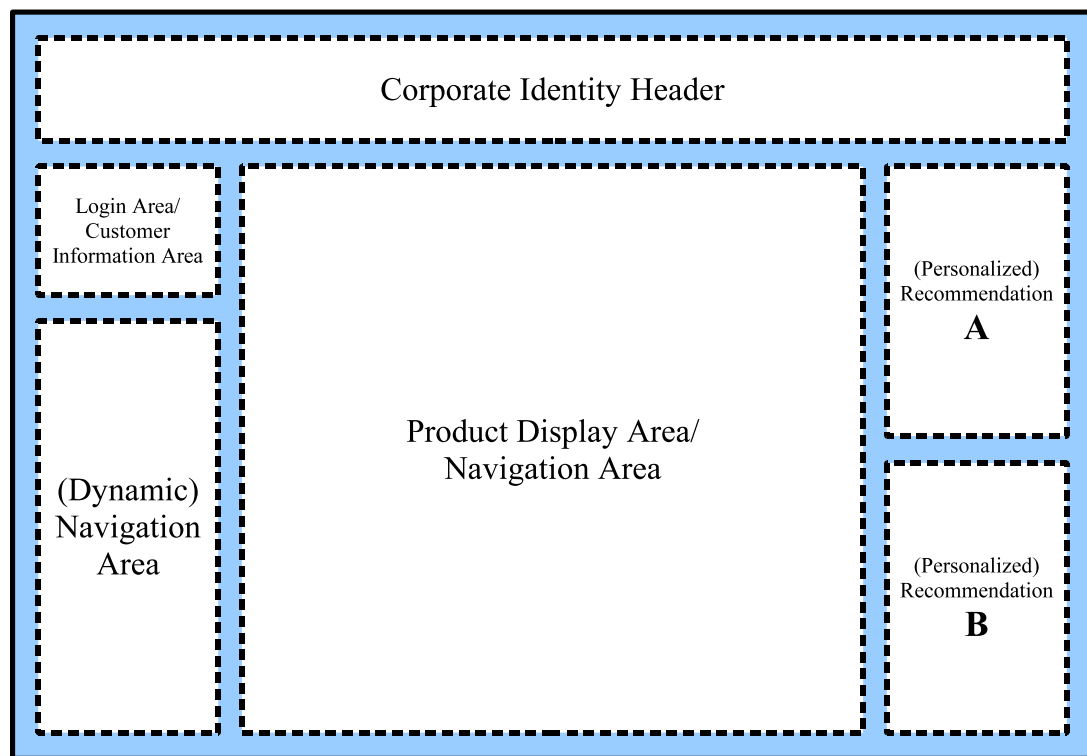


Figure 31: Template for a fictitious EC Web site.

⁹Compare definition 2.4 on page 20.

¹⁰Compare section 2.1.4 on page 11.

¹¹For instance, the Web content management system TYPO3 (see footnote 22 on page 38) offers the TypoScript template description language to create fixed or more flexible templates [Altmann et al., 2004, chapter 6].

Figure 31 on the previous page depicts the assumed template for the exemplary Web site. The template constitutes the basic page layout consisting of several placeholder areas, for example, a header containing the corporate logo or an area that contains a navigational menu. Whenever the application server creates a Web page, it fills in the placeholders with dynamic or fixed contents to create a complete Web page, which is then delivered to the user's browser.

In the context of the showcase of this chapter, the two placeholders on the right hand side of the template in figure 31 on the preceding page, recommendations **A** and **B**, must be filled in by the application server with personalized product recommendations for every single page delivered. To this end, the application server can embark on different strategies.

In a survey of recommender system examples provided by Schafer et al. [2001], the authors present numerous examples of feasible personalization strategies realized by established EC Web sites ranging from hard-wired recommendations based on less personal information, for instance, suggestions of best sellers, to personalized suggestions based on preference data such as interest information about products and services and purchase history.

Before any personalization activities can be tackled and deployed, *use cases* must be developed, since not all personalization methods are applicable at any point in time due to missing compulsory preconditions required for the application of the respective method. Consequently, to fill in the recommendation areas of the template in figure 31 on the previous page, the following case differentiation provides the basis for the the exemplary showcase personalization project of this chapter.

As stated before, the application server is supposed to fill in the recommendations for every page generated. The following case differentiation allows for different conceivable preconditions. As a basic rule, products a user has previously viewed in the same session or products that appear in a user's purchase history must not be recommended.¹² Furthermore, a product must not be recommended more than three times per session.¹³

Case 1: Anonymous User. This case refers to a user that is neither logged in nor identifiable with a cookie.

- (a) *No product view history from the current session is available.* If no product view history from the current session is available, only generalized recommendations can be made, for example, recommendations of best sellers (possibly allowing for additional constraints, such as time of day or weekday).
- (b) *A product view history is available.* If a product view history is available, personalized recommendations can be derived from the last n products viewed in the session. If no valid recommendations can be found with this approach, the strategy of the previous case is pursued.¹⁴

¹²In order to assure the first constraint, an advantage of application servers mentioned in section 3.2.2 on page 37, their sophisticated session handling, can be leveraged. According to Hall and Brown [2003, chapter 9], the session handling mechanism creates a session object for each session initiated on the application server. Apart from basic session information such as number of requests or assigned session ID, more complicated information can be stored within a session object, for instance, a record of viewed products. Commercial application servers normally allow for smooth and automated updates of such additional information. Consequently, the mentioned constraint can be accounted for with relatively little effort.

¹³This is an arbitrary restriction. However, common sense suggests that users are being annoyed by recommendations recurring too often. On the other hand, if a user does not click on a product recommendation even if proposed several times, this can be regarded as an implicit user feedback expressing the user's current lack of interest in the recommended product.

¹⁴Given that there exists a large number of products for sale and that a user has not viewed all products yet, case item 1a leads to valid recommendations at any time.

Case 2: **Known User.** This case refers to a user that has been identified either through his log-in or through a cookie.

- (a) *No individual purchase history is available.* After a user has been assigned to a certain cluster, the approaches of case item 1 on the facing page are pursued, restricted to the user's cluster instead of the complete customer base.
- (b) *An individual purchase history is available.* Based on the user's past n purchases, the recommendations are derived from precalculated association rules. Depending on the amount of available purchase records, the association rules can be computed for each cluster of customers or for the entire customer base. If no valid suggestions can be found, recommendations are chosen following the previous case.

5.1.4 Requirements for the Data Pool and Methodical Requirements

At this point, the question arises as to how the use cases of the previous section can be realized concretely. A quick look at section 3.1.3 on page 33 pinpoints four alternative approaches for Web personalization. Sarwar et al. [2002] state that collaborative filtering is one of the earliest and most successful techniques for Web personalization.¹⁵ However, as stated in item 2 on page 34, collaborative filtering is typically based on *explicit* user feedback rather than behavioral data (*implicit* user feedback). This means that the application server must provide the infrastructure to store and process explicit user feedback. As the KDD Cup 2000 data neither contain explicit user ratings nor information suitable for the content-based filtering alternative, the showcase is restricted to manual decision rules and WUA personalization approaches, the first and fourth alternative of personalization mentioned in section 3.1.3 on page 33.

5.1.4.1 Requirements for Anonymous Users

In case of item 1a on the facing page (anonymous user without product view history), a list of product IDs is required sorted in descending order according to the number of sales in a predefined period, for example, the past seven days. This list can be computed once a day and made available in a way that realizes efficient access to the list. For anonymous users starting a new session, the top two products are employed as recommendations for slots **A** and **B** in figure 31 on page 97. The recommendations can be refined by generating separate top selling lists for different times of day. Top selling lists can be created with simple OLAP queries given that an order mart is available containing all purchased items of the desired period.

In case of item 1b on the facing page (anonymous user with product view history), different approaches are feasible. This case is the most challenging of all, since no recommendations can be prepared in advance for anonymous users and the decision about which recommendations are displayed next must be made in real-time, as the product view history, which is supposed to be the basis for recommendations, arises in real-time, too. The approach pursued for this case relies on a set of pre-computed *frequent item sets*¹⁶.

Following Mobasher et al. [2001], the recommendation engine takes a pre-computed collection of frequent item sets and generates recommendations by matching a user's current product view history against the item sets. The application server must be configured to charge the

¹⁵Herlocker et al. [2004] describe recommender systems based on collaborative filtering in great detail.

¹⁶Item sets for association rules have been introduced in item 2ii on page 28. The computation of frequent item sets is a prerequisite for most association rules algorithms [Witten and Frank, 2005, section 4.5]. They refer to items that frequently occur together in transactions.

recommendation engine with the last n products viewed¹⁷ in a user's current session before the next page is delivered to the user's browser. Given the last n product views, all item sets of size $n + 1$ satisfying a specified support threshold¹⁸ and containing the last n product views are considered. As Mobasher et al. [2001] state, the recommendation value of each candidate product is based on the confidence¹⁹ of the association rule $A \Rightarrow B$ where A is the set of the most recent n product views and B is the recommendation candidate. To realize this procedure, the following subproblems must be tackled:

- (1) Choose an appropriate value for n and an appropriate support threshold.²⁰ The computation of frequent item sets is the first step of association rules algorithms [Witten and Frank, 2005, section 4.5].²¹
- (2) Compute frequent item sets on a regular basis for a predefined period based on the product views of all sessions during this period. This can be done with one of XELOPES's high-performance association rules algorithms. To this end, product views must be extracted from a clickstream data mart conforming to transactional data format [see Thess and Bolotnicov, 2004, section 6.6.4].²²
- (3) Store computed frequent item sets in a data structure facilitating efficient access to the frequent item sets required by the recommendation engine. Mobasher et al. [2001] propose a *frequent item set graph*.²³

Wrapping up the case of anonymous users, the following tasks must be accomplished with WUSAN as a premise to deploy the recommendation engine for anonymous users: (i) create an order data mart, (ii) create a clickstream data mart. Both tasks are discussed in more detail in section 5.2.3 on page 104.

5.1.4.2 Requirements for Known Users

If the the customer base is large, it can be segmented into clusters of customers so as to focus personalization efforts on each cluster instead of the entire customer base. Meaningful customer segments are of central strategic importance for a comprehensive CRM strategy and hence, customer segmentation is not part of core WUA activities. Rather, customer segmentation, that is, the creation and maintenance of meaningful customer segments, is a separate task, the results of which can be leveraged for WUA.²⁴

¹⁷More precisely, the last n *distinct* product views.

¹⁸Compare equation (3.3) on page 29.

¹⁹Compare equation (3.5) on page 29.

²⁰The higher both values, the less recommendations can be derived. Both parameters must be set to values that result in a fair amount of frequent item sets and candidate association rules.

²¹Any subset of a frequent item set of length n must be a frequent item set itself. Hence, frequent item sets are computed *gradually* starting with item sets of length 1. These item sets are then composed into candidate item sets of length 2. Then, all valid frequent item sets of length 2 are identified and so on.

²²In terms of market basket analysis (compare item 2ii on page 28), sessions can be regarded as transactions containing product views.

²³A frequent item set graph is a directed acyclic graph. Level 0 contains a root node, level 1 all frequent item sets of length 1 in lexicographic order and so on. Each frequent item set is sorted in lexicographic order, too. Given n product views in lexicographic order, it must be checked if they are contained as a frequent item set on level n . If so, iterate through all connected nodes. These nodes contain all candidate frequent item sets that can be employed to derive association rules having the last n product views as a premise and a single product as a consequence. If no proper rules can be found, iterate with last $n - 1$ product views.

²⁴Customer segmentation can be addressed with various data mining methods. Kelly [2003] provides an introduction to customer segmentation with data mining.

In case of item 2a on page 99 (known user without a purchase history), the procedure is the same as described in the previous section, that is, a user's product view history is employed to generate personalized recommendations. However, in item 2 on the preceding page, frequent item sets are computed separately for each customer cluster, that is, customers from different clusters with identical product view histories may be provided with different recommendations.

In case of item 2b on page 99 (an individual purchase history is available), the individual purchase history is taken as input for personalized recommendations rather than the product view history of the current session. While recommendations that are based on the current session aim at motivating users to view products being popular among customers of their assigned cluster (making them stick to the Web site longer), recommendations that are based on individual purchases clearly aim at motivating users to buy the recommended products.

The subproblems to be tackled for this case are quite similar to the three subproblems discussed in the previous section. However, frequent item sets are computed from actual purchases, that is, products purchased by customers of the same cluster in a predefined period. Consequently, apart from frequent item set graphs covering product views during sessions, frequent item set graphs of actual purchases must be computed for each customer cluster and the entire customer base as a fall-back position (required if training data per cluster are too few to derive meaningful frequent item sets).

In contrast to anonymous users, where the current session behavior must be evaluated before any personalized recommendations can be derived, suggestions based on purchase histories provide an opportunity to pre-compute personalized recommendations for every single customer on a regular basis. These recommendations can then be accessed conveniently and incorporated into any delivered page as soon as a user has been identified. This approach reduces overhead required for the computation of recommendations in real-time.

As purchase histories are available in an order data mart, no more data marts must be created for the recommendation engine than the two data marts mentioned in the previous section: (i) an order data mart and (ii) a clickstream data mart. Nevertheless, both data marts imply a number of other data marts and standardized dimension tables as discussed in the following section.

5.2 Creating a Closed Loop

This section investigates the tasks and steps required to create a closed loop for the WUA process of the recommendation engine showcase. Figure 32 on the next page depicts the mappings that must be set up and configured to automate the WUA process for the recommendation engine in terms of an *automated data flow*.

5.2.1 Required Mappings

As the recommendation engine showcase points up, the data flow in figure 32 on the following page must be traversed on a regular basis, since the data mining models involved into the recommendation process are built from data that are obsolete in no time. On large EC Web sites, thousands of users are served every day in thousands of sessions, which accumulate large volumes of behavioral and transactional data both of which are relevant for the recommendation strategy outlined in section 5.1.3 on page 97. In order to provide accurate personalized recommendations, it is necessary to capture current customer flavors and preferences and to map them promptly to the data mining models of the recommendation engine.²⁵

²⁵Automating the WUA process has been identified as an important prerequisite for efficient WUA in section 2.2.4 on page 20 and section 3.3 on page 41.

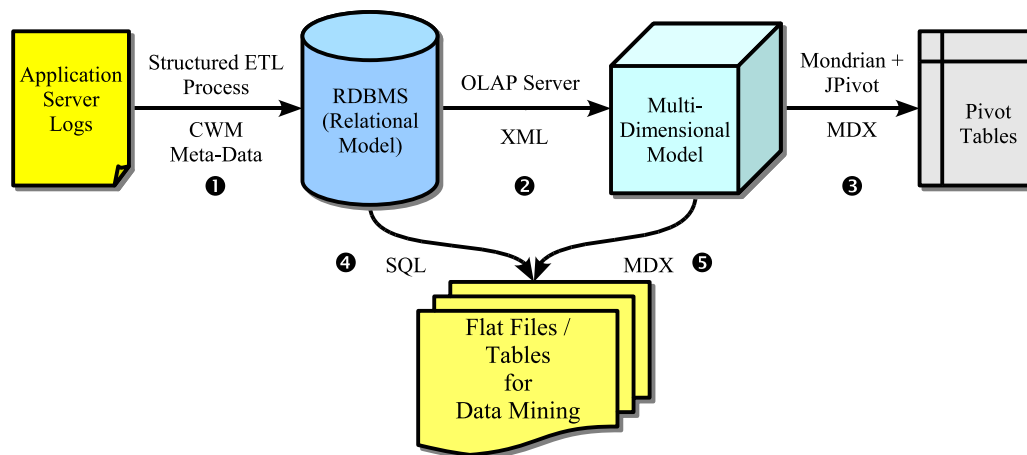


Figure 32: Mappings related to the data flow for the recommendation engine.

In detail, the mappings in figure 32 comprise the following tasks:

- Mapping 1: This mapping is the most complex, since it presupposes conceptual preparatory work to map application server logs to a relational data model.²⁶ If the conceptual work is completed, the data must be mapped to comply with the conceived concepts. And that is the point where WUSAN and the LOORDSM come into play: they support efficient and intuitive modeling of the ETL process.
- Mapping 2: This mapping maps the relational data model to a multi-dimensional data model, which is suitable for OLAP analyses. As discussed in item 1 on page 54, this mapping is done by the ROLAP engine, which according to figure 15 on page 53 is a constituent of WUSAN. In case of the Mondrian ROLAP engine, this mapping must be provided through an XML schema file. Schema files are discussed in the appendix section G.4 on page 172.
- Mapping 3: This mapping is done with MDX. As described in item 2 on page 55, JPivot acts as a Mondrian client and can be regarded as a graphical MDX query builder. Assembled MDX queries are then sent to the OLAP engine, which translates them to SQL queries based on the XML schema file mentioned in connection with the previous mapping.
- Mapping 4: According to Witten and Frank [2005, section 2.2], the input to data mining algorithms is generally expressed as a table of independent instances of the concept to be learned. Consequently, data from the data warehouse must be denormalized prior to applying data mining algorithms to it.²⁷ SQL queries always produce relations suitable as input for data mining algorithms employing the database streams discussed in item 3 on page 71.

²⁶Conceptual work primarily refers to the actual design of relational schemas within the data warehouse. See remark on page 54.

²⁷At this point, performance issues play a crucial role. Ramakrishnan and Gehrke [2003, chapter 3] state that denormalization may improve performance for queries to an RDBMS involving attributes from several previously decomposed relations. The decisions to what extent relations should be stored in a denormalized way must be taken at the conceptual level and cannot be simply passed on to the optimizers of an RDBMS. As discussed in section 5.3 on page 109, the Mondrian OLAP server offers features to incorporate denormalized relations. However, performance tuning generally is a manual task with little automation, yet it must be addressed if the WUA process is to be deployed as a closed loop.

Mapping 5: This mapping refers to tapping the data warehouse for data mining with multi-dimensional queries. Since the multi-dimensional data model of the data warehouse cannot be forthrightly employed for data mining, relations must be derived from the multi-dimensional data model. This can be achieved in two ways: (a) by accessing underlying non-aggregated data with drill-throughs (discussed in section G.3.2 on page 171) and (b) by mining two-dimensional pivot tables, which represent a two-dimensional extract of the aggregated data from a data mart. The latter approach must be handled with care, since pivot tables may contain rows with aggregated data from different hierarchies, that is, different rows represent different concepts. As mentioned before, data sets for data mining must contain instances of the *same* concept – the one to be learned.²⁸

5.2.2 Essential Tasks for Creating and Populating a Data Mart

This section focuses on the first mapping on the preceding page, that is, the population of the data warehouse with transformed application server logs. Figure 26 on page 83 depicts the raw data flow for this task. Although this illustration may suggest that this mapping can be accomplished in a straightforward manner, the following list enumerates all *conceptual* and *technical* tasks required to populate the data warehouse for the recommendation engine:

- (i) Determine which attributes are required for intended analyses, that is, attributes for populating the order and clickstream data marts (conceptual task).
- (ii) Since not all required attributes can be tracked directly, it is necessary to figure out how needed attributes can be derived from those attributes that can be logged by the application server (conceptual task).
- (iii) If necessary, adjust the tracking mechanism of the application server to make it capture all missing attributes (technical task).²⁹
- (iv) Determine relational structure of the data marts to be populated (conceptual task).
- (v) Create meta-data for the database tables of the previous task and create physical database tables thereof (technical task). Modeling the dimensions involved in the data warehouse for the recommendation engine is discussed in the appendix section G.1 on page 149.
- (vi) Model ETL transformations and their execution order with the LOORDSM through WUSAN's WusanML interface (technical task).
- (vii) Create source streams from application server logs and execute ETL process with WUSAN's framework (technical task).

²⁸Tapping the pivot tables that have been created with JPivot has not yet been implemented in WUSAN. Nevertheless, this feature could be realized in principle with Mondrian's API. However, a stream class deploying this feature would have to assure that only rows from the same hierarchical (conceptual) level are accessed as discussed before. Realizing this constraint is quite complex, since pivot tables may contain rows from various hierarchy levels. On the other hand, users employing such a stream would have to make sure that mining this stream is reasonable from a semantic perspective.

²⁹This task may turn out to be very laborious. According to section 3.2.2 on page 37, application servers principally provide customized logging capabilities. However, the actual complexity of customizations heavily depends on the concrete application server system's capabilities, especially its ease of use in view of customizations.

Basically, above tasks must be traversed one time given that sufficient conceptual preparatory work has been accomplished. However, in practice, the tasks are tackled *iteratively* with jumps back and forth as described for the WUA process in the remark on page 32.³⁰

Conceptual aspects such as systemizing what attributes must be logged for certain analytical objectives and how to best map these attributes to data marts conceptually rather point at open research issues than simple recipes that can be derived easily and deployed in a standardized way. Modeling data marts for WUA is the primary conceptual task to be accomplished in above enumeration. It is considered in detail in Sweiger et al. [2002, chapter 6] and Kimball and Merz [2000, chapter 6]. However, the actual feasibility of the proposed designs strongly depends on the available data sources. Consequently, the KDD Cup 2000 data [KDDCUP-DATA] strongly designate the data marts in the course of the following discussion.

5.2.3 Creating the Data Warehouse with WUSAN

This section ties in with item vi on the previous page and item vii on the preceding page of the previous section, that is, the concrete realizations of the two data marts required for the recommendation engine are discussed. While the data reside in the data storage layer in figure 16 on page 60, the ETL modeling is accomplished in the ETL layer with the LOORDSM. The modeling in this section is illustrated with WusanML, which represents WUSAN's user interface for ETL modeling.³¹

Similarly to section 4.3 on page 82, for this section, it is assumed that the reader is familiar with dimensional modeling and the associated terms *dimension table* and *data mart*, each of which refers to the data storage layer in figure 16 on page 60.³²

5.2.3.1 Order Mart

A *data mart* is a logical and physical subset of a data warehouse [Kimball and Merz, 2000, p. 362]. As for the physical subset, the term refers to a collection of database tables that are linked through foreign key references. Figure 33 on the next page illustrates the actual foreign key references of the order mart in the RDBMS (the database tables that are referenced by the referenced fact tables are not displayed).

An order consists of one or more order lines, each of which covers a single product [Kohavi et al., 2000]. The solid lines imply that the referenced tables participate in the ETL process initiated by an instance of the `StarSchema` class with the fact table `ORDER_FACT`. Unlike the solid lines, the dashed lines imply that the referenced tables do not participate in the ETL process initiated by the `ORDER_FACT` table and must be populated in a separate ETL process. Hence, the dashed lines represent ETL modeling according to the gradual ETL approach in item 2 on page 91 (and the solid lines represent ETL modeling following the counterpart instantaneous ETL approach).

Tables referenced by the `ORDER_FACT` table in figure 33 refer to dimension tables or

³⁰A common reason for additional cycles are changing analytical requirements over time. These involve additional required attributes or changed attributes, leading to modified database tables. However, such modifications of relational tables can only be accomplished inasmuch as the underlying RDBMS supports them.

³¹Actual WusanML and PMML models are displayed in the appendix chapter G on page 149.

³²As depicted in figure 16 on page 60, the OLAP layer operates on the data storage layer, that is, the dimension tables and data marts in the RDBMS. The ETL layer "merely" creates and populates the database tables required for the data warehouse. Although there exist strong relationships between the notions defined in section 4.3 on page 82 and similar notions of dimensional modeling, it is important to emphasize that the former belong to the ETL layer and the latter to the data storage layer. A one-to-one correspondence does not exist!

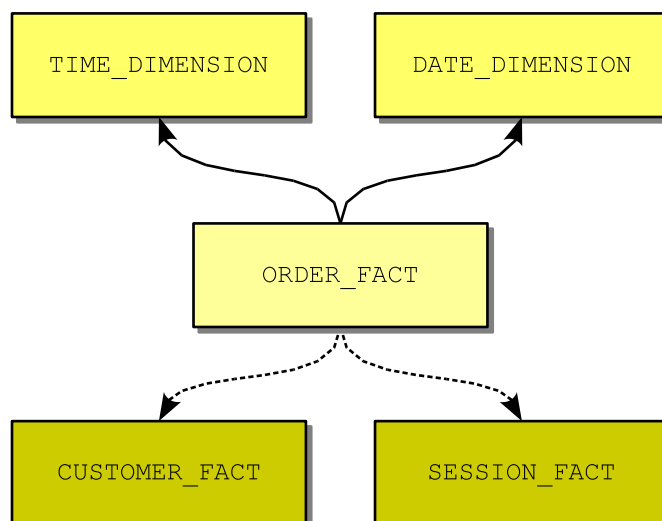


Figure 33: Foreign key references of the order mart.

fact tables, each of which is discussed in more detail in appendix section G.1 on page 149 and appendix section G.2 on page 160. The actual modeling to be accomplished by users setting up the recommendation engine consists in modeling with WusanML intuitively and in a straightforward manner the star schemas and ETL transformations of the data marts, depicted in detail in appendix section G.2.2 on page 162.

Remark (Modeling through a GUI). WUSAN’s consequential employment of XML interfaces for modeling the ETL process can be regarded as a perfect jumping-off point to create a GUI that can be leveraged to support users in modeling the ETL process. The GUI would support users with all conceptual tasks of section 5.2.2 on page 103 and automatically create XML models as starting points for the technical tasks, which can be executed automatically based on the XML models (see figure 34). In this scenario, users would employ the GUI to create the WusanML and PMML models referenced above. Since a GUI does not directly contribute any new insights into ETL modeling (as it mainly contributes to adopting the LOORDSM with less efforts), its realization has been omitted in this thesis.

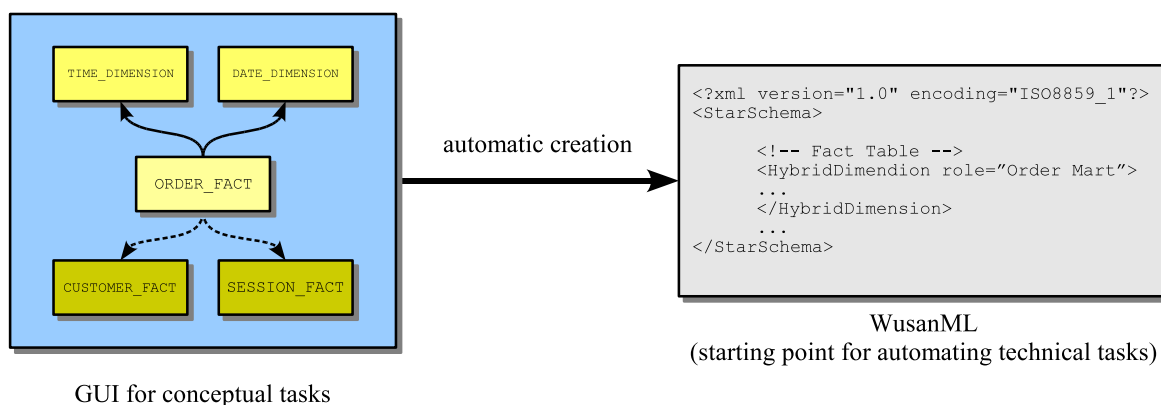


Figure 34: Modeling through a GUI.

This remark aims at outlining the functionalities and benefits of a GUI for modeling the ETL process. Basically, the GUI assists users in modeling the data flow depicted in figure 26 on page 83, that is,

- (i) it supports setting up streams (sources of the ETL process),
- (ii) it supports modeling the data marts (targets of the ETL process), and
- (iii) it supports modeling ETL transformations to transform the source streams into target streams, thereby populating the data marts.

Modeling the ETL process with the GUI could proceed as follows: (a) Create model of target data mart by graphically combining fact and dimension tables, resulting in a diagram similar to figure 33 on the preceding page. Connect the fact table to the dimensions either by solid or dashed lines.³³ (b) Click on each dimension and compose its meta-data. Following figure 28 on page 88, role meta-data can be created automatically for each dimension and added to the meta-data of the fact table. The modeled meta-data conform to figure 17 on page 65 assuring CWM compatibility. (c) Add attributes to the meta-data of the fact table if degenerated attributes are required. Attributes of foreign keys are added automatically. (d) Click on each dimension (including degenerate dimensions) and compose ETL transformations for that very dimension. Users are offered a library of WUA-specific transformations that can be composed to more complex transformations as discussed in chapter 4 on page 59. (e) Iterate as necessary.

This approach results in two types of XML descriptions. First, PMML listings describing dimensions such as those of section G.1 on page 149 and second, WusanML listings such as those of section G.2 on page 160 describing a star schema and its assigned ETL process. The former are employed to create or modify the physical database tables (retaining CWM compatibility) and the latter are employed to actually execute the ETL process. All transformations contained in the latter listings fully comply with the theoretical model discussed in detail in chapter 4 on page 59.

5.2.3.2 *Order Line Mart*

The order line mart is modeled similarly to the order mart. It centers the line items that contribute to an order [compare Kohavi et al., 2000]. As before, figure 35 on the facing page depicts the foreign key references of the fact table `ORDER_LINE_FACT`, which indicate that, as before, both ETL approaches – instantaneous and gradual – are employed. The corresponding WusanML model is illustrated in listing G.14 on page 164.

5.2.3.3 *Clickstream and Session Marts*

Although the clickstream and session marts are modeled separately, they must be discussed in conjunction, since the clickstream mart references the session mart through the instantaneous ETL approach of item 1 on page 91. Using the same semantics as before, figure 36 on the next page depicts the foreign key references of the `SESSION_FACT` table of the session mart. The

³³Within the LOORDSM of figure 27 on page 87, the ETL process is handled by the `StarSchema` class, which invokes all ETL sub-processes of tables referenced directly (that is through solid lines in the diagrams). If the referenced table is a fact table, the ETL process of the `StarSchema` class to which the fact table has been assigned is invoked recursively. Note that a physical database table acting as a fact table can be embedded in only one instance of the `HybridDimension` class and this instance can again be embedded in only one instance of the `StarSchema` class. That is, the mentioned recursive invocation of an ETL sub-process is unambiguous.

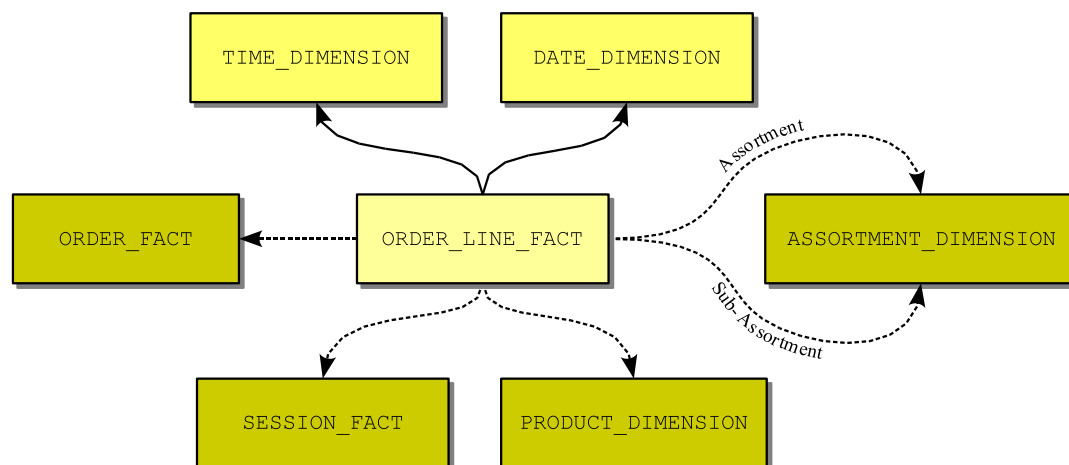


Figure 35: Foreign key references of the order line mart.

corresponding WusanML model is depicted in listing G.15 on page 166 and is discussed in more detail in the appendix section G.2.4 on page 166.

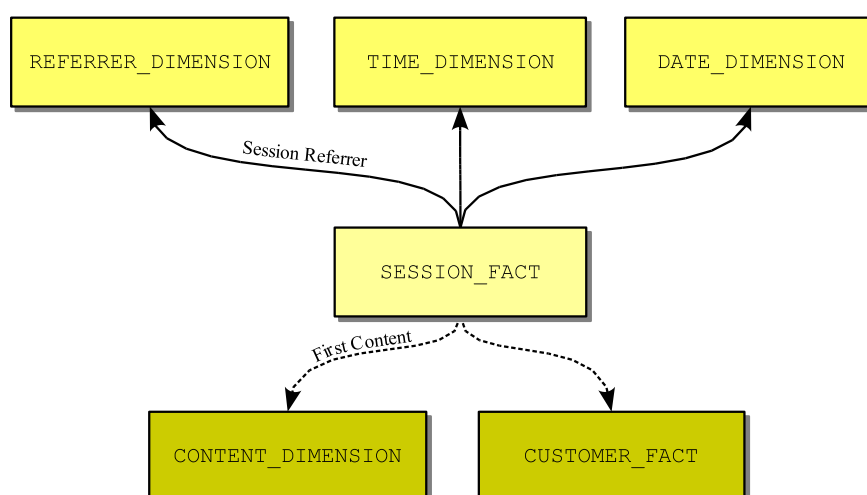


Figure 36: Foreign key references of the session mart.

Finally, figure 37 on the following page illustrates the last data mart for the WUSAN data warehouse, the clickstream mart. As the solid line from the CLICKSTREAM_FACT table to the SESSION_FACT table indicates, the session mart is linked directly, pursuing the instantaneous ETL approach, that is, both data marts are populated *simultaneously*. The corresponding WusanML model is illustrated in listing G.16 on page 168.

Remark. At this point, mapping 1 of section 5.2.1 on page 101 has been tackled and can be established through WUSAN's WusanML interface as discussed in the previous sections and in the referenced appendix sections. Mapping 2 can be created through another XML interface, which is implemented by Mondrian. The details are tedious routine and are discussed in appendix section G.4 on page 172. Mapping 3 is managed by JPivot and mapping 4 is covered by the database stream classes. Mapping 5 is discussed in the following section.

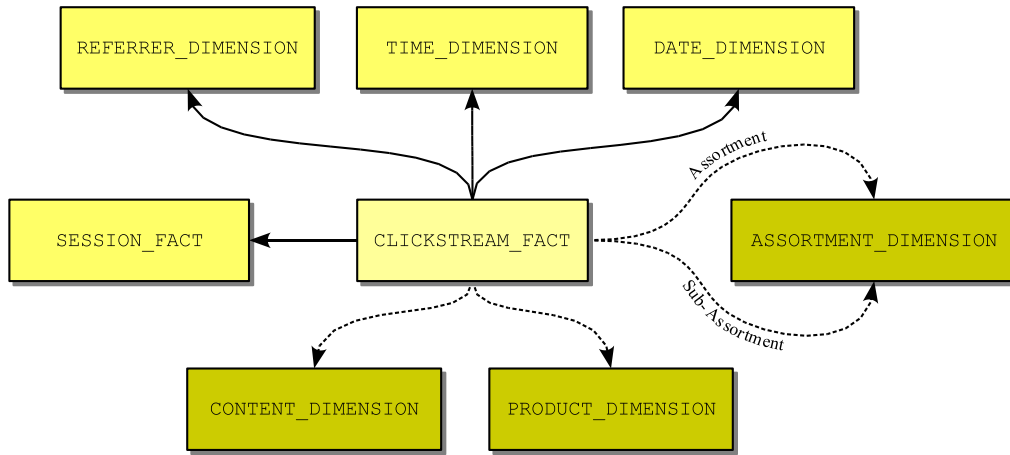


Figure 37: Foreign key references of the clickstream mart.

5.2.4 Automating the Recommendation Engine

Given the data warehouse modeled and populated with the models and utilities of the previous section and the referenced appendix sections, it is now time to recapitulate the tasks required to regularly configure and adjust the recommendation engine. Those tasks must be automated so as to minimize any manual interactions. It is important to mention that – once the conceptual tasks of section 5.2.2 on page 103 are completed – the ETL process can be automated through its WusanML and PMML models, that is, the data pool exploited by the recommendation engine can be updated automatically. This is a crucial prerequisite for automated updates of the models and data structures from which the recommendation engine is fed. Consequently, the ETL process can be executed on a regular basis, for example, in daily or weekly periods feeding the data warehouse with the latest behavioral and transactional data.

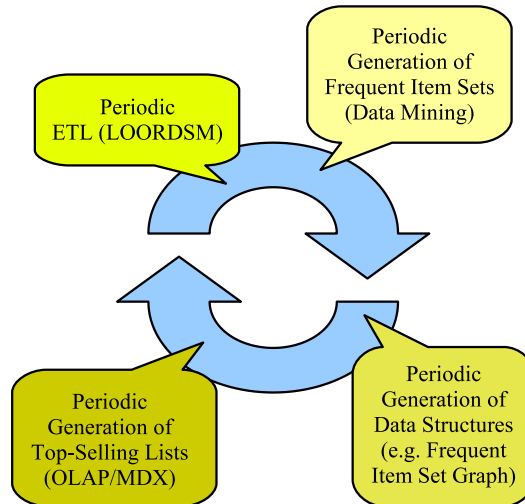


Figure 38: Tasks for the recommendation engine that can be automated.

In section 5.1.4.1 on page 99 and section 5.1.4.2 on page 100 the recommendation strategies for anonymous and known users have been outlined in detail. Those strategies involved various pre-computations of data mining models and data structures to efficiently access these models. While the computations of the data mining models can be easily automated with XE-LOPES’s interface for invoking data mining algorithms given that all required parameters are

known, determining those parameters, for instance, the support threshold mentioned in item 1 on page 100, poses a great challenge. Appropriate values can only be determined through experiments that somehow evaluate the perceived recommendation quality for a given support threshold. This is an open issue involving laborious one-time efforts. However, as stated before, once the parameters are known automating periodic model generations is merely a technical task. MDX queries on the other hand, such as those required to create the top-selling lists mentioned in section 5.1.4.1 on page 99, can be easily automated through Mondrian's API.

Figure 38 on the preceding page summarizes the very tasks to be automated for the recommendation engine. The LOORDSM significantly contributes to closing this loop as its structured formal model on the one hand, supports the transition from conceptual tasks to concrete technical tasks (compare remark on page 105 and section 5.2.2 on page 103) and on the other hand, supports its automated execution.

5.3 Performance Issues

WUSAN-specific performance issues and general data warehousing-specific performance issues may not be neglected when deploying the recommendation engine. Poor system performance may bungle the automated closed loop discussed in the previous section due to frequent inevitable manual interventions. A smooth high-performance system is compulsory for any practical projects. WUSAN, however, is still in the state of a prototype and hence, a general proof of feasibility has been more important than ease of use and in-depth performance tuning. This section aims at briefly outlining the endless field of performance tuning for WUSAN and data warehousing in general.

Two performance issues are particularly important for WUSAN. First, an efficient ETL process with high data throughput is a critical factor for closing the loop of the WUA process, as large volumes of data must be preprocessed on a regular basis (compare item 1 on page 42). Second, the data warehouse, or, more precisely, the underlying RDBMS, must be configured for high performance in view of the queries issued by Mondrian OLAP. Both aspects are a science in their own right and cannot be covered in detail in this thesis.³⁴ Nevertheless, even for a prototype, performance issues must be accounted for so as to obtain a feasible system.

5.3.1 Implementational Aspects

One key performance aspect has been realized in the Java classes of the LOORDSM, as discussed in item 1 on page 89: it was sketched how to minimize reading and writing database accesses through caches within the `MiningTableSqlStream` class extensively used during the ETL process. Another major performance aspect inherent to the LOORDSM is transformation modeling (compare section 4.2.3 on page 72). Although the XELOPES library carves out the possibilities of modeling transformations, the transformation atoms of section 4.2.3 on page 72, the `OneToOneMapping`, `OneToManyMapping`, and `MultipleToMultipleMapping` classes, should make use of caches and avoid unnecessary computations, if feasible. Furthermore, redundant mappings should be avoided, for example, unneeded attribute realignments or nonessential duplicate mappings.

Further performance issues concern the underlying RDBMS of the data storage layer [see Kimball and Caserta, 2004, chapter 8]:

³⁴Wood [2005] discusses three performance issues for Mondrian OLAP: (i) a generalized tuning process for Mondrian, (ii) recommendations for database tuning, and (iii) *aggregate tables*. The last cited are an important aspect for performance tuning (see section 5.3.2 on the following page).

- (i) **Effect of Indexes.** Extensive employment of indexes³⁵ in an RDBMS significantly improves the query performance of the data warehouse. Hence, a rather aggressive index strategy should be pursued in order to accelerate the data warehouse for analysis. Yet, during ETL, indexes significantly thwart the overall process due to the overhead required to maintain them on each vector insertion. Hence, it is recommended to drop all critical indexes prior to executing the ETL process. After completion, they should be reinstalled in order to provide for a satisfactory query performance during analysis.
- (ii) **Foreign Key Constraints.** Foreign key constraints for columns of a database table that reference the primary key of another database table help to ensure referential integrity within the data storage model. Adding constraints such as cascading deletions to foreign key references may significantly simplify the maintenance of the data model, for instance, if unwanted and depending vectors are to be removed efficiently. As with indexes, foreign key constraints add significant overhead to vector insertions, since their constraints must be checked on every insertion. Thus, as before, critical foreign key references must be dropped prior to executing the ETL process.³⁶

Remark. The `PrimaryKeyGenerator` class discussed in item 1 on page 89 adds to the computational complexity during the ETL process. That is why the computationally expensive MD5 primary key mapping discussed in footnote 51 on page 89 should be replaced with a less computationally expensive mapping, if feasible. For example, as shown in the XML listings of appendix G referenced in the previous section, the `TrivialPrimaryKey` class can be employed for database tables that include a key candidate attribute as first attribute.

5.3.2 Aggregate Tables

In footnote 34 on the previous page, it was mentioned that aggregate tables can be leveraged for tuning Mondrian's performance. Some of the fact tables derived from the KDD Cup 2000 data represent *aggregate tables*, for instance, the `SESSION_FACT` table is an aggregate table of the `CLICKSTREAM_FACT` table. Emberson [2005] states that unlike many OLAP servers, Mondrian does not store data on the hard disk: it works exclusively on the data in the RDBMS combined with a cache. In order to accelerate measure computations, an aggregate table that coexists with the base fact table and contains pre-aggregated measures built from the fact table can be added to a cube by means of the schema file. Mondrian then decides whether to access the fact table or one of the aggregate tables when computing an MDX query.

According to Emberson [2005], Mondrian will support two aggregation techniques in forthcoming versions, "*lost dimension*" and "*collapsed dimension*". The former means that the fact table is aggregated across all values of the lost dimension, and the latter means that the aggregate table is obtained by inserting the dimension into the fact table. The author concludes that making the decision as to which aggregate tables to build is analogous to choosing which indexes to build on a table: it is application dependent and calls for considerable experience.

It can be concluded that tuning the data warehouse in view of WUA-, EC-, and ECRM-specific measures and analyses still necessitates significant research efforts. Not only must

³⁵An introduction to indexes is given in Ramakrishnan and Gehrke [2003, chapter 8].

³⁶Although the filtering mechanism of the ETL process employed in section 5.2.3.3 on page 106 behaves similarly to cascading deletions, it does not actually set up foreign key constraints in the RDBMS. Neither the gradual nor instantaneous ETL approach automatically implements foreign key references in the RDBMS. Therefore, checking referential integrity after completing the ETL process by manually creating the foreign key constraints in the RDBMS for foreign key columns is highly recommended.

useful measures be identified, but managing performance tuning (for instance, dropping and creating indexes, creating aggregate tables, and adding deltas to them) must also be addressed so as to develop domain-specific procedures and best practices. WUSAN and especially the LOORDSM can be regarded as enablers for future research activities in WUA, EC, and ECRM, as they significantly lower the preprocessing hurdle – an indispensable step prior to any analysis activities – as yet an impediment for further research interactions.

5.4 Summary

Based on a realistic recommendation engine showcase, it was demonstrated in this chapter and the referenced appendix chapter G on page 149 as to how the LOORDSM – introduced in chapter 4 as a structured model – can be deployed concretely by modeling ETL for the KDD Cup 2000 data. Apart from the benefits summarized in section 4.4 on page 92, the following benefits of the LOORDSM became apparent in this chapter:

- (1) The LOORDSM significantly simplifies the overall WUA process by easing modeling ETL, a sub-process of the WUA process. The WUA process has been presented as a response to the problem discussed in chapter 2: as to how EC Web sites respond to transforming market conditions in the Web channel. Consequently, as WUSAN and the LOORDSM both improve the WUA process, the initial challenge of winning over the customer in the Web channel can be tackled more efficiently.
- (2) The LOORDSM with its clearly structured mathematical data and transformation model greatly fosters the creation of an automated closed loop in concrete applications such as a recommendation engine.
- (3) As the LOORDSM leverages the CWM's meta-data model, arbitrary log formats can be mapped to WUSAN's data warehouse following the same structured approach.
- (4) XML interfaces make it possible to simplify the set up of the ETL process by supporting conceptual tasks graphically, that is, all required XML models can be assembled semi-automatically.
- (5) WUSAN and especially the LOORDSM can be regarded as enablers for future research activities in WUA, EC, and ECRM, as they significantly lower the preprocessing hurdle – an indispensable step prior to any analysis activities – as yet an impediment for further research interactions.

The LOORDSM was exposed as the key component of WUSAN. It represents the fundamental prerequisite for creating a powerful analytical system, since it significantly simplifies the preprocessing phase. As shown in the previous chapters, it is this inevitable step that is neglected by existing research approaches. In this connection, it is compulsory to recall the discussion in the introduction of section 3.3 on page 41, which stated that, unlike other application domains of data mining, WUA deals with large volumes of data on a regular basis, hence demanding automated preprocessing with minimal manual interactions. Furthermore, it must be emphasized that the LOORDSM cannot compensate any weaknesses in data collection, that is, its benefits strongly depend on the data collection issues discussed in section 3.2 on page 34.

Chapter VI

SUMMARY

This summary provides both a look back at the issues covered in this thesis and a look ahead to the consequences the results may have as well as the opportunities and challenges for further research. Many aspects have already been discussed in the conclusions of each chapter, yet some issues arise only from a comprehensive perspective.

Chapter 2 picked up the problem posed in the introduction on page 1: How can organizations make advances in winning over the customer in markets where the power is radically shifting from sellers to buyers? The Web channel has been identified as the primary battleground for this challenge, and the retail sector has been considered as the sector that is particularly affected by this development due to the co-action of various aspects such as technological factors, globalization, and human factors.

It was argued that in this environment, a cost leadership strategy is insufficient and a combination of several strategic approaches is required with an emphasis on *differentiation*. At this point, CRM came into play as a well-established discipline delineating the options for differentiation in terms of proactively shaping and cultivating customer relationships. Many organizations embrace CRM as a strategic response to implement differentiation in changing markets and transfer its concepts to the Web channel. It is ECRM that is perceived as being of paramount importance in tackling the challenge of winning over the customer in the Web channel, since the WWW offers unparalleled options to collect behavioral data about customers and prospects. These data are supposed to be of great value for nurturing customer relationships. Figure 6 on page 23 outlined a high-level framework for ECRM, which represents the overall concept pursued in this dissertation.

Immediately, the question arose as to how the ECRM framework can be deployed and shaped concretely. To this end, chapter 3 on page 25 provided the foundations of the Web channel. From the technical and research perspectives, it was discussed what analyses can be done, how data can be best collected, and which general Web site architecture is favorable for WUA, the analysis technique applicable for the Web channel. *Data warehousing* was introduced as a best practice to provide a flexible data repository with high data quality that can be employed for a wide range of analytical tasks. But how must the volumes of data be transformed and pre-processed, and how can the data warehouse be populated? Although some research efforts and practical approaches exist, this question has been identified as a key issue, yet to be answered, since the volumes of data available for WUA do not tolerate extensive manual interventions during the preprocessing phase.

At this point, WUSAN was introduced, a framework for effective WUA, which broadly follows the ECRM framework in figure 6 on page 23 and the general architecture for Web usage mining in figure 11 on page 43. After a thorough analysis of the strengths and weaknesses of existing tools and approaches, the WUSAN architecture in figure 15 on page 53 was proposed as a powerful instrument for WUA, which eliminates many of the weaknesses of existing tools. The WUSAN architecture makes extensive use of standards: the PMML and the CWM. Notably, the latter is a central constituent for WUSAN: through the XELOPES data mining library, the CWM transformation and data mining packages are intensely leveraged within the WUSAN framework.

The ETL component in figure 15 on page 53 remained as the final missing element to complete the framework. Chapter 4 on page 59 contributed the LOORDSM, which models this last required constituent. In figure 16 on page 60, WUSAN's four-layer architecture was illustrated, the ETL layer of which actually hosts the LOORDSM. As a prerequisite of the LOORDSM, transformation modeling was discussed and streams were introduced, each of which was presented in both in a mathematical perspective that concisely delineated its ideas, instruments, and mechanisms and a nuts and bolts perspective, that is, UML class diagrams that visualized how the LOORDSM has been concretely deployed in Java. As mentioned in the introduction on page 2, this chapter supplied the following contributions with an emphasis on the theoretical perspective:

- (1) It presented the LOORDSM with its clearly structured mathematical data and transformation model. For the first time, a mathematical meta-data and transformation model conforming to the *Common Warehouse Meta-Model (CWM)* was inferred and leveraged for consistent, uniform ETL modeling.
- (2) It provided a structured XML interface for ETL transformation modeling. This means that the theoretical model conveys its structured approach to WUSAN's WusanML interface, providing users with a standardized XML user interface.
- (3) It described how the LOORDSM provides for compatibility with the CWM, thereby allowing for the integration of arbitrary CWM compatible generic data pools into the ETL process and the WUA framework WUSAN.
- (4) It discussed how the LOORDSM supports automating the preprocessing phase of the WUA process, the phase being particularly challenging for the WUA domain.

Finally, in chapter 5, deployment of the LOORDSM was concretely demonstrated by modeling ETL based on a realistic recommendation engine showcase for the KDD Cup 2000 data. The LOORDSM was exposed as the key component of WUSAN, representing the fundamental prerequisite for creating a powerful analytical system, since it significantly simplifies the preprocessing phase. Apart from the benefits of the theoretical perspective, the following practical benefits of the LOORDSM have been extracted in this chapter:

- (1) It was shown that the LOORDSM significantly simplifies the overall WUA process by easing modeling ETL, a sub-process of the WUA process. Consequently, as WUSAN and the LOORDSM both improve the WUA process, the initial challenge of winning over the customer in the Web channel can be tackled more efficiently.
- (2) It was discussed that the LOORDSM with its clearly structured formal mathematical data and transformation model fosters the creation of an *automated closed loop* in concrete practical applications such as a recommendation engine.
- (3) It was demonstrated that due to the fact that the LOORDSM leverages the CWM's meta-data model, arbitrary log formats can be mapped to WUSAN's data warehouse following the same structured approach.
- (4) It was shown that WUSAN's XML interface facilitates the set up of the ETL process, and it was pinpointed that through this XML interface, it is possible to support conceptual tasks graphically, that is, all required XML models can be assembled semi-automatically.

- (5) It was concluded that WUSAN and especially the LOORDSM can be regarded as enablers for future research activities in WUA, EC, and ECRM, as they significantly lower the preprocessing hurdle – an indispensable step prior to any analysis activities – as yet an impediment for further research interactions.

As stated in the introduction, this dissertation does not provide the philosopher's stone for ECRM but rather addresses and overcomes the obstacle of preprocessing Web usage data and related data in a systematic, straightforward, and automated manner. It is this capability that is the key contribution, as it had thus far been lacking, representing an insurmountable hurdle for many research efforts in WUA, EC, ECRM, and beyond and impeding research in those essential areas, as the time spent for preprocessing previously dominated the real research challenges. The LOORDSM has a great potential to alleviate this problem and act as a catalyst for all research activities in the mentioned areas that depend on sophisticated preprocessing capabilities.

Additionally, this dissertation detected a number of open issues and unresolved research questions, the most important of which are summarized in the following, structured in three broad categories:

- (1) **Analytical Issues.** Although this dissertation realized a crucial step towards establishing the ECRM framework in figure 6 on page 23, the actual analytical challenges in view of targeting the initial problem of winning over the customer in the Web channel remains unsolved. Yet, WUSAN provides an instrument that can be leveraged to further investigate these issues.

From an OLAP perspective, defining new useful measures and deploying agreed-upon measures with MDX is a promising field, furthering existing Web reporting advances. From a data mining perspective, the meaningful combination of OLAP and data mining remains a challenge, as well as purposefully employing Web usage mining methods for shaping customer relationships. Both perspectives take the preprocessing capabilities for granted and can now be handled more easily.

- (2) **Modeling Issues.** The LOORDSM is not complete by far. For instance, it does not reflect the special transformations mentioned on page 74. In this context, it is important to decide whether to precalculate a measure with a regular transformation or a special transformation during the ETL process and hence materialize it in the database or whether to calculate the measure solely with MDX. Not only is this a performance issue, but it is also a question of modeling convenience. Related to this issue is the question of how to perform aggregations and abstractions in the data warehouse, for instance, the transition from clicks to sessions. Section 5.3.2 on page 110 outlined various alternatives for aggregated data modeling in the data warehouse. Yet, it is unclear which alternative is the best for specific analytical situations and application domains, and structured approaches are required.

Although storage space is cheap, sooner or later the issue emerges as to what data must be kept on what aggregation level for long-term analysis and what data can be deleted. As mentioned before, WUA involves large volumes of data on a recurring basis. Not all the information collected is decisive in the long-term perspective. For this point, the overall strategic goals (that are subject to constant change) may play a key role, complicating the decision what data must be kept.

- (3) **Technical Issues.** In order to improve WUSAN's usability, the WusanML interface should be used as a basis for creating a GUI that supports easy and intuitive ETL modeling as

sketched in the remark on page 105. Furthermore, the performance issues mentioned in section 5.3 on page 109 must be addressed and deployed in future versions. In particular, the performance tuning issues for the RDBMS are of central importance, since the database's performance has great influence on the overall performance during ETL and any data accesses during analysis.

All aspects considered, this thesis has solved the central aspect of ETL modeling, but at the same time accumulates many new, dependent open issues and research challenges. However, the brisk activities in this research area and even in the open source community¹ demonstrate that the broad direction of this dissertation points to a promising field of future research and business activities.

¹Apart from the open source packages employed for WUSAN, many other open source projects are currently contributing to transforming business intelligence into software that is deployed and accepted on a broad basis. For instance, BIZGRES is a project that develops a data warehouse for business intelligence aiming at covering the complete analytical process. OPENI aims at improving the graphical analytical capabilities for business intelligence, which are compulsory for a broad acceptance. Finally, KETTLE aims at supporting and easing ETL and data migrations for business intelligence.

Appendix A

ADDITIONAL ISSUES OF ELECTRONIC COMMERCE

This chapter covers two amendments to section 2.1 on page 5. In section A.1, the classification of EC applications is completed and, in section A.2 on the next page, the issue of privacy and trust for EC is briefly discussed.

A.1 Classification of Electronic Commerce Applications

In section 2.1.3 on page 7, a classification scheme for EC has been introduced and the B2B and B2C categories have been discussed. The remaining two categories, C2B and C2C, are sketched in section A.1.1 and section A.1.2, respectively.

A.1.1 Consumer-To-Business Electronic Commerce

Consumer-to-business (C2B) EC is similar to B2C EC but entails one fundamental difference. The old system, where organizations offer products and services at certain prices and the decision whether to accept or decline offers lies with consumers, is inverted [Harmsen, 2001]. A C2B marketplace provides consumers with the opportunity to submit offers for precisely defined products and services to organizations over a specialized C2B marketplace. It is then up to the participating organizations to accept or decline offers. The C2B model is hence only suitable for buyer markets.

The most prominent C2B EC example is *Priceline* [PRICELINE], a Web site implementing the *name-your-own-price-model*. This model enables consumers to achieve significant savings by submitting their maximum prices for products or services [Turban and King, 2003, appendix 2A]. Basically, the concept is that of a *reverse auction*, in which sellers submit offers and the lowest-priced seller makes the deal [Harmsen, 2001]. Priceline presents offers submitted by consumers to sellers, who can serve as much of Priceline's guaranteed demand as they can. Alternatively, Priceline searches a database that contains sellers' minimum prices and tries to match supply against demand. In turn, Priceline's customers have to commit themselves to accepting any offer that matches their inquiry, if it is at or below the submitted price.

A.1.2 Consumer-To-Consumer Electronic Commerce

Consumer-to-consumer (C2C) EC refers to definition 2.1 on page 7, limiting the type of transactions to transactions between consumers. C2C EC sites are a modern form of a marketplace (without limitations to a physical location) where buyers and sellers meet and where prices are negotiated on the basis of the rules the marketplace sets up [Vulkan, 2003, p. 153].

Auction Web sites such as *Ebay*¹ [EBAY] or *QXL* [QXL] can be regarded as paradigms for C2C EC. Making use of *forward auctions* (auctions where the price increases with time), individuals and organizations place items for sale on auctioneers' Web sites, who then run the

¹Ebay is the world's largest auction Web site with 100 million registered members as of fall 2004 [EBAY]. It is the largest online marketplace for the sale of goods and services by a mixed community of individuals and SMEs.

auctions on behalf of their customers for a fee. Exchange of products and services and payment take place directly between sellers and auction winners [Turban and King, 2003, appendix 2A].

A.2 Privacy and Trust in Electronic Commerce

As mentioned in section 2.1.3.2 on page 9, privacy and trust is a crucial success factor in EC. Operating and maintaining an infrastructure to conduct EC transactions involves the processing of customers' sensitive personal data. Such data is required to complete transactions and to ensure that customers receive the products and services they purchase promptly and reliably. Moreover, organizations rely on personal data to tailor products and services to their customers' needs, far beyond core transaction processing.

Personalization efforts seem to be in a caustic conflict with customers' privacy concerns, as customers become increasingly anxious about threats to their privacy, when they are prompted to divulge personal information or when they are aware of being tracked online [Kobsa, 2002]. Such privacy concerns on the customer side may have serious consequences for organizations conducting EC: customers are reported to leave Web sites, to provide false registration information, or to generally refrain from conducting online transactions due to privacy concerns.

Over the past decade, three fundamental approaches addressing privacy and trust have evolved: enforcing privacy (i) through *legislation* (compare section A.2.1), (ii) through *self-regulation* (compare section A.2.2 on the facing page), and (iii) through *technical standards* (compare section A.2.3 on page 120), each of which is discussed in the following.

A.2.1 Privacy Through Legislation

Although privacy laws emphasize different aspects of privacy in different countries, they are usually based on a small number of fundamental privacy principles [Kobsa, 2002]. These principles have been stated by the Organization for Economic Co-Operation and Development (OECD) in the guidelines on the protection of privacy and transborder flows of personal data [OECD-PRIVACY], the most important of which from the customer perspective are singled out in the following list.

- (1) **Collection Limitation Principle.** "There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject."
- (2) **Purpose Specification Principle.** "The purposes for which personal data are collected should be specified not later than at the time of data collection and the subsequent use limited to the fulfillment of those purposes or such others as are not incompatible with those purposes and as are specified on each occasion of change of purpose."
- (3) **Use Limitation Principle.** "Personal data should not be disclosed, made available or otherwise used for purposes other than those in accordance with the *Purpose Specification Principle* item 2 except with the consent of the data subject or by the authority of law."
- (4) **Openness Principle.** "There should be a general policy of openness about developments, practices and policies with respect to personal data. Means should be readily available of establishing the existence and nature of personal data, and the main purposes of their use, as well as the identity and usual residence of the data controller."
- (5) **Individual Participation Principle.** "An individual should have the right:

- a) to obtain from a data controller, or otherwise, confirmation of whether or not the data controller has data relating to him;
- b) to have communicated to him, data relating to him within a reasonable time; at a charge, if any, that is not excessive; in a reasonable manner; and in a form that is readily intelligible to him;
- c) to be given reasons if a request made under subparagraphs 5a) and 5b) is denied, and to be able to challenge such denial; and
- d) to challenge data relating to him and, if the challenge is successful to have the data erased, rectified, completed or amended.”

Thorough privacy legislation as in the EU has two major drawbacks. First, the sluggish legislation process bears the risk that laws fall behind technological development. Second, it is difficult to impose such laws on other countries that pursue a different privacy philosophy, for example, the US. This aspect is described in the section below.²

A.2.2 Privacy Through Self-Regulation

Spiekermann et al. [2001] point out that the US relies on self-regulation instead of legislation concerning privacy issues. Organizations conducting EC hence focus on privacy statements and privacy seals to account for privacy issues. The basic idea of self-regulation is the assumption that customers are well aware of privacy issues and thus, market forces are strong enough to assure that only organizations conforming to customers’ expected high privacy standards will eventually survive in a tough competitive environment. Yet as Spiekermann et al. [2001] note, there is often a discrepancy between how customers actually act – concerning careless divulgement of sensitive personal data – and what they demand when asked for privacy.

Moore and Dhillon [2003] projected EC sales to exceed \$100 billion in 2002, reduced by \$3 billion due to (potential) customers’ privacy concerns. Given such figures, the US EC industry has tried to establish a self-regulation policy that centers on the use of *privacy seals*. A privacy seal is meant to create trust between EC Web sites and their customers in terms of privacy. Nowadays, three major privacy seals are prevalent in the US, the most common of which is *TRUSTe* [TRUSTE].³

According to Benassi [1999], TRUSTe is a non-profit privacy seal program for Web sites that are willing to back privacy disclosures by credible third-party assurance. The seal is founded on a set of privacy principles that are similar to those mentioned in section A.2.1 on the preceding page. Namely, TRUSTe covers the following points [compare Benassi, 1999]:

- (1) **Notice.** Web sites must post a privacy statement linked from the home page that includes disclosure of the Web site’s information gathering and dissemination practices.
- (2) **Choice.** Web sites must provide at least the ability for users to opt out of having their personal information used by third parties for secondary purposes.
- (3) **Security.** Web sites must implement reasonable procedures to protect personal information from loss, misuse, or unauthorized alteration.

²Zwick and Dholakia [2001] discuss the contrasting EU and US regulatory philosophies underlying the debates on privacy for EC.

³The other two privacy seals are *WebTrust* [WEBTRUST] and *BBBOnline* [BBB-ONLINE].

- (4) **Data Quality and Data Access.** Web sites must provide a mechanism for users to correct inaccuracies in information stored about them.
- (5) **Verification and Oversight.** TRUSTe provides assurance to users that participating Web sites are following stated privacy practices through initial and periodic reviews in order to ensure that all required criteria continue to be met.
- (6) **Complaint Resolution.** TRUSTe provides users with a structure to resolve complaints that cannot be adequately resolved by TRUSTe licensees.
- (7) **Consequences.** Depending on the severity of the privacy breach, TRUSTe may take legal action or revoke a licensee's privacy seal.⁴

Although privacy seals are the best way to implement privacy self-regulation, following Moores and Dhillon [2003], they cannot take the place of sound privacy legislation, since they create many loopholes that may be exploited even by reputable EC organizations.

A.2.3 Privacy Through Technical Standards: The P3P Privacy Standard

In order to make privacy standards more transparent for customers, the World Wide Web Consortium (W3C) has set up the *Platform for Privacy Preferences (P3P) Project* [W3C-P3P], which proposes a standardization approach for privacy protection on the Web. The idea is to integrate customers' privacy preferences into technology in order to make it easier for them to check what kind of privacy preferences are applicable to certain EC Web sites. Unlike legislation, the P3P approach has the advantage of being able to quickly respond to technological changes that may have an effect on privacy.

According to Grimm and Rosnagel [2000], P3P describes three different tasks in view of exchanging information in a way that must conform to a certain privacy statement. (i) It specifies privacy protection policies as a set of formalized statements. (ii) It describes a process of submitting privacy policy proposals to users, which they can accept or decline. Furthermore, it defines a protocol that enables users to initiate automated negotiation about privacy policies, if desired. (iii) It describes another protocol for the transfer of sensitive personal data.

P3P applications support customers in being informed about the privacy practices of Web sites. Further, they assist in delegating negotiations about how to alter offered privacy statements to make them acceptable for both parties to a computer agent that manages relationships with Web sites semi-automatically. The central element of P3P is its mechanism to manage privacy disclosures rather than to decide whether they are actually compliant with privacy legislation or standards.

Privacy policy proposals are parsed by user agents automatically and compared to pre-configured privacy preferences [Reagle and Cranor, 1999]. In case of incompatibilities, user agents may either prompt users for explicit confirmation or denial or start automatic negotiation by sending a modified privacy policy proposal back to the issuing Web site. The Web site may, in turn, modify the proposal and continue negotiations. P3P does not provide a mechanism to

⁴Current US law imposes few restraints on private parties communicating information about people. The chief legal protection available to seek redress for breach of privacy is *contract law*. If a business promises to keep information private, customers can hold it to this promise by legal means [Volokh, 2000]. Contract law is the most powerful instrument that organizations issuing a privacy seal can apply on behalf of all customers of an EC Web site. Privacy seal licensors make use of this instrument to ensure that privacy seal licensees observe the specified privacy standards.

actually assure that Web sites act according to stated privacy policies. This is the point where neutral instances, issuing privacy seals as described in the previous section, come into play.⁵

⁵Ashley et al. [2002] propose the *Platform for Enterprise Privacy Practices* (E-P3P), which is a privacy policy model that can be used to assure that all information processing activities in an organization comply to the privacy policy stated to its customers. All kinds of internal data accesses must be approved by the E-P3P policy, which is maintained and updated by a *Chief Privacy Officer* (CPO). In contrast to P3P, E-P3P is a means to technically prevent accidental internal violations of stated privacy policies.

Appendix B

MORE ON CRM STRATEGY IMPLEMENTATION

This section raises the question of how to set up and implement CRM strategies. To this end, two strategic frameworks for CRM strategy implementation are introduced.

- (1) The first framework by Sue and Morin [2001] (discussed in section B.1) links knowledge about customers to profits and is used to sift out essential elements of CRM strategies. Furthermore, this work discusses how these elements are interlinked and how they have an effect on corporate profit. The framework can be regarded as an extended version of the functional chain of CRM in figure 3 on page 16.
- (2) The second framework by Payne [2003b] (discussed in section B.2 on page 126) is based on the interaction of five cross-functional corporate processes that deal with *strategy development*, *value creation*, *multi-channel integration*, *information management*, and *performance assessment*. It centers on the alignment of customer-facing processes and dependent internal processes, a prerequisite for CRM that was mentioned in item ii on page 17.

B.1 Strategic Functional Chain of CRM

The strategic functional chain of CRM (depicted in figure 39 on the following page) is modeled using the *results chain technique*, which is described in Thorp [2003]. This technique makes use of three components, which are illustrated in table 2. The components of the strategic functional chain can be divided into the following parts (1) *knowing your customers*, (2) *increasing value-add*, (3) *customer interaction*, (4) *divesting unprofitable customers*, (5) *customer base*, (6) *capturing increased value*, and (7) *ultimate benefits*, each of which is discussed next following Sue and Morin [2001].




Component	Meaning
	An initiative is an action that can be taken to support the desired change. It may be a simple task or an entire project. An initiative can contribute to the production of one or more outcomes.
	An outcome is a change in result. It can be the consequence of one or more initiatives or prior outcomes. Outcomes should ideally be measurable in order to determine if the change program was successful in achieving its goals.
	A contribution refers to the role that an initiative or outcome plays in helping to bring about a subsequent outcome.

Table 2: Components of the results chain technique [compare Thorp, 2003].

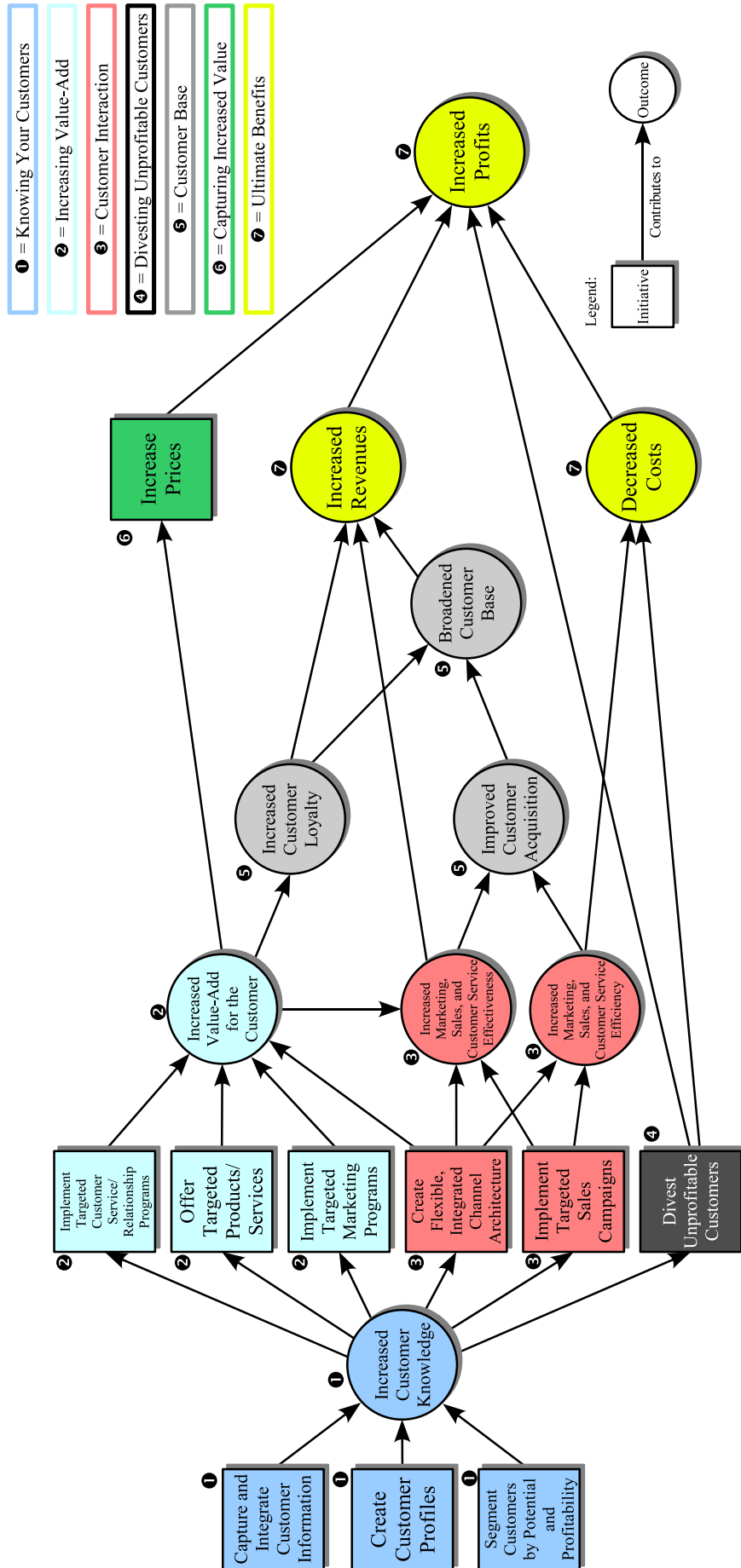


Figure 39: Strategic functional chain of CRM proposed by Sue and Morin [2001].

- (1) **Knowing Your Customers.** Customer knowledge is the necessary starting point for any CRM strategy that aims at increasing corporate profit. This goal can be achieved by providing increased value to targeted segments of profitable and *potentially* profitable customers. In order to gain broadened knowledge about customers, the strategic functional chain in figure 39 on the facing page allows for three initiatives.
 - (a) *Capture and integrate customer information from multiple sources.* This includes demographic and psychographic data, buying and service histories, preferences, complaints, and all other communications with organizations. The data originate from a multi-channel environment as the one depicted in figure 4 on page 19. The integrated data can be used for two purposes: (i) to create customer profiles that can be used to tailor communications and interactions with customers and (ii) to segment customers so as to develop appropriate products and services and supportive marketing programs.
 - (b) *Create customer profiles.* These capture the demographics, needs, purchases, preferences, favored channels, and behaviors of individual customers. They are made available at every customer touchpoint in order to provide for consistent communications.
 - (c) *Segment customers by potential and profitability.* The *Pareto Rule* (or 80:20 rule [compare Newbold et al., 2003, p. 26] suggests that, for organizations, 80% of profits originate from 20% of their customers [Chatranon et al., 2001]. Hence, it makes sense to extract the currently or future most profitable 20% of the customer base by segmenting customers in terms of (potential) profitability to create actionable customer segments.
- (2) **Increasing Value-Add.** As stated in section 2.2 on page 13, CRM involves the creation of value for customers, leading to increased profits for organizations. In order to continuously increase the value-add for customers, the strategic functional chain in figure 39 on the preceding page proposes the following three initiatives: (a) *implement targeted customer service/relationship programs*, for example, loyalty and retention programs, (b) *implement target marketing programs*, that is, build differentiated customer value for different customer segments, and (c) *implement targeted product/service offerings*, that is, implement a differentiation strategy¹ through tailored offers.
- (3) **Customer Interaction.** Customer segments are designed to provide increased value for customers in terms of convenience and service and for organizations in terms of reduced customer service costs (compare footnote 21 on page 17). Organizations are striving for increased marketing, sales, and customer service *efficiency*² and *effectiveness*³. To this end, organizations should create an integrated multi-channel architecture (compare remark below figure 4 on page 19) and implement targeted sales campaigns.
- (4) **Divesting Unprofitable Customers.** In order to achieve the two goals of *increased profits* and *decreased costs*, organizations can choose to divest themselves of unprofitable customers by reducing customer service or raising prices for unprofitable customer segments.
- (5) **Customer Base.** The outcomes of previous initiatives concerning item 2 and item 3 indirectly result in a broadened customer base due to improved customer retention, which is a direct result of customer loyalty⁴ and improved customer acquisition.

¹Compare section 2.2.2 on page 15.

²Efficiency refers to the degree of effort and amount of resources used to achieve customer interaction goals.

³Effectiveness refers to the degree to which organizations are able to meet customer interaction goals.

⁴Customer loyalty generates increased sales and higher revenue. The link between customer loyalty and customer retention has been verified by Lewis [2004].

- (6) **Capturing Augmented Value.** Price increases that are based on delivering augmented value and that are appropriately communicated to the customer base do not inevitably impact market shares negatively and directly result in increased profits.
- (7) **Ultimate Benefits.** As mentioned in the introduction of section 2.2 on page 13, increasing profits by maximizing the CLVs of customers is the ultimate goal of CRM activities. *Increased Profits* represents the final element of the strategic functional chain of CRM in figure 39 on page 124, which is the result of the initiatives and outcomes described in item 1 on the previous page to item 6.

B.2 Strategic Process Alignment Framework for CRM

While the functional chain of CRM in figure 39 on page 124 emphasizes the functional interrelations of strategic initiatives and outcomes, the strategic process alignment framework in figure 40 on the facing page, proposed by Payne [2003b], identifies five elementary corporate processes that have to be optimized and coordinated in order to establish a basis for all CRM activities. Following Payne [2003b], the five CRM-enabling processes (1) *strategy development process*, (2) *value creation process*, (3) *multi-channel management process*, (4) *information management process*, and (5) *performance assessment process* are briefly discussed next.

- (1) **Strategy Development Process.** As stated in section 2.2.2 on page 15, CRM must be embedded into a higher-ranking corporate strategy, since it is more than merely deploying ICTs. On the one hand, organizations must constantly monitor and adjust their *business strategies* in terms of (i) strategy profile, strategy purpose, strategy performance, and strategic position, (ii) the state of the industry's evolution, (iii) competitors' profiles and activities, (iv) delivery channels and multi-channel management, and (v) advances in ICTs. On the other hand, organizations are also obliged to constantly review their *customer strategies* in terms of (a) the current customer strategy status, (b) quality and goodness of customer segments, (c) knowledge that can be retrieved from the customer base (and its value and benefit for the organizations), and (d) product/service involvement and the complexity of customer purchasing behavior.
- (2) **Value Creation Process.** This process deals with transforming the outputs of the strategy development process into programs that both *extract* and *deliver* value, that is, (i) determine what value organizations can *provide* to their customers, (ii) determine the value that organizations *receive* from their customers, and (iii) manage this value exchange, eventually maximizing the CLVs of customers in profitable customer segments.
- (3) **Multi-Channel Integration Process.** In the context of collaborative CRM, multi-channel management has been mentioned as a prerequisite for CRM in section 2.2.3 on page 18. Due to its high complexity, Payne [2003b] assigns the multi-channel integration process to the set of CRM-enabling processes. The multi-channel integration process involves decisions about the most suitable combination of channels, initiatives that ensure customer experiences will be positive, and efforts to achieve consistent cross-channel communications with customers [Payne, 2003a].

Definition B.1 (Multi-Channel Management). Multi-channel management is the use of more than one channel to manage customer relationships in a way that is consistent and coordinated across all channels used [Stone et al., 2002].

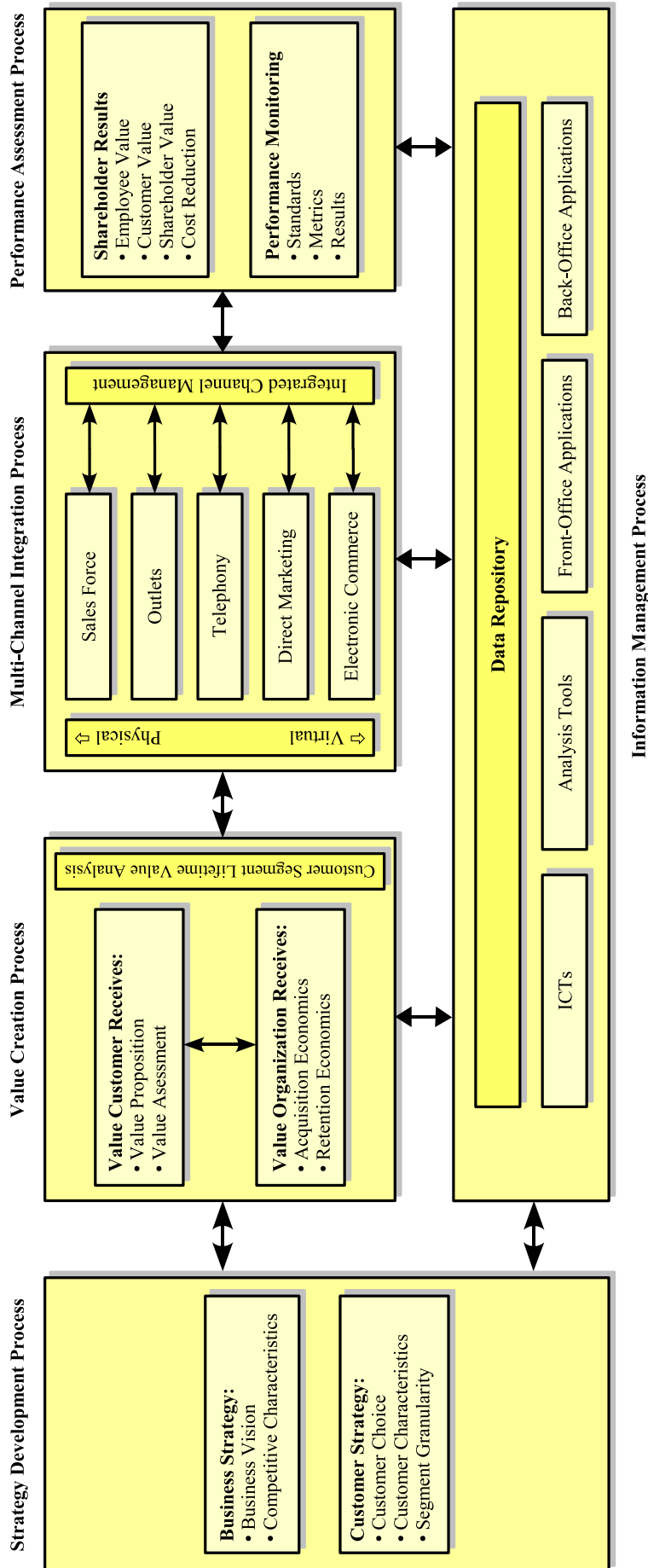


Figure 40: Strategic process alignment framework for CRM, adapted from Payne [2003b].

According to Stone et al. [2002], there are two reasons for the importance of multi-channel management. (i) Recent developments in ICTs for new channel types, namely increasing reliability, performance, and unlimited storage, accelerate the convergence of Internet-based channels (compare remark below definition 2.4 on page 20). (ii) Customers expect new ICTs to be used to manage them consistently across all channels.

From the point of view of organizations, multi-channel management of a multi-channel environment (compare figure 4 on page 19) holds the following risks: (i) *Deficient channel coordination* causes frustrated customers. If poorly coordinated channels are offered, customers may not be able to choose the optimal communication channel. Their frustration is aggravated if multiple channels disseminate inconsistent messages [Schögel and Sauer, 2002]. (ii) *Data integration problems* arise inevitably, since different channels make use of diverse data collection methods leading to behavioral data with varying granularity and data quality. (iii) *Organizational boundaries* may complicate the synchronization of internal processes and customer-facing processes (both from an organizational and an ICTs perspective) [Stone et al., 2002].

In contrast, multi-channel management holds the following benefits: (i) *Increased market coverage*, that is, avoidance of partial market coverage due to limited multi-channel capabilities [Schögel and Sauer, 2002]. (ii) *Increased customer benefits* for two reasons. On the one hand, customers can choose the way they wish to interact with organizations and are able to switch between channels when it suits them [Stone et al., 2002]. On the other hand, different channels can be configured to address specific customer segments with particular requirements. (iii) *Increased efficiency* through sharing of processes, ICTs, and information, which eventually leads to more organizational flexibility and reduced overall channel costs [Stone et al., 2002]. (iv) *Balance of risks* due to the avoidance of strong dependencies on specific customer segments. A multi-channel environment targets different customer segments, hence lowering such dependencies and minimizing the overall risk [Stone et al., 2002].

- (4) **Information Management Process.** As depicted in figure 40 on the previous page, the information management process is concerned with the collection of customer information from all customer touchpoints in order to create customer profiles that help to improve the quality of customer experiences. It involves integrating data repositories, analytical tools, ICTs, and front- and back-office applications.

This process plays an important role for EC channels (see item 5 on page 19), as large volumes of behavioral data can be collected easily in such channels, if sophisticated data collection methods are available (compare section 3.2 on page 34).

- (5) **Performance Assessment Process.** This process helps to ensure that strategic CRM goals are actually met. According to Payne [2003b], the two main instruments of assessing the achievement of strategic goals are (i) *shareholder results*, that is, a *macro view* on CRM performance of organizations, and (ii) *performance monitoring*, that is, a *micro view* on different aspects of CRM activities of organizations, which is primarily comprised of metrics and key performance indicators.

Appendix C

MINING BASES

This chapter refers to item 7 on page 68. The unified view of data mining mentioned there is outlined briefly here and the notion of a mining base is introduced. Following Thess and Bolotnicov [2004, chapter 3], applying data mining algorithms consists of two phases: (1) the *training phase* and (2) the *application phase*.

- (1) **Training Phase.** Let $\mathbf{T} = (X_1, \dots, X_n)^\top$, $X_\ell \in \mathbb{R}^d$, $\ell = 1, \dots, n$ be a data matrix of training data with d attributes. A function $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$, $f \in V$ is sought with V being an appropriate function space over \mathbb{R}^d . Given an operator $R : V \rightarrow \mathbb{R}$, f is the solution of

$$\min_{f \in V} R(f). \quad (\text{C.1})$$

A data mining algorithm solves equation (C.1) for an appropriate sub-space of V . The situation for the training phase is illustrated in figure 41.

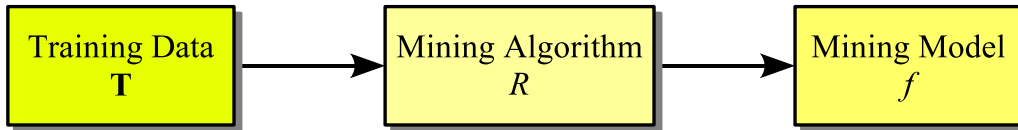


Figure 41: The training phase of data mining algorithms.

- (2) **Application Phase.** Let $\tilde{\mathbf{T}} = (\tilde{X}_1, \dots, \tilde{X}_m)^\top$, $\tilde{X}_\ell \in \mathbb{R}^d$, $\ell = 1, \dots, m$ be a data matrix of application data compatible with \mathbf{T} . Applying f to $\tilde{\mathbf{T}}$ yields $\tilde{Y} = \{f(\tilde{X}_\ell) \in \mathbb{R}^d\}_{\ell=1}^m$. The application phase amounts to applying the mining model to the application data as illustrated in figure 42.

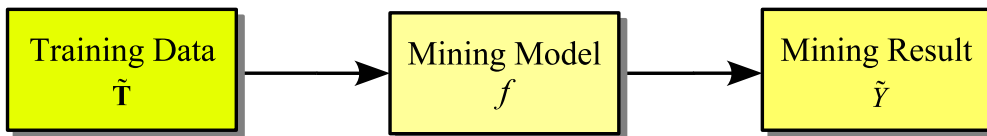


Figure 42: The application phase of data mining algorithms.

All records involved in data mining are represented by real-valued vectors that initially lack an “interpretation”. This interpretation is provided by meta-data $M_{\mathbf{T}} = M_{\tilde{\mathbf{T}}}$. $M_{\mathbf{T}}$ is referred to as a *mining basis* of \mathbf{T} .

In linear vector spaces, the coordinates of a vector represent an interpretation of the vector relative to its basis. If the vector’s interpretation relative to a different basis is sought, a *basis transformation* is applied to the vector, yielding new coordinates.

In data mining, the situation is similar. By default, all records are provided relative to the standard basis of \mathbb{R}^d . If a record’s interpretation relative to a specific application domain is sought, a meta-data mapping (corresponding to the basis transformation) is applied to it, yielding its interpretation. Hence, it can be concluded that the `MiningDataSpecification` class provides a means of modeling a mining basis.

Appendix D

MORE ON STREAMS

This chapter provides the UML class diagrams for the different stream classes introduced in section 4.2.2.2 on page 70. Furthermore, in section D.6 on page 133 and section D.7 on page 137, the two helper classes `MiningUpdatableSqlSource` and `VectorFilter` are discussed. The former is a prerequisite for the database streams discussed in item 3 on page 71 and the latter is employed by the filter stream class `MiningVectorFilterStream` discussed in section D.5 on page 133. The detailed Java API documentation of class employment and initialization has been linked with the respective classes in the PDF version of this thesis.¹

D.1 A Stream Prototype

This section ties in with section 4.2.2 on page 68 where the general stream properties were discussed. Figure 43 depicts the UML class diagram of the abstract class `MiningInputStream`, which models a stream prototype.

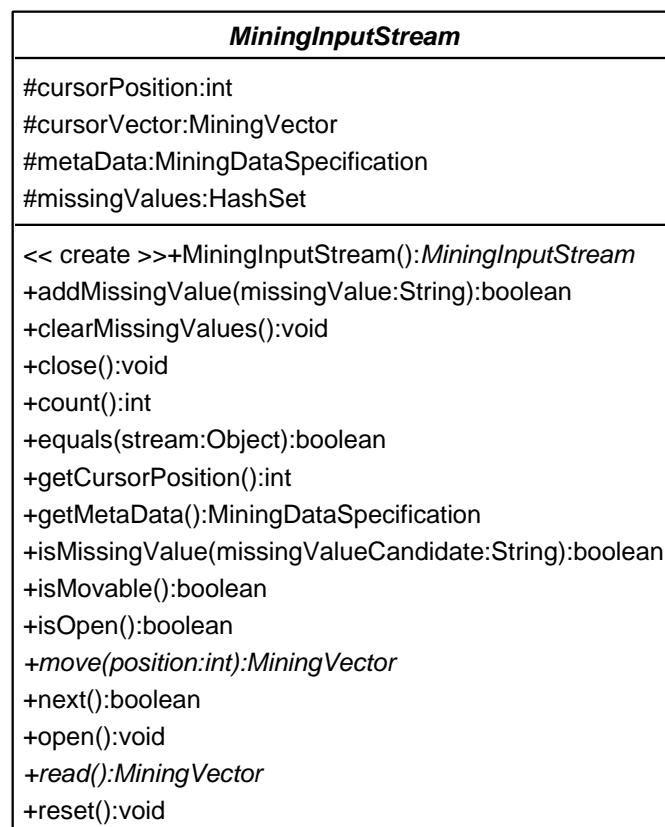


Figure 43: The `MiningInputStream` class models the prototype of a stream.

¹See the remark on page 4.

Figure 44 depicts the `UpdatableStream` interface, which is employed to add writing access to a stream class.

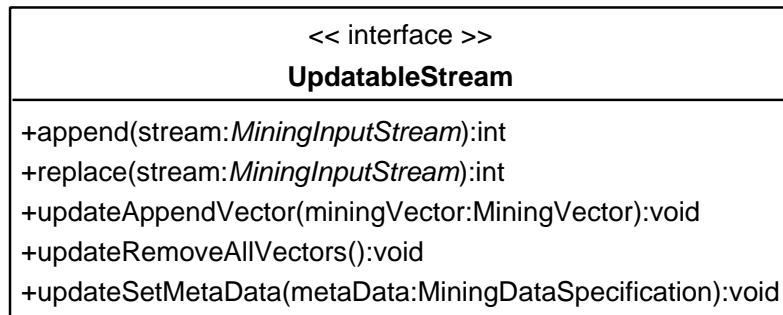


Figure 44: The `UpdatableStream` interface provides writing stream access.

D.2 Memory Streams

Memory streams were discussed in item 1 on page 70, where the two memory stream classes `MiningCollectionStream` and `MiningArrayStream` were introduced. Their UML class diagrams are depicted in figure 45 and in figure 46 on the facing page.

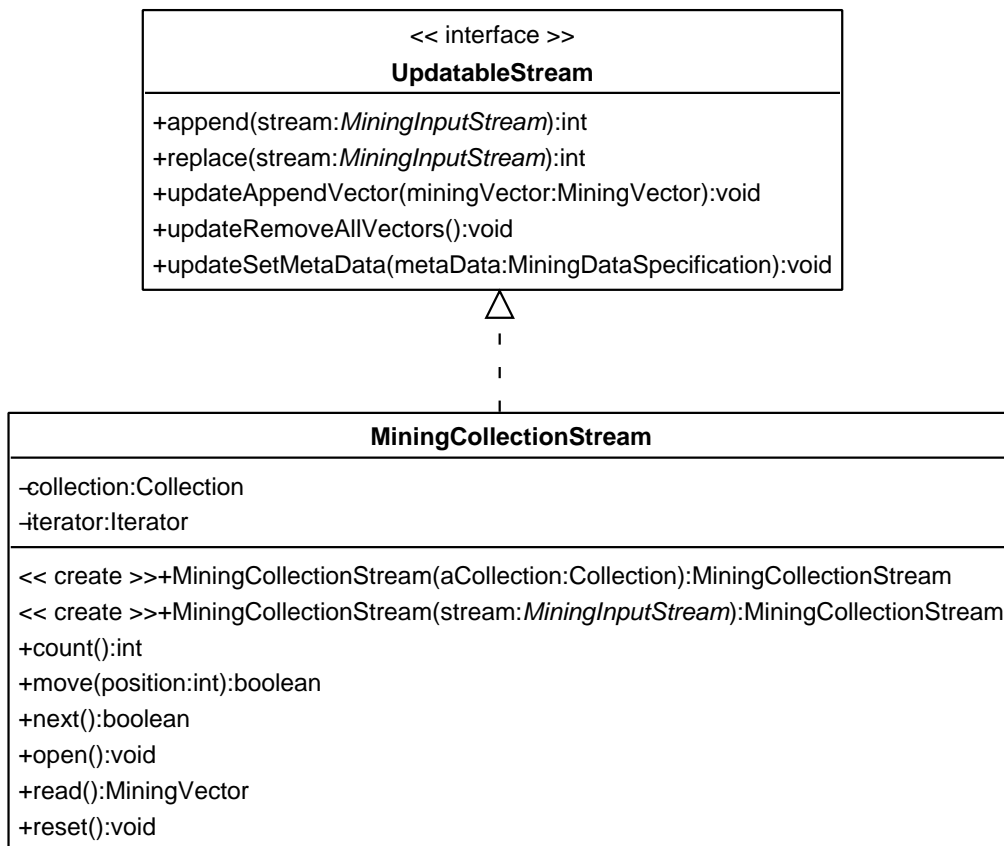


Figure 45: WUSAN's `MiningCollectionStream` class (based on Java collections).

MiningArrayStream
-doubleArray:double[][] -objectArray:Object[][]
<< create >>+MiningArrayStream(aMiningDoubleArray:double[][]):MiningArrayStream << create >>+MiningArrayStream(aMiningDoubleArray:double[],someMetaData:MiningDataSpecification):MiningArrayStream << create >>+MiningArrayStream(stream:MiningInputStream):MiningArrayStream << create >>+MiningArrayStream(aMiningObjectArray:Object[],someMetaData:MiningDataSpecification):MiningArrayStream +count():int +move(position:int):boolean +next():boolean +read():

Figure 46: XELOPES' `MiningArrayStream` class (based on arrays).

D.3 Flat File Streams

Flat file streams were discussed in item 2 on page 70, where the flat file stream classes of figure 47 on the following page were sketched. The abstract `MiningFileStream` class models the prototype of a flat file stream (adopted from XELOPES without changes). Compared to the original XELOPES classes, the `MiningArffStream` and the `MiningCsvStream` classes have been adapted to implement writing stream access.²

D.4 Database Streams

Database streams were discussed in item 3 on page 71, where the database stream classes of figure 48 on page 135 were discussed in detail. Database streams are employed within the data warehousing component of WUSAN in figure 15 on page 53.

D.5 Filter Streams

Filter streams were discussed in item 4 on page 71, where the classes depicted in figure 49 on page 136 were introduced. The abstract `MiningFilterStream` class represents the prototype of a filter stream, the principal feature of which is an embedded instance of the `MiningInputStream` class in figure 43 on page 131. The filters and transformations of the sub-classes of the `MiningFilterStream` class operate on the embedded stream.

D.6 Managing SQL Data Sources

Database streams were discussed in item 3 on page 71. It was stated that the relational tables required for `MiningTableSqlStreams` and `MiningQuerySqlStreams` must conform to a special format. The `MiningUpdatableSqlSource` class accomplishes the management and administration of such tables.

²Some of the parameters of the constructors of the `MiningCsvStream` class can be set to `null`, in order to activate auto-detection for this parameter. A detailed description of feasible parameter settings can be found in the Java API documentation. Furthermore, the sub-class `MiningUpdatableCsvStream` implements constructors with the same signatures and functionality as its super-class. These constructors are omitted in figure 47 on the next page to allow a more compact illustration.

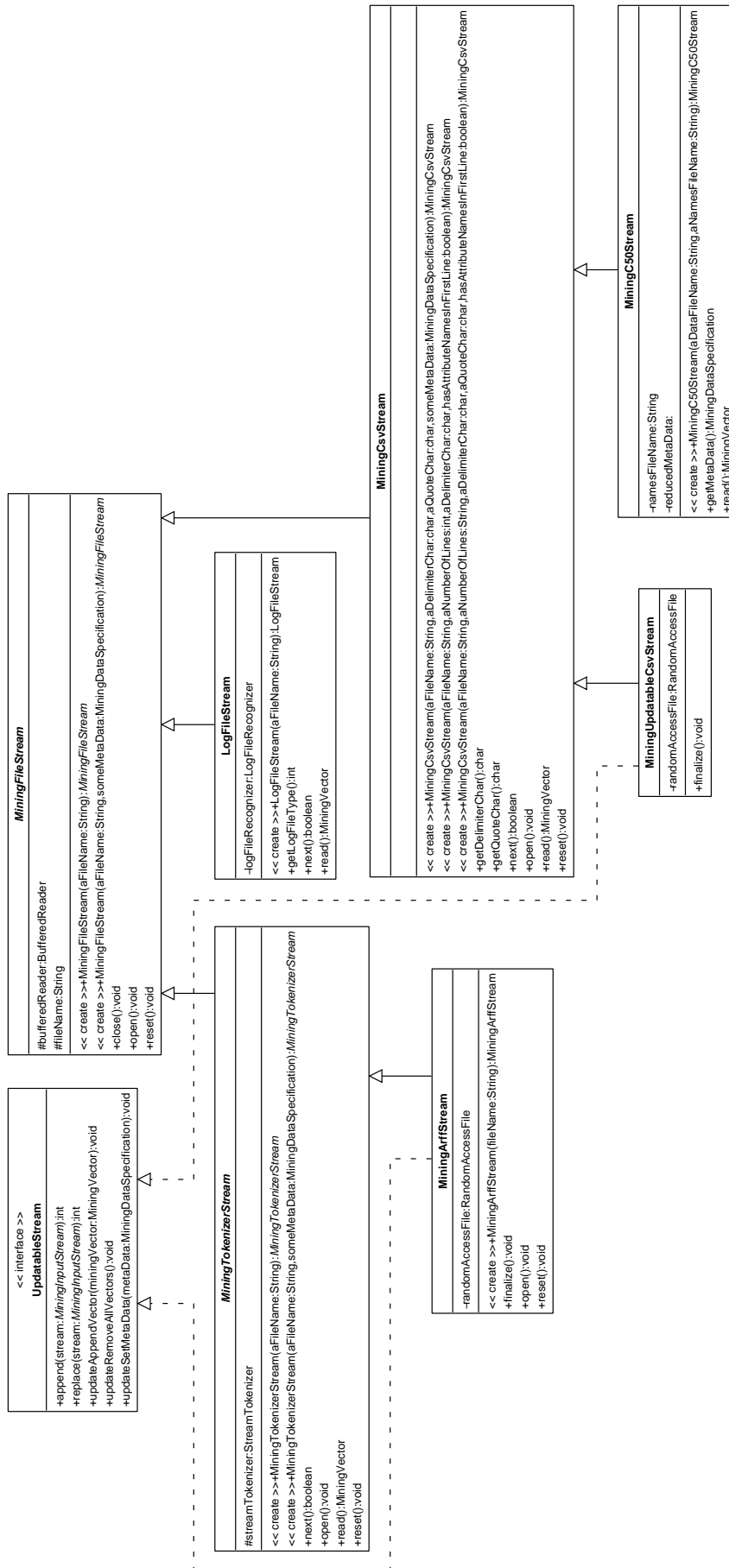


Figure 47: The flat file stream classes (UML class diagram).

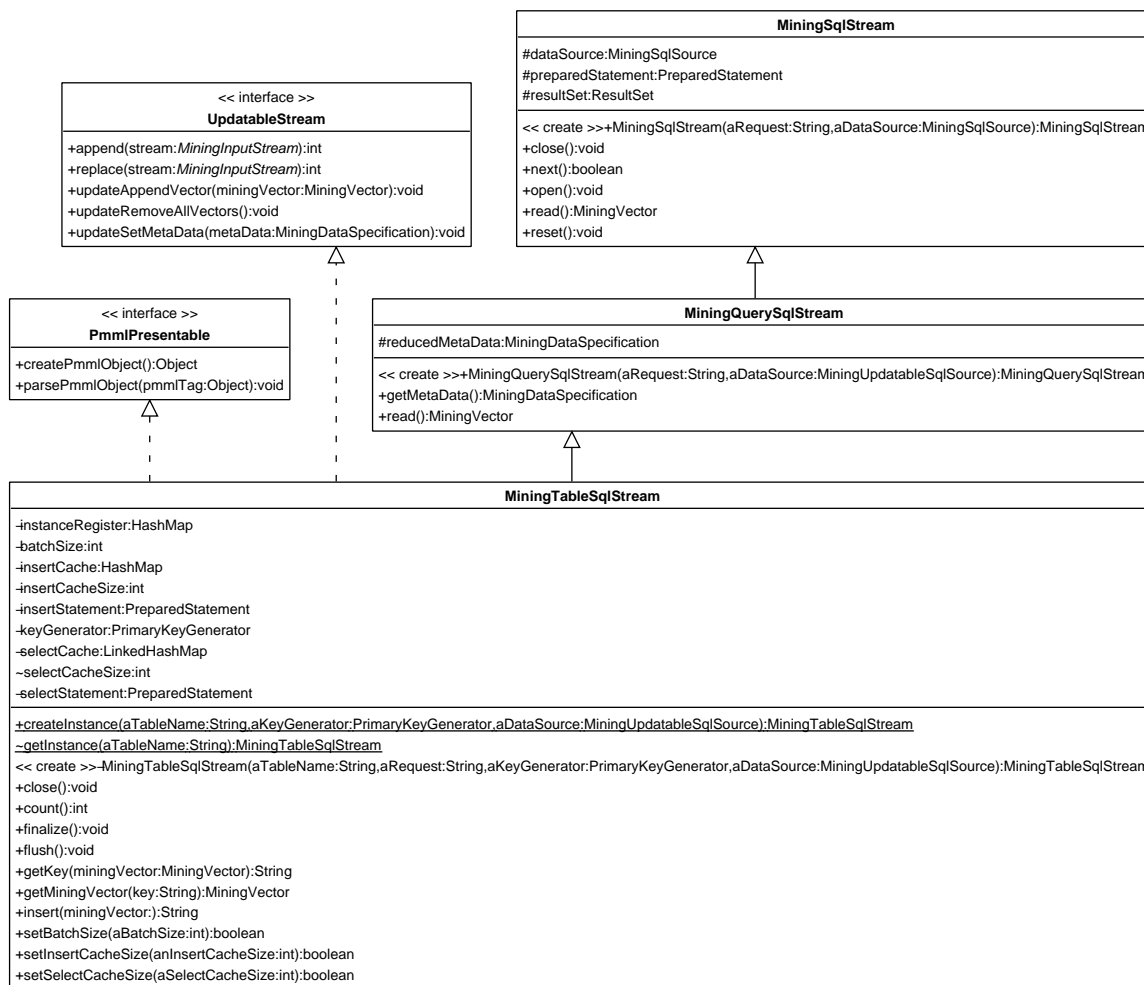


Figure 48: WUSAN’s database streams (UML class diagram).

The UML class diagram of figure 50 on page 137 illustrates its relevant methods. Instances of the class are created by reading a properties file that contains the data required to access an RDBMS. A certain instance operates on one particular schema of the database. It is then possible to create database tables by employing the `createTable` method. This method takes meta-data conforming to figure 17 on page 65 as input and maps each attribute canonically to an RDBMS attribute. For example, a categorical attribute is mapped to `VARCHAR(100)`, which is a good choice for most categorical attributes. However, some categorical attributes such as an attribute modeling URLs contains categories larger than 100 characters. The class features the two methods `addKeywordForLColumn` and `addKeywordForXLColumn`, both of which add keywords submitted as parameters to internal lists. All categorical attributes with names matching a keyword of one of the two lists are mapped to `VARCHAR(500)` or `VARCHAR(1000)`, as the case may be.

The meta-data submitted to create a database table are stored in a repository in PMML format and can be retrieved with the `readMetaData` method or altered with the `writeMetaData` method.

Furthermore, this class handles the management of instances of the `StarSchema` class from the LOORDSM in figure 27 on page 87. By registering a star schema with the `registerStarSchema` method, its WusanML description is retrieved and stored in an internal

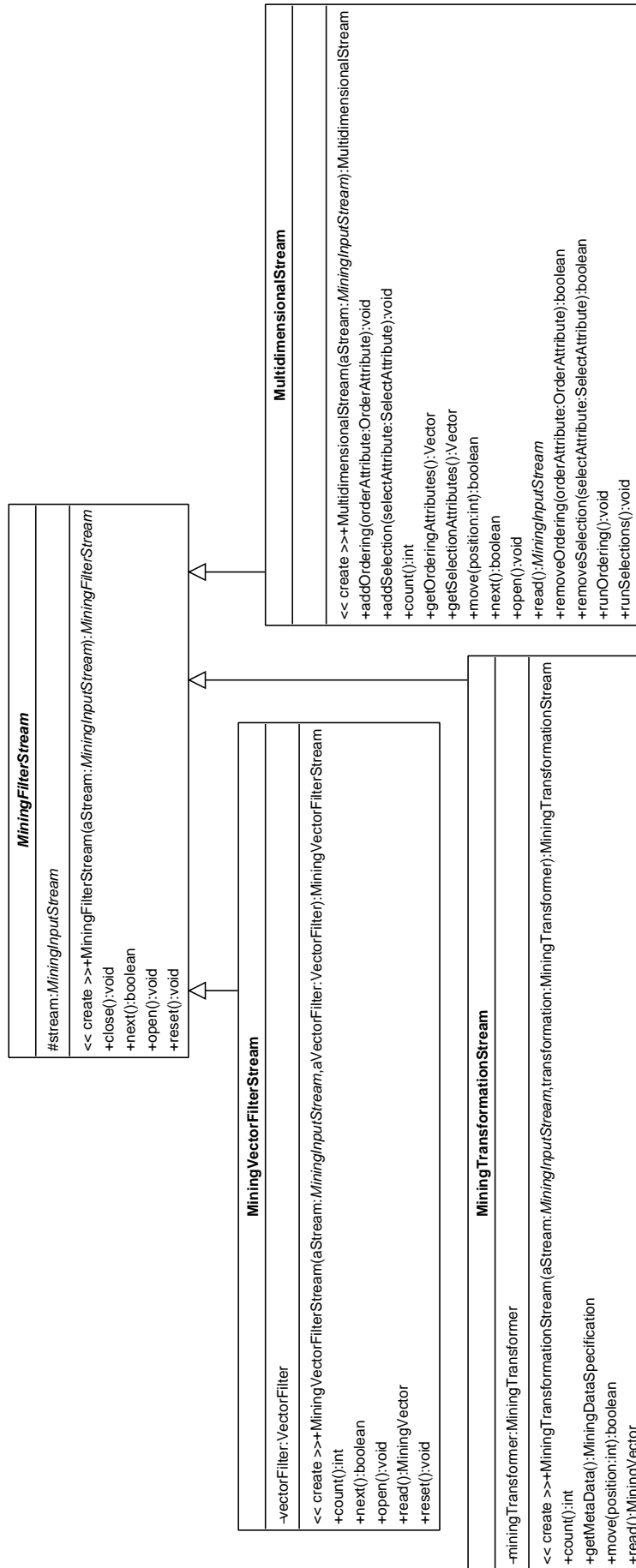


Figure 49: The filter stream classes (UML class diagram).

registration table.³ Consequently, the star schema becomes persistent and can be restored later with the `readStarSchema` method, the argument of which indicates the star schema's fact table. Before a star schema's table can be dropped, it must be unregistered with the `unregisterStarSchema` method.⁴

All other methods are self-explanatory. Generally, methods concerning meta-data management and star schema management should not be invoked explicitly by users, since these methods are invoked implicitly by instances of database streams and star schemas.

MiningUpdatableSqlSource
-databaseSchema:String -keywordListForLColumns:ArrayList -keywordListForXLColumns:ArrayList -tableNameCache:ArrayList
<pre> << create >>+MiningUpdatableSqlSource(prpFile:String,aSchema:String):MiningUpdatableSqlSource +addKeywordForLColumn(keyword:String):void +addKeywordForXLColumn(keyword:String):void +clearKeywordLists():void +clearTable(tableName:String):void +count(tableName:String):int +dropTable(tableName:String):boolean +hasTable(tableName:String):boolean +readMetaData(tableName:String):MiningDataSpecification +readStarSchema(tableName:String):StarSchema +registerStarSchema(starSchema:StarSchema):void +removeMetaData(tableName:String):void +unregisterStarSchema(starSchema:StarSchema):boolean +writeMetaData(tableName:String,metaData:MiningDataSpecification):void </pre>

Figure 50: The `MiningUpdatableSqlSource` class manages access to an RDBMS.

D.7 Vector Filters

Instances of the `VectorFilter` class in figure 51 on the following page are employed to filter vectors. This means that they are created for a given `MiningDataSpecification`. A filter expression can be assigned to each attribute by invoking one of the `addFilterExpression` methods. Figure 52 on page 139 depicts the internal filter structure: one or more filter expressions – connected by disjunction – comprise an attribute filter. A filter expression is a *Boolean expression* in case of a numeric attribute and a *regular expression* otherwise. These attribute filters can then be linked with a Boolean expression (referred to as the *linkage expression*). This mechanism of linking filter expressions and attribute filters results in a very

³The WusanML DTD is depicted in the appendix chapter F on page 147.

⁴It is important to note that the `dropTable` method of the `MiningUpdatableSqlSource` class drops the indicated table without accounting for the fact that the table may be part of one or more star schemas. Hence, tables involved in star schemas should never be dropped directly by invoking this method. Rather, the `drop` method of the `StarSchema` class in figure 27 on page 87 should be employed to drop a star schema. This method determines whether a table is involved in other star schemas and drops tables only if no dependencies exist.

powerful vector filter class. Once the attribute filters are configured, the complete vector filter can be applied to a vector by invoking the `match` method, which returns `true` if the filter matches and `false` otherwise.

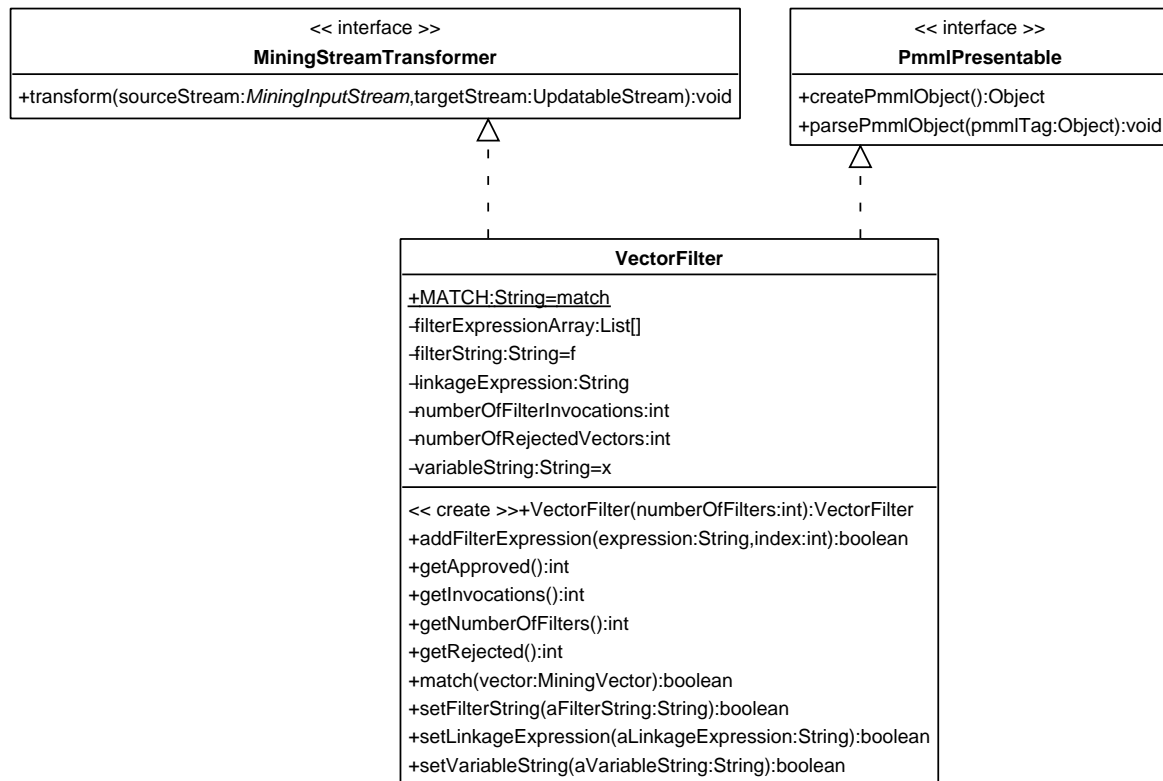


Figure 51: The `VectorFilter` class (UML class diagram).

Example (Configuration of a Vector Filter). Given is a stream with meta-data that comprises three attributes modeling a resource invocation on a Web server: (i) a categorical attribute "URL", (ii) a categorical attribute "timestamp dd-MM-yyyy_hh:mm:ss", and (iii) a numeric attribute "view time in seconds". Suppose that the data required for analysis must meet the following conditions, which can be established in an initial data cleaning pre-processing step: (1) all URL invocations of JPEG and GIF pictures must be filtered, (2) only data from March 2005 may be considered, and (3) only view times of 10 seconds or more are relevant. The resulting vector filter configuration with WusanML is depicted in listing D.1.

Listing D.1: Vector filter configuration with WusanML.

```

1 <VectorFilter filterString="filter" variableString="x" linkageExpression="\
  →filter1_||_!filter2_||_filter3">
  <AttributeFilter>
3     <FilterExpression>\. [gG] [iI] [fF]</FilterExpression>
     <FilterExpression>\. [jJ] [pP] [eE] [gG]</FilterExpression>
5 </AttributeFilter>
  <AttributeFilter>
7     <FilterExpression>03-2005</FilterExpression>
  </AttributeFilter>
  <AttributeFilter>
9     <FilterExpression>x<10</FilterExpression>
11 </AttributeFilter>
</VectorFilter>
  
```

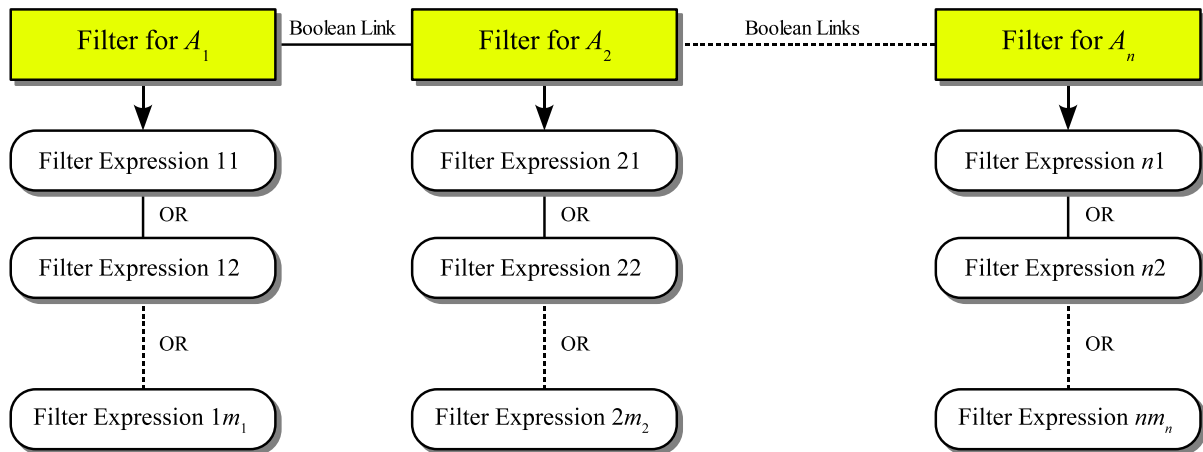


Figure 52: Internal structure of an instance of the `VectorFilter` class.

Instead of defining the vector filter with `WusanML`, it can be set up directly by creating an instance of the `VectorFilter` class in figure 51 on the preceding page. The `setFilterString` method sets the string that refers to an attribute filter in the linkage expression. In listing D.1 on the facing page, this string is set through the `filterString` parameter. The `linkageExpression` parameter in listing D.1 on the preceding page contains a Boolean expression that links the attribute filters to determine whether the vector filter matches. In the example, the attribute filters are linked by disjunction with the second attribute filter negated. In figure 51 on the facing page, the `setLinkageExpression` method can be employed to set the linkage expression.⁵ Finally, the `setVariableString` method is employed to set the string representing variables for Boolean expressions for numeric attributes. In listing D.1 on the preceding page, the `variableString` parameter is employed to set this string.

Remark. (a) Vector filters are a fundamental prerequisite for the `MiningVectorFilterStream` class in figure 51 on the facing page. This class applies its embedded vector filter to the embedded stream. The resulting stream delivers only those vectors that do not match the filter criterion. The `VectorFilter` class implements the `MiningStreamTransformer` interface and can hence be regarded as a special transformation.⁶

(b) In item 1 on page 27, it was mentioned that *data cleaning* is a fundamental task during the preprocessing phase. The `VectorFilter` and `MiningVectorFilterStream` classes address this fundamental aspect of preprocessing and can therefore be regarded as a crucial contribution to covering the preprocessing phase with WUSAN.

⁵Setting the linkage expression is not mandatory, since attribute filters are linked through disjunction by default.

⁶Compare page 74 for special transformations.

Appendix E

MORE ON MAPPINGS

E.1 One-To-One Mappings

This section complements the discussion of one-to-one mappings in item 1 on page 76 investigating implementational details of the `OneToOneMapping` class, which is depicted in figure 53 as a UML class diagram. Since attribute selections in XELOPES are realized with attribute names that must be unique, the source attribute of the one-to-one mapping A is set with the `setSourceName` method. The `transformAttribute` method implements the meta-data transformation T_{MA} that delivers the target attribute \bar{A} , which bears the name set with the `setTargetName` method. If the Boolean variable `removeSourceAttributeFlag` is set to false with the `setRemoveSourceAttributeFlag` method, the mapping's image consists of the target attribute \bar{A} and the source attribute A ; otherwise the source attribute A is removed. The `transformAttributeValue` method computes the result of the real-valued transformation $t_A(a)$, $a \in \mathbb{R} \cup \{\emptyset\}$.

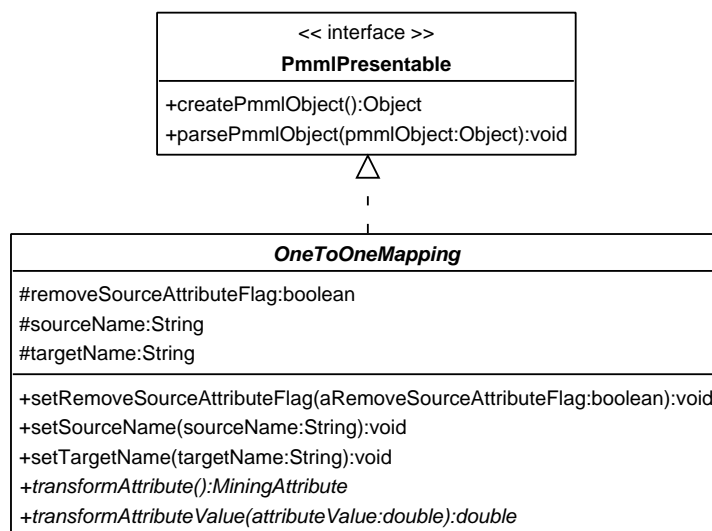


Figure 53: Crucial methods and variables of the `OneToOneMapping` class.

Remark. The `OneToOneMapping` class implements the `PmmlPresentable` interface, which consists of the two methods `createPmmlObject` and `parsePmmlObject`. The former can be employed to create a PMML serialization of an instance; the latter can be employed to import a PMML description containing the relevant information to restore the instance. Figure 53 actually depicts WUSAN's modified version of the original XELOPES class. Its realization of the `PmmlPresentable` methods implement a WusanML serialization. This XML variant (sketched in the appendix chapter F on page 147) provides a description that models CWM transformations of arbitrary complexity in order to compensate the PMML's weakness with respect to modeling transformations.¹

¹Previously mentioned in item iii on page 50.

E.2 One-To-Multiple and Multiple-To-One Mappings

This section complements the discussion of one-to-multiple and multiple-to-one mappings in item 2 on page 76. Both mappings are modeled in XELOPES with the `OneToMultipleMapping` class, which is depicted in figure 54 as a UML class diagram. It comprises the `isOneToMultipleMapping` flag, which indicates the mapping type. If the class models a one-to-multiple mapping T_A , the `setFeatureName` method can be employed to set the name of the source attribute A and the `setClassifierName` method can be employed to set the names of the target attributes $\bar{A}_1, \dots, \bar{A}_m$. If the class models a multiple-to-one mapping, both methods are used conversely. All remaining methods are used analogously as in the case of a one-to-one mapping.

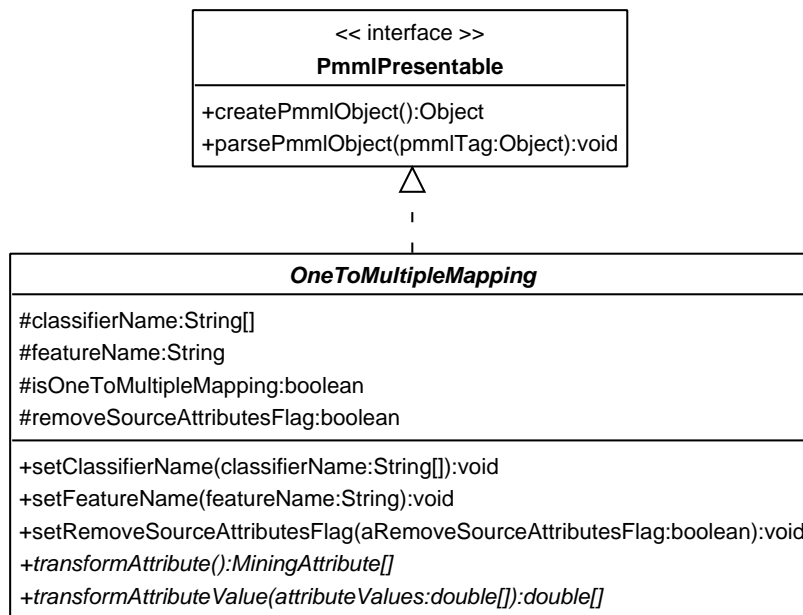


Figure 54: Crucial methods and variables of the `OneToMultipleMapping` class.

Remark. The remark for the `OneToOneMapping` class above also applies to the `OneToMultipleMapping` class.

E.3 Multiple-To-Multiple Mappings

This section complements the discussion of the `MultipleToMultipleMapping` class in item 3 on page 77. This class provides three alternatives for creating the actual multiple-to-multiple mapping: (1) it can be created by embedding one or several instances of the `OneToOneMapping` or `OneToMultipleMapping` classes, which together comprise a *decomposable* multiple-to-multiple mapping or (2) it can be modeled with a sub-class of the `MultipleToMultipleMapping` class by overriding the `transformAttribute` and `transformAttributeValue` methods, leading to an *indecomposable* multiple-to-multiple mapping, that is, it is a mapping in the strict sense or (3) it can be modeled with a combination of both approaches, which results in a *decomposable* multiple-to-multiple mapping.²

²The `OneToOneMapping` and `OneToMultipleMapping` classes must be embedded in a `MultipleToMultipleMapping` class, since neither of them implements the `MiningTransformer` interface, which is required for actually firing the transformation. Compare page 78.

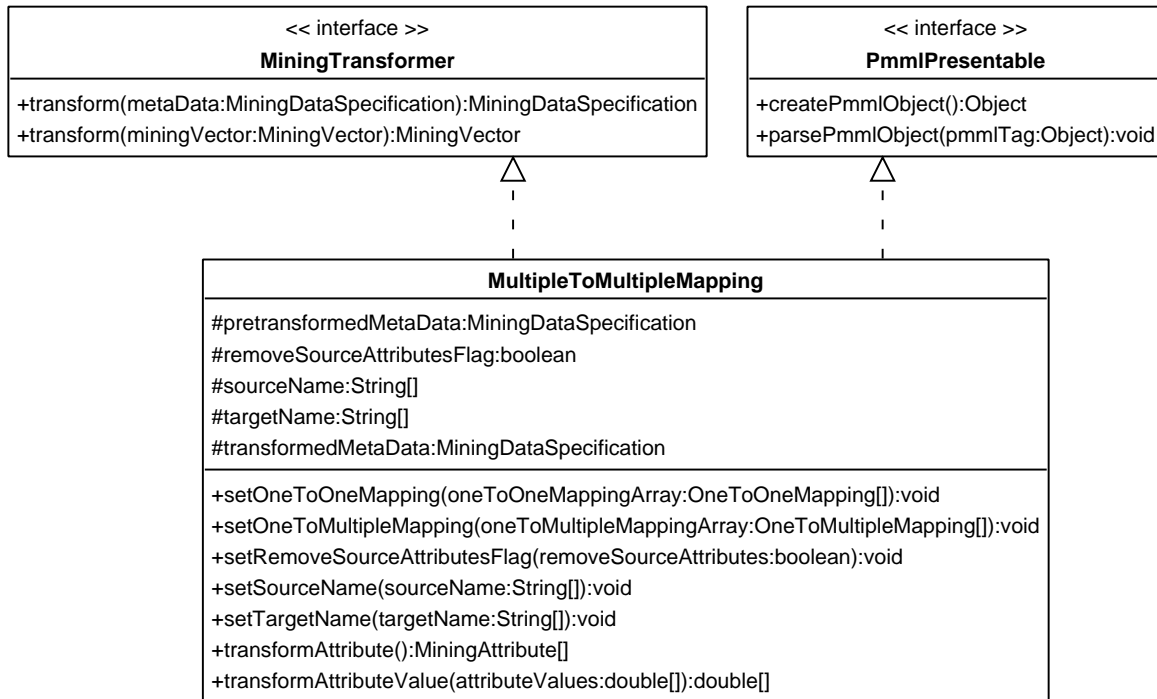


Figure 55: Crucial methods and variables of the `MultipleToMultipleMapping` class.

The crucial methods and variables of the `MultipleToMultipleMapping` class, depicted as a UML class diagram in figure 55, are discussed in the following:

- (i) **pretransformedMetaData variable.** This variable stores the source meta-data, which are the same for all involved mappings.³
- (ii) **removeSourceAttributesFlag variable.** This variable is employed in the same manner as for the other mapping classes.
- (iii) **sourceName variable.** This variable contains the names of the source attributes

$$A_1, \dots, A_m,$$

each of which must be contained in the `pretransformedMetaData` variable.

- (iv) **targetName variable.** This variable contains the names of the target attributes

$$\bar{A}_1, \dots, \bar{A}_m,$$

each of which is contained in the `transformedMetaData` variable after executing the meta-data transformation method of the `MiningTransformer` interface.

- (v) **transformedMetaData variable.** This variable stores the transformed meta-data of the overall transformation and simultaneously serves as a cache to avoid unnecessary meta-data transformations.
- (vi) **MiningTransformer interface.** As the `MultipleToMultipleMapping` class implements this interface, it models a transformation that can be directly executed. The

³Compare footnote 32 on page 78.

following execution order for the contained mappings applies: (a) If applicable, all embedded one-to-one mappings are executed in the order of their insertion. (b) If applicable, all embedded one-to-multiple and multiple-to-one mappings are executed in the order of their insertion. (c) If applicable, the multiple-to-multiple mapping is executed.

- (vii) **setOneToOneMapping method.** This method sets the array of embedded one-to-one mappings.
- (viii) **setOneToMultipleMapping method.** This method sets the array of embedded one-to-multiple and multiple-to-one mappings.

All remaining methods and interface implementations are employed analogously to the other two mapping classes.

E.4 Composing Transformations and Mappings

This section complements the paragraph about nesting and concatenating transformations on page 77. It was discussed there how the `MiningTransformationStep` class is employed to compose multiple-to-multiple mappings horizontally. Its UML class diagram is illustrated in figure 56. As opposed to the original XELOPES class, WUSAN's version of the class realizes a WusanML serialization.

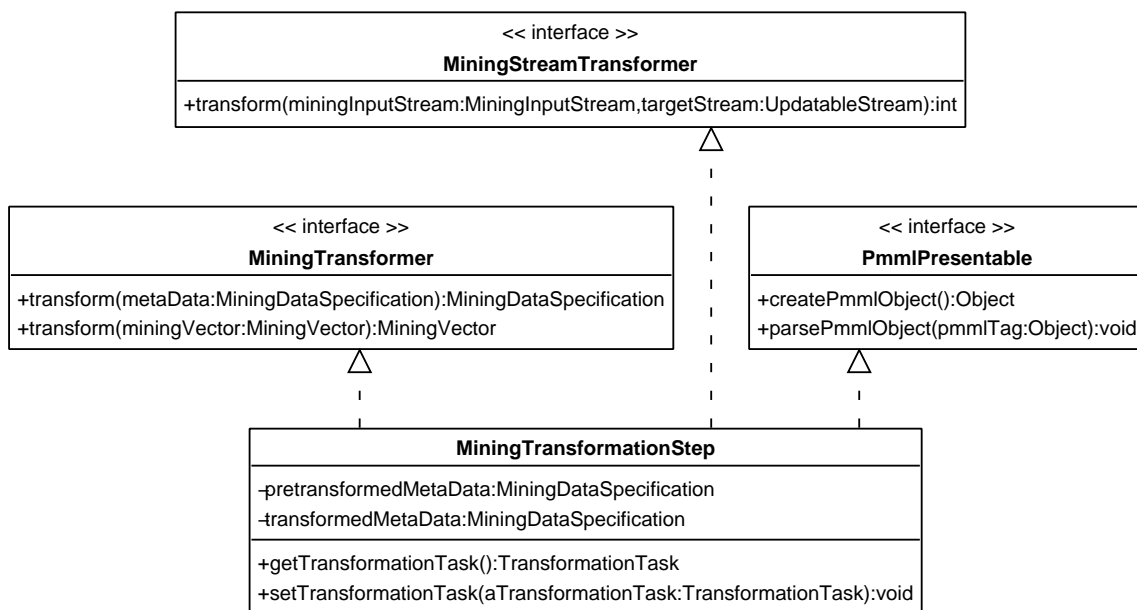


Figure 56: Crucial methods and variables of the `MiningTransformationStep` class.

Furthermore, it was discussed that the `MiningTransformationActivity` class is employed to compose transformation steps vertically. Its UML class diagram is shown in figure 57 on the next page. As opposed to the original XELOPES class, WUSAN's version of the class realizes a WusanML serialization.

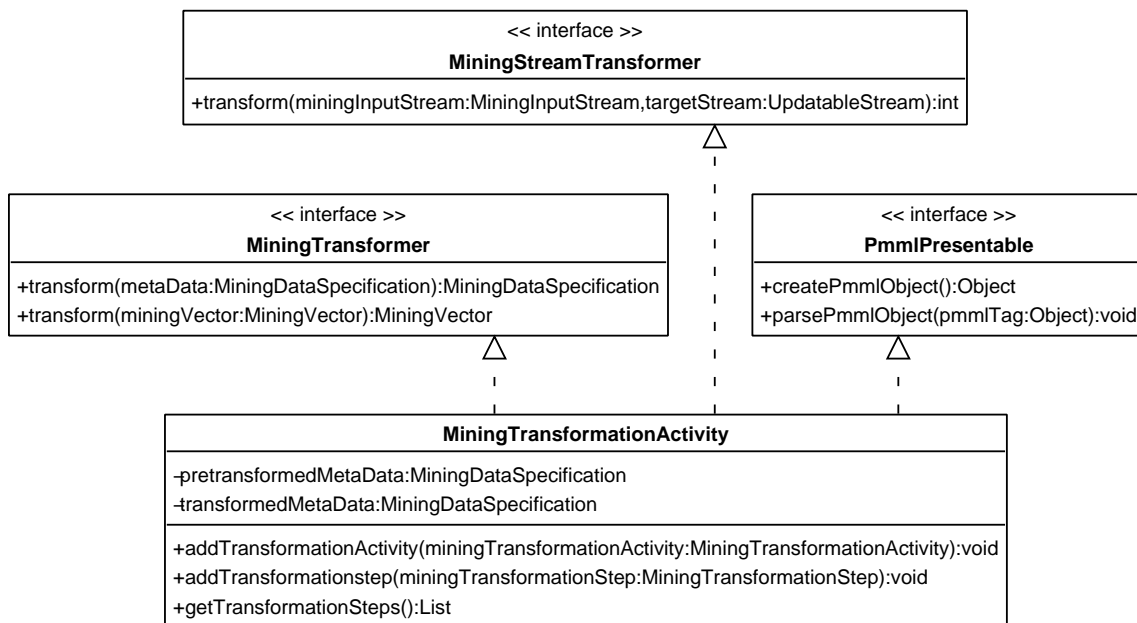


Figure 57: Methods and variables of the `MiningTransformationActivity` class.

Appendix F

WUSANML

WusanML is an XML-based markup language that has been created for this thesis to describe star schemas in order to make them persistent. The DTD that defines WusanML is depicted in listing F.1. Furthermore, WusanML can be regarded as a jumping-off point to create a GUI for modeling star schemas and assigned ETL transformations.

In this scenario, a GUI translates the graphical modeling of the WUA process into a standardized XML description, which is then imported into WUSAN to create the Java classes required to actually deploy the WUA process as it has been modeled. Although this scenario has not yet been implemented, WusanML establishes the basis for simplifying the practical application and deployment of the LOORDSM in real projects.

The DTD for WusanML in listing F.1 employs the PMML 2.0 `DataDictionary` tag (and all dependent tags). The `DataDictionary` tags and all other PMML 2.0 tags are discussed in detail in DMG-PMML.

Listing F.1: The DTD for WusanML.

```
<?xml version="1.0" encoding="UTF-8"?>
2
<!ELEMENT StarSchema (HybridDimension, (DegenerateDimension|
  ↳RegularDimension|HybridDimension)+)>
4
<!ELEMENT HybridDimension (TableStream)>
6
<!ATTLIST HybridDimension
  role CDATA #IMPLIED>
8
<!ELEMENT DegenerateDimension (DataDictionary, Transformation?,
  ↳VectorFilter?)>
10
<!ELEMENT RegularDimension (Transformation, TableStream, VectorFilter?)>
12
<!ATTLIST RegularDimension
  role CDATA #REQUIRED>
14
<!ELEMENT TableStream (MissingValueEncoding?)>
16
<!ATTLIST TableStream
  table CDATA #REQUIRED
  keyGenClass CDATA #REQUIRED
  insertCacheSize CDATA #REQUIRED
  selectCacheSize CDATA #REQUIRED
  batchSize CDATA #REQUIRED>
22
<!ELEMENT VectorFilter (AttributeFilter+)>
24
<!ATTLIST VectorFilter
  filterString CDATA #REQUIRED
  variableString CDATA #REQUIRED
  linkageExpression CDATA #IMPLIED>
26
28
<!ELEMENT AttributeFilter (FilterExpression*)>
30
<!ELEMENT FilterExpression (#PCDATA)>
```

```

32 <!ELEMENT MissingValueEncoding (MissingValue+)>
34 <!ELEMENT MissingValue (#PCDATA)>
36 <!ELEMENT Mapping (AttributeList, AttributeList, Mapping*)>
38 <!ATTLIST Mapping
40     className CDATA #REQUIRED
42     removeSourceAttributes (true|false) #REQUIRED
44     parameter CDATA #IMPLIED>
46 <!ELEMENT Step (Mapping+)>
48 <!ELEMENT Transformation (Step+)>
50 <!ELEMENT AttributeList (Name*)>
52 <!ELEMENT Name (#PCDATA)>
54 <!-- PMML tags -->
56 <!ELEMENT DataDictionary (DataField+) >
58 <!ATTLIST DataDictionary
60     numberOfFields CDATA #IMPLIED>
62 <!ELEMENT DataField ((Interval*|Value*))>
64 <!ATTLIST DataField
66     name CDATA #REQUIRED
68     displayName CDATA #IMPLIED
70     optype (categorical|ordinal|continuous) #REQUIRED
72     dataType (string|integer|float|double|boolean|datePrudsys) #\
74     →IMPLIED
76     isCyclic (0|1) "0">
78 <!ELEMENT Interval EMPTY>
80 <!ATTLIST Interval
82     closure (openClosed|openOpen|closedOpen|closedClosed) #REQUIRED
84     leftMargin CDATA #IMPLIED
86     rightMargin CDATA #IMPLIED>
88 <!ELEMENT Value EMPTY >
90 <!ATTLIST Value
92     value CDATA #REQUIRED
94     displayValue CDATA #IMPLIED
96     property (valid|invalid|missing|positive|negative) "valid">

```


Appendix G

MORE ON DEPLOYING THE LOORDSM

This appendix chapter ties in with chapter 5 on page 95 by elaborating the data warehouse and ETL modeling outlined in section 5.2 on page 101. Section G.1, discusses how the dimension tables referenced by the data marts in section G.2 on page 160 are modeled. Then, in section G.2 on page 160, modeling and populating the data marts with WusanML is discussed in detail, that is, all data marts required for the recommendation engine showcase of section 5.1 on page 95 and their ETL processes – tailored to the KDD Cup 2000 logs – are elaborated with WusanML. In section G.3 on page 170 various alternatives for tapping the data warehouse for data mining are discussed including a stream that is based on an MDX query. Finally, in section G.4 on page 172, making the data warehouse operational by creating the schema file mentioned for the second mapping in section 5.2.1 on page 101 is discussed in detail.

G.1 Creating and Populating Dimension Tables

This section refers to item v on page 103 and demonstrates how to model the dimension tables for the data warehouse with PMML. A *dimension table* refers to a database table that is referenced by other tables but itself contains no foreign key references. A dimension table's information hence does not depend on any other database table. As mentioned in section 5.2.2 on page 103, it must be pointed out that the KDD Cup 2000 data strongly designate the designs of the data marts in this chapter. Actually, the essential tasks discussed in that very section advise a different approach, that is, any conceptual tasks should be conducted driven by analytical requirements rather than merely driven by available attributes.¹ However, for this thesis, item ii on page 103 and item iii on page 103 were not feasible since an application server was not available.² On this account, all models of this chapter are driven by the data actually available.

G.1.1 Date Dimension Table

For the date dimension table, it is generally not known a priori what periods it covers, since it may be referenced by utterly different data marts with distinct application domains. For the data warehouse of the recommendation engine, the date dimension table is not populated a priori. Consequently, two building blocks are required for its modeling: first, meta-data in PMML format and second, a raw ETL transformation, which is employed as a building block for the actual ETL transformation.³

The relational structure of table 3 on the following page, which has been previously derived in a conceptual step according to item iv on page 103, must be transformed into meta-data conforming to definition 4.4 on page 64. In section 4.2.1 on page 61 it was discussed that the PMML's `DataDictionary` tag can be employed to describe meta-data conforming to definition 4.4 on page 64. Listing G.1 on the next page depicts the meta-data of the date

¹See item i on page 103.

²Compare section 5.1.1 on page 95.

³Compare the notion of a raw transformation in section 4.3.1 on page 83.

dimension, which is taken as input for the `createTable` method of the `MiningUpdateableSqlSource` class in figure 50 on page 137. Once the database table is created, its assigned ETL classes can be deployed.

Attribute Name	Attribute Type (CWM)	Example
Timestamp	Categorical	2006-01-01 01:23:45
Day	Integer	1
Month Label	Ordinal	January
Month Numeric	Integer	1
Year	Integer	2006
Weekday Label	Ordinal	Sunday
Weekday Numeric	Integer	7
Day of Year	Integer	1
Week of Month Label	Ordinal	Week 1
Week of Month Numeric	Integer	1
Week of Year Label	Ordinal	Week 1
Week of Year Numeric	Integer	1
Quarter Label	Ordinal	Quarter 1
Quarter Numeric	Integer	1

Table 3: Attributes of the date dimension.

Listing G.1: Meta-data of the date dimension table (modeled with PMML).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <PMML version="2.0">
3   <Header copyright="(c)_2005_by_Thilo_Maier" description="Meta-data_for_
4     _the_date_dimension">
5     <Application version="1.0" name="WUSAN" />
6     <Timestamp>2005-07-07 11:54:06</Timestamp>
7   </Header>
8   <DataDictionary numberOfFields="9">
9
10    <!-- Original Timestamp -->
11    <DataField name="Timestamp" optype="categorical" dataType="string"
12      />
13
14    <!-- Day -->
15    <DataField dataType="integer" name="Day" isCyclic="1" optype="
16      continuous">
17      <Interval leftMargin="1" rightMargin="31" closure="
18        closedClosed" />
19    </DataField>
20
21    <!-- Month Label -->
22    <DataField dataType="string" name="Month_Label" isCyclic="0"
23      optype="categorical">
24      <Value value="January" property="valid" />
25
26      <!-- Remaining months are modeled analogously. -->
27    </DataField>

```

```

25 <!-- Month Numeric -->
    <DataField name="Month_Numeric" optype="continuous" dataType="\
        →integer" isCyclic="0">
        <Interval closure="closedClosed" leftMargin="1" rightMargin="\
            →12" />
27 </DataField>

29 <!-- Year -->
    <DataField name="Year" dataType="integer" isCyclic="0" optype="\
        →continuous">
        <Interval leftMargin="-Infinity" rightMargin="Infinity" \
            →closure="openOpen" />
31 </DataField>

33 <!-- Weekday Label -->
    <DataField dataType="string" name="Weekday_Label" isCyclic="0" \
        →optype="categorical">
        <Value value="Monday" property="valid" />
37

    <!-- Remaining weekdays are modeled analogously. -->
39 </DataField>

41 <!-- Weekday Numeric -->
    <DataField name="Weekday_Numeric" optype="continuous" isCyclic="0" \
        → dataType="integer">
        <Interval closure="closedClosed" leftMargin="1" rightMargin="7" \
            → " />
43 </DataField>

45 <!-- Day of Year -->
    <DataField dataType="integer" name="Day_of_Year" isCyclic="1" \
        →optype="continuous">
        <Interval leftMargin="1" rightMargin="366" closure="\
            →closedClosed" />
49 </DataField>

51 <!-- Week of Month Label -->
    <DataField name="Week_of_Month_Label" optype="categorical" \
        →dataType="string">
        <Value value="Week_1" property="valid" />
53

    <!-- Remaining weeks are modeled analogously. -->
55 </DataField>

57 <!-- Week of Month Numeric -->
    <DataField dataType="integer" name="Week_of_Month_Numeric" \
        →isCyclic="0" optype="continuous">
        <Interval leftMargin="1" rightMargin="5" closure="closedClosed" \
            → " />
61 </DataField>

63 <!-- Week of Year Label -->
    <DataField name="Week_of_Year_Label" optype="categorical" dataType \
        →="string">
        <Value value="Week_1" property="valid" />
65

    <!-- Remaining weeks are modeled analogously. -->
67 </DataField>

```

```

69      <!-- Week of Year Numeric -->
71      <DataField dataType="integer" name="Week_of_Year_Numeric" isCyclic\
      →="1" optype="continuous">
          <Interval leftMargin="0" rightMargin="53" closure="\
          →closedClosed" />
73      </DataField>

75      <!-- Quarter Label -->
76      <DataField dataType="string" name="Quarter_Label" isCyclic="0" \
      →optype="ordinal">
77          <Value value="Quarter_1" property="valid" />

79      <!-- Remaining quarters are modeled analogously. -->
80      </DataField>

81      <!-- Quarter Numeric -->
82      <DataField name="Quarter_Numeric" optype="continuous" dataType="\
83      →integer" isCyclic="1">
          <Interval closure="closedClosed" leftMargin="1" rightMargin="4\
          →" />
85      </DataField>
86      </DataDictionary>
87 </PMML>

```

The raw ETL transformation in listing G.2 is required for modeling this dimension table with a `RegularDimension` class.⁴ Note that the target names of the raw ETL transformation in listing G.2 must match the attribute names in the above listing. Furthermore, if the dimension table is to be included in the ETL process of a certain data mart, the source attribute names of the raw ETL transformation must match the attribute names of the corresponding attributes of the ETL source stream.

Listing G.2: Raw ETL transformation for populating the date dimension table (WusanML).

```

1  <?xml version="1.0" encoding="ISO8859_1"?>
2  <Transformation>
3      <Step>
4
5          <!-- Surrounding multiple-to-multiple mapping -->
6          <Mapping className="com.prudsys.pdm.Transform.\
7          →MultipleToMultipleMapping" removeSourceAttributes="true">
8              <AttributeList />
9              <AttributeList />
10
11         <!-- Transform timestamp -->
12         <Mapping className="wusan.pdm.Transform.OneToMultiple.\
13         →TransformTimestamp" removeSourceAttributes="true" \
14         →parameter="yyyy-MM-dd">
15             <AttributeList>
16                 <Name>Date</Name>
17             </AttributeList>
18             <AttributeList>
19                 <Name>Timestamp</Name>

```

⁴In practice, dimension classes are not created directly. ETL transformations of dimensions must be placed within a WusanML description of a star schema, for example, in line 21 in listing G.12 on page 161. When a WusanML description is read by a `StarSchema` class, all required dimension classes are created automatically based on the information provided by the XML model.

```

17         <Name>Day</Name>
18         <Name>Month Label</Name>
19         <Name>Month Numeric</Name>
20         <Name>Year</Name>
21         <Name>Weekday Label</Name>
22         <Name>Weekday Numeric</Name>
23         <Name>Day of Year</Name>
24         <Name>Week of Month Label</Name>
25         <Name>Week of Month Numeric</Name>
26         <Name>Week of Year Label</Name>
27         <Name>Week of Year Numeric</Name>
28         <Name>Quarter Label</Name>
29         <Name>Quarter Numeric</Name>
30         <Name>Hour</Name>
31         <Name>Minute</Name>
32         <Name>Second</Name>
33         <Name>Time of Day Label</Name>
34         <Name>Time of Day Numeric</Name>
35     </AttributeList>
36 </Mapping>
37 </Mapping>
38 </Step>
39 <Step>
40
41     <!-- Remove unneeded attributes -->
42     <Mapping className="com.prudsys.pdm.Transform.MultipleToMultiple.\
43     →RemoveAttributes" removeSourceAttributes="true">
44         <AttributeList />
45         <AttributeList>
46             <Name>Hour</Name>
47             <Name>Minute</Name>
48             <Name>Second</Name>
49             <Name>Time of Day Label</Name>
50             <Name>Time of Day Numeric</Name>
51         </AttributeList>
52     </Mapping>
53 </Step>
54 </Transformation>

```

G.1.2 Time Dimension Table

The time dimension table is similar to the date dimension table of the previous section. Hence, the discussion of the preceding section also applies to this dimension table. The attributes of the time dimension gathered in a conceptual step are shown in table 4 on the following page. The two resulting building blocks for the time dimension table are depicted in listing G.3 and in listing G.4 on page 155. The raw ETL transformation is quite similar to that of the date dimension table, as only the parsing pattern in line 11 in listing G.4 on page 155 is altered and another set of attributes is chosen in the last transformation step.

Listing G.3: Meta-data of the time dimension table (modeled with PMML).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <PMML version="2.0">
3     <Header copyright="(c)_2005_by_Thilo_Maier" description="Meta-data_for_
4     →_the_time_dimension">
5         <Application version="1.0" name="WUSAN" />

```

Attribute Name	Attribute Type (CWM)	Example
Timestamp	Categorical	2006-01-01 01:23:45
Hour	Integer	1
Minute	Integer	23
Second	Integer	45
Time of Day Label	Ordinal	Late Night
Time of Day Numeric	Integer	1

Table 4: Attributes of the time dimension.

```

5      <Timestamp>2005-07-07 22:30:06</Timestamp>
</Header>
7      <DataDictionary numberOfFields="5">
9
10     <!-- Original Timestamp -->
<DataField name="Timestamp" optype="categorical" dataType="string"
11     → />
12
13     <!-- Hour -->
<DataField dataType="integer" name="Hour" isCyclic="1" optype="
14     →continuous">
<Interval leftMargin="0" rightMargin="23" closure="
15     →closedClosed" />
</DataField>
16
17     <!-- Minute -->
<DataField dataType="integer" name="Minute" isCyclic="1" optype="
18     →continuous">
<Interval leftMargin="0" rightMargin="59" closure="
19     →closedClosed" />
</DataField>
20
21     <!-- Second -->
<DataField dataType="integer" name="Second" isCyclic="1" optype="
22     →continuous">
<Interval leftMargin="0" rightMargin="59" closure="
23     →closedClosed" />
</DataField>
24
25     <!-- Time of Day Label -->
<DataField dataType="string" name="Time_of_Day_Label" isCyclic="1"
26     → optype="ordinal">
<Value displayValue="late_night" property="valid" value="late_
27     →night" />
<Value displayValue="daybreak" property="valid" value="
28     →daybreak" />
<Value displayValue="morning" property="valid" value="morning"
29     → />
<Value displayValue="lunchtime" property="valid" value="
30     →lunchtime" />
<Value displayValue="afternoon" property="valid" value="
31     →afternoon" />
<Value displayValue="evening" property="valid" value="evening"
32     → />
<Value displayValue="night" property="valid" value="night" />
33
34
35

```

```

37     </DataField>
39     <!-- Time of Day Numeric -->
41     <DataField dataType="integer" name="Time_of_Day_Numeric" isCyclic=
43     →"1" optype="continuous">
        <Interval closure="closedClosed" leftMargin="1" rightMargin="7"
        →" />
    </DataField>
</DataDictionary>
</PMML>

```

Listing G.4: Raw ETL transformation for populating the time dimension table (WusanML).

```

1 <?xml version="1.0" encoding="ISO8859_1"?>
<Transformation>
3   <Step>
5       <!-- Surrounding multiple-to-multiple mapping -->
6       <Mapping className="com.prudsys.pdm.Transform.
7       →MultipleToMultipleMapping" removeSourceAttributes="true">
8           <AttributeList />
9           <AttributeList />
11          <!-- Transform timestamp -->
12          <Mapping className="wusan.pdm.Transform.OneToMultiple.
13          →TransformTimestamp" removeSourceAttributes="true"
14          →parameter="hh:mm:ss">
15              <AttributeList>
16                  <Name>Time</Name>
17              </AttributeList>
18              <AttributeList>
19                  <!-- Same target attribute names as in
20                  listing G.2 on page 152. -->
21              </AttributeList>
22          </Mapping>
23      </Mapping>
24  </Step>
25  <Step>
26      <!-- Remove unneeded attributes -->
27      <Mapping className="com.prudsys.pdm.Transform.MultipleToMultiple.
28      →RemoveAttributes" removeSourceAttributes="true">
29          <AttributeList />
30          <AttributeList>
31              <Name>Day</Name>
32              <Name>Month Label</Name>
33              <Name>Month Numeric</Name>
34              <Name>Year</Name>
35              <Name>Weekday Label</Name>
36              <Name>Weekday Numeric</Name>
37              <Name>Day of Year</Name>
38              <Name>Week of Month Label</Name>
39              <Name>Week of Month Numeric</Name>
40              <Name>Week of Year Label</Name>
41              <Name>Week of Year Numeric</Name>
42              <Name>Quarter Label</Name>
43              <Name>Quarter Numeric</Name>

```

```

43     </AttributeList>
    </Mapping>
  </Step>
45 </Transformation>

```

G.1.3 Referrer Dimension Table

The referrer dimension table is employed to store the components of a referring URL. As there exist countless URLs, it is not feasible to populate this dimension table a priori. Hence, it is included in the ETL process of the clickstream and session marts in section 5.2.3.3 on page 106. The meta-data of the referrer dimension table are depicted in listing G.5 in PMML format and its assigned raw ETL transformation is shown in listing G.6.⁵

Listing G.5: Meta-data of the referrer dimension table (modeled with PMML).

```

1 <?xml version="1.0" encoding="UTF-8"?>
  <PMML version="2.0">
3     <Header copyright="(c)_2005_by_Thilo_Maier" description="Meta-data_for_
      _the_referrer_dimension">
        <Application version="1.0" name="WUSAN" />
5         <Timestamp>2005-07-09 10:31:12 CET</Timestamp>
      </Header>
7     <DataDictionary numberOfFields="5">
9         <!-- protocol -->
        <DataField name="protocol" optype="categorical" dataType="string" \
          />
11
        <!-- host name -->
13        <DataField name="host_name" optype="categorical" dataType="string" \
          />
15
        <!-- path -->
        <DataField name="path" optype="categorical" dataType="string" />
17
        <!-- country -->
19        <DataField name="country" optype="categorical" dataType="string" / \
          />
21
        <!-- IP address -->
        <DataField name="IP_address" optype="categorical" dataType="string \
          " />
23    </DataDictionary>
  </PMML>

```

Listing G.6: Raw ETL transformation for populating the referrer dimension table (WusanML).

```

1 <?xml version="1.0" encoding="ISO8859_1"?>
  <Transformation>
2     <Step>
4

```

⁵Many attributes could be added to this dimension table, if the employed `SplitUrl` mapping were extended. The current version of this mapping is restricted to such attributes that can be computed with available Java classes. Furthermore, this mapping is a bottleneck within the ETL process, since the conversion of a URL into an IP address, or vice versa, requires accessing a DNS server and necessitates a network round trip for every processed vector (alleviated through caching).


```

6      <!-- Surrounding multiple-to-multiple mapping -->
7      <Mapping className="com.prudsys.pdm.Transform.\
      →MultipleToMultipleMapping" removeSourceAttributes="true">
8          <AttributeList />
9          <AttributeList />
10
11         <!-- Split URL -->
12         <Mapping className="wusan.pdm.Transform.OneToMultiple.SplitUrl\
      →" removeSourceAttributes="true">
13             <AttributeList>
14                 <Name>URL</Name>
15             </AttributeList>
16             <AttributeList>
17                 <Name>protocol</Name>
18                 <Name>host name</Name>
19                 <Name>country</Name>
20                 <Name>path</Name>
21                 <Name>query</Name>
22                 <Name>IP address</Name>
23             </AttributeList>
24         </Mapping>
25     </Mapping>
26 </Step>
27 <Step>
28
29     <!-- Remove unneeded attributes -->
30     <Mapping className="com.prudsys.pdm.Transform.MultipleToMultiple.\
      →RemoveAttributes" removeSourceAttributes="true">
31         <AttributeList />
32         <AttributeList>
33             <Name>query</Name>
34         </AttributeList>
35     </Mapping>
36 </Step>
</Transformation>

```

Remark. When the referrer dimension table is created, there is one aspect that must be taken into account. In section D.6 on page 133, it is stated that a categorical attribute is mapped to VARCHAR(100) in the database table by default. However, the `path` attribute in line 16 of listing G.5 on the preceding page may take values of length greater than 100 characters. The solution to this problem is shown in line 3 in listing G.7, following the approach sketched in section D.6 on page 133.

Listing G.7: Creating the referrer dimension table (Java).

```

// create data source
2 MiningUpdatableSqlSource dataSource = new MiningUpdatableSqlSource("db2-\
    →wusan.prp", "WUSAN");
dataSource.addKeywordForLColumn("path");
4
// load meta-data for referrer dimension
6 MiningDataSpecification metaData = new MiningDataSpecification();
try {
8     metaData.readPmml(new BufferedReader(new FileReader("resources/\
    →metadata/dimensions/referrer_dimension.xml")));
} catch (FileNotFoundException e) {
10     throw new MiningException(e.getMessage());

```

```

}
12
// create table for referrer dimension
14 dataSource.createTable("REFERRER_DIMENSION", metaData);

```

G.1.4 Assortment Dimension Table

As opposed to the date, time, and referrer dimension tables in the previous sections, all of the records for the assortment dimension table are determined a priori. Hence, it can be entirely populated without integration into the ETL process of a data mart. To this end, in listing G.8, a star schema comprising only one degenerate dimension is modeled. As all required attributes are available in the source stream, no raw ETL transformation is included, which means that in equation (4.13) on page 85, the raw ETL transformation $T_{\hat{A}_1, \dots, \hat{A}_\ell}$ is an identical mapping and the ETL transformation $T_{\tilde{A}_1, \dots, \tilde{A}_\ell}$ corresponds to the projection Π_{A_1, \dots, A_m} .

This ETL model is a good example for the pitfall mentioned in footnote 32 on page 104: the notions of dimensional modeling on the data storage layer in figure 16 on page 60 are not congruent to the corresponding notions of the ETL layer. In listing G.8, the assortment dimension table is treated as a fact table (in terms of the LOORDSM) and the star schema itself is *degenerate* as it comprises one degenerate dimension only.

Listing G.8: Star schema for populating the assortment dimension table (WusanML).

```

<?xml version="1.0" encoding="ISO8859_1"?>
2 <StarSchema>
4
  <!-- Fact Table -->
  <HybridDimension role="assortment">
6     <TableStream selectCacheSize="2000" table="ASSORTMENT_DIMENSION" \
      →insertCacheSize="2000" batchSize="1000" keyGenClass="wusan.\
      →pdm.Input.Relational.TrivialPrimaryKey">
8         <MissingValueEncoding>
9             <MissingValue>null</MissingValue>
10            <MissingValue>?</MissingValue>
11        </MissingValueEncoding>
12    </TableStream>
13 </HybridDimension>
14
  <!-- Attribute Selections -->
  <DegenerateDimension>
16     <DataDictionary numberOfFields="6">
17         <DataField name="Assortment_ID" optype="categorical" dataType=\
18            →"string" />
19         <DataField name="Assortment_Type" optype="categorical" \
20            →dataType="string" />
21         <DataField name="Assortment_Level" optype="continuous" \
22            →dataType="double" isCyclic="0">
23             <Interval closure="openOpen" leftMargin="-Infinity" \
24                →rightMargin="Infinity" />
25         </DataField>
26         <DataField name="Assortment_Level_2_Path" optype="categorical" \
27            → dataType="string" />
28         <DataField name="Assortment_Level_3_Path" optype="categorical" \
29            → dataType="string" />
30         <DataField name="Assortment_Level_4_Path" optype="categorical" \
31            → dataType="string" />

```

```

        </DataDictionary>
    </DegenerateDimension>
</StarSchema>

```

Remark. It is not necessary to create the assortment dimension table separately before initializing the star schema. As mentioned in footnote 56 on page 90, the fact table of a star schema is created automatically if it does not exist. Hence, only the following Java code must be executed in order to create and populate the assortment dimension table:

Listing G.9: Java code required for creating and populating the assortment dimension table.

```

1 // create data source
MiningUpdatableSqlSource dataSource = new MiningUpdatableSqlSource("db2-
    wusan.prp", "WUSAN");
3
// create reader for star schema description
5 BufferedReader reader = new BufferedReader(new FileReader("resources/
    StarSchemas/AssortmentDimension.xml"));
7
StarSchema assortmentDimension = StarSchema.createInstance(reader,
    dataSource);
9
// create source stream
MiningC50Stream sourceStream = new MiningC50Stream("C:\\KddCup\\
    questionland2.data", "resources/metadata/C50/questionland2.names");
11
// create ETL transformation
13 assortmentDimension.createEtlTransformation(sourceStream.getMetaData());
15
// execute ETL process
assortmentDimension.etl(sourceStream);

```

G.1.5 Content Dimension Table

This dimension table is modeled analogously to the assortment dimension table in the previous section. The corresponding ETL model is depicted in listing G.10. Executing the ETL process in Java can be accomplished analogously as shown in listing G.9.

Listing G.10: Star schema for populating the content dimension table (WusanML).

```

<?xml version="1.0" encoding="ISO8859_1"?>
2 <StarSchema>
4
    <!-- Fact Table -->
    <!-- Insert fact table of line 5 in listing G.8 on the facing page here.
6         Set table="CONTENT_DIMENSION" and role="content". -->
8
    <!-- Attribute Selections -->
    <DegenerateDimension>
10        <DataDictionary numberOfFields="3">
            <DataField name="Content_ID" optype="categorical" dataType="
                string" />
12        <DataField name="Content_Level_2_Path" optype="categorical"
            <DataField name="Content_Level_3_Path" optype="categorical"
            </DataField>
            </DataDictionary>
        </DegenerateDimension>
    </StarSchema>

```

```

14     </DataDictionary>
15     </DegenerateDimension>
16 </StarSchema>

```

G.1.6 Product Dimension Table

This dimension table is modeled analogously to the assortment and content dimension tables in the previous sections. The corresponding ETL model is depicted in listing G.11. Executing ETL in Java can be accomplished analogously as shown in listing G.9 on the preceding page.

Listing G.11: Star schema for populating the product dimension table (WusanML).

```

<?xml version="1.0" encoding="ISO8859_1"?>
2 <StarSchema>
4     <!-- Fact Table -->
5     <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
6         Set table="PRODUCT_DIMENSION" and role="product". -->
8     <!-- Attribute Selections -->
9     <DegenerateDimension>
10        <DataDictionary numberOfFields="20">
11            <DataField name="Product_Object_ID" optype="categorical" \
12                <-dataType="string" />
13            <DataField name="Product_Family_ID" optype="categorical" \
14                <-dataType="string" />
15            <DataField name="Product_Object_Status" optype="categorical" \
16                <-dataType="string" />
17            <DataField name="BrandName" optype="categorical" dataType="\
18                <-string" />
19
20        <!-- Remaining product properties are modeled analogously. -->
21        </DataDictionary>
22    </DegenerateDimension>
23 </StarSchema>

```

G.2 Creating and Populating Data Marts

This section refers to item vi on page 103 and demonstrates how to model data marts and their ETL process. All data marts required for the recommendation engine showcase of section 5.1 on page 95 are elaborated in detail completing the discussion of section 5.2.3 on page 104.

G.2.1 Customer Mart

In section 5.1.4.2 on page 100 it was mentioned that maintaining customer segments is not part of the core WUA process. It can even be stated that maintaining a customer mart is an activity in its own right, which should be administered with a long-term perspective within the overall corporate CRM strategy. Hence, the customer mart proposed in this section is not realistic due to the fact that it is solely fueled from the behavioral Web usage data available in the KDD Cup 2000 data disregarding any other potentially accessible customer data (for example, data from other customer touchpoints). However, in order to realize the recommendation strategy for known users sketched in section 5.1.4.2 on page 100, the customer mart is indispensable.

Figure 58 illustrates the actual foreign key references of the customer mart in the RDBMS. The solid lines imply that the referenced tables participate in the ETL process initiated by an instance of the `StarSchema` class with the fact table `CUSTOMER_FACT`.

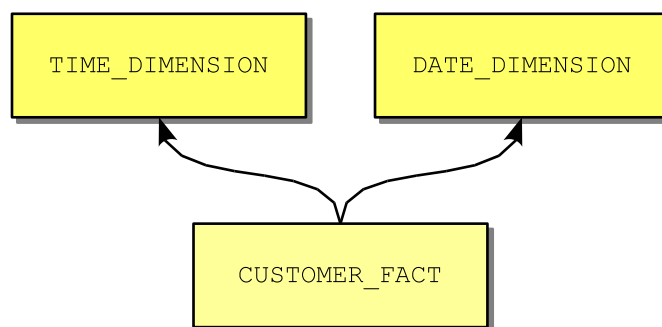


Figure 58: Foreign key references of the customer mart.

Listing G.12 illustrates how the ETL process for the customer mart is modeled. It consists of a `HybridDimension`, which models the fact table, two `DegenerateDimensions`, each of which is employed to extract various attributes from the source stream (compare section G.1.4 on page 158), and two `RegularDimensions`, each of which hosts an ETL transformation. The arrangement of the attributes within the degenerate dimensions and the arrangement of the dimensions themselves define the layout of the fact table.

Basically, attribute arrangement is not relevant, except for the `Customer ID` attribute, which must be placed as the first attribute, since it represents a natural primary key for the fact table taken as input for the `TrivialPrimaryKey` mapping. Given the model of the customer mart in listing G.12, again, executing its ETL process is a simple task and works analogously as illustrated in listing G.9 on page 159. As all referenced database tables are included in the ETL process, this model corresponds to the instantaneous ETL approach.⁶

Listing G.12: Star schema for populating the customer mart (WusanML).

```

1 <?xml version="1.0" encoding="ISO8859_1"?>
  <StarSchema>
3
4   <!-- Fact Table -->
5   <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
6       Set table="CUSTOMER_FACT" and role="customer". -->
7
8   <!-- Attribute Selections -->
9   <DegenerateDimension>
10      <DataDictionary numberOfFields="46">
11         <DataField name="Customer_ID" optype="categorical" dataType="
12             →string" />
13         <DataField name="WhichDoYouWearMostFrequent" optype="
14             →categorical" dataType="string" />
15
16         <!-- Remaining attributes are modeled analogously. -->
17      </DataDictionary>
18   </DegenerateDimension>
19
20   <!-- Date Dimension -->
21   <RegularDimension role="Account_Creation_Date">
  
```

⁶Compare item 1 on page 91.

```

21      <!-- Insert raw ETL transformation of listing G.2 on page 152 here with \
        → source attribute "Account_Creation_Date" in line 13. -->
        <TableStream selectCacheSize="2000" table="DATE_DIMENSION" \
        →insertCacheSize="2000" batchSize="1000" keyGenClass="wusan.\
        →pdm.Input.Relational.TrivialPrimaryKey">
23          <MissingValueEncoding>
            <MissingValue>null</MissingValue>
25            <MissingValue>?</MissingValue>
          </MissingValueEncoding>
27        </TableStream>
      </RegularDimension>

29
30      <!-- Time Dimension -->
31      <RegularDimension role="Account_Creation_Time">

33        <!-- Insert raw ETL transformation of listing G.4 on page 155 here with \
        → source attribute "Account_Creation_Time" in line 13. -->
        <TableStream selectCacheSize="2000" table="TIME_DIMENSION" \
        →insertCacheSize="2000" batchSize="1000" keyGenClass="wusan.\
        →pdm.Input.Relational.TrivialPrimaryKey">
35          <MissingValueEncoding>
            <MissingValue>null</MissingValue>
37            <MissingValue>?</MissingValue>
          </MissingValueEncoding>
39        </TableStream>
      </RegularDimension>
41 </StarSchema>

```

G.2.2 Order Mart

This section complements section 5.2.3.1 on page 104. Listing G.13 depicts the ETL modeling for the order mart. To assure referential integrity for the dashed lines, both corresponding columns of the `ORDER_FACT` table may not contain any `NULL` values.⁷ The `ReplaceNullValues` mapping replaces any `NULL` values with the primary key string `ReplaceNullValues.NULL_STRING`, which references a vector containing `NULL` values only.⁸

Listing G.13: Star schema for populating the order mart (WusanML).

```

1 <?xml version="1.0" encoding="ISO8859_1"?>
  <StarSchema>
3
4     <!-- Fact Table -->
5     <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
        Set table="ORDER_FACT" and role="order". -->
7
8     <!-- Attribute Selections -->
9     <DegenerateDimension>
        <DataDictionary numberOfFields="9">

```

⁷Referring to a dimension with a `NULL` value leads to indeterminate results if the logical multi-dimensional model based on such database tables is queried with MDX in Mondrian. Best is to ensure that foreign key attributes always reference valid vectors in the dimension table.

⁸Compare remark item 2 on page 92. Note that this approach of assuring referential integrity only works if the `MiningTableSqlStream` of the referenced dimension table employs the `TrivialPrimaryKey` class to generate its primary keys. This class simply takes over an existing key candidate attribute, for example, an ID.

```

11     <DataField name="Order_ID" optype="categorical" dataType="\
        →string" />
12     <DataField name="Order_Discount_Amount" optype="continuous" \
        →dataType="double" isCyclic="0">
13         <Interval closure="openOpen" leftMargin="-Infinity" \
            →rightMargin="Infinity" />
14     </DataField>
15     <DataField name="Order_Promotion_Code" optype="categorical" \
        →dataType="string" />
16     <DataField name="Order_Source" optype="categorical" dataType="\
        →string" />
17     <DataField name="Order_Status" optype="categorical" dataType="\
        →string" />
18     <DataField name="Order_Amount" optype="continuous" dataType="\
        →double" isCyclic="0">
19         <Interval closure="openOpen" leftMargin="-Infinity" \
            →rightMargin="Infinity" />
20     </DataField>
21     <DataField name="Order_Shipping_Amount" optype="continuous" \
        →dataType="double" isCyclic="0">
22         <Interval closure="openOpen" leftMargin="-Infinity" \
            →rightMargin="Infinity" />
23     </DataField>
24     <DataField name="Order_Tax_Amount" optype="continuous" \
        →dataType="double" isCyclic="0">
25         <Interval closure="openOpen" leftMargin="-Infinity" \
            →rightMargin="Infinity" />
26     </DataField>
27     <DataField name="Order_Credit_Card_Brand" optype="categorical" \
        → dataType="string" />
28 </DataDictionary>
29 </DegenerateDimension>
30
31 <!-- Assure Referential Integrity -->
32 <DegenerateDimension>
33     <DataDictionary numberOfFields="2">
34         <DataField name="Order_Customer_ID" optype="categorical" \
            →dataType="string" />
35         <DataField name="Order_Session_ID" optype="categorical" \
            →dataType="string" />
36     </DataDictionary>
37     <Transformation>
38         <Step>
39             <Mapping className="com.prudsys.pdm.Transform.\
                →MultipleToMultipleMapping" removeSourceAttributes="\
                →true">
40                 <AttributeList />
41                 <AttributeList />
42                 <Mapping className="wusan.pdm.Transform.OneToOne.\
                    →ReplaceNullValues" removeSourceAttributes="true">
43                     <AttributeList>
44                         <Name>Order Customer ID</Name>
45                     </AttributeList>
46                     <AttributeList>
47                         <Name>Order Customer ID</Name>
48                     </AttributeList>
49                 </Mapping>

```

```

51         <Mapping className="wusan.pdm.Transform.OneToOne.\
        →ReplaceNullValues" removeSourceAttributes="true">
        <AttributeList>
53             <Name>Order Session ID</Name>
        </AttributeList>
        <AttributeList>
55             <Name>Order Session ID</Name>
        </AttributeList>
57     </Mapping>
    </Mapping>
59 </Step>
</Transformation>
61 </DegenerateDimension>

63 <!-- Date Dimension -->
<RegularDimension role="Order_Date">
65     <!-- Modeled analogously as before in line 19
67         of listing G.12 on page 161. -->
</RegularDimension>
69

71 <!-- Time Dimension -->
<RegularDimension role="Order_Time">
73     <!-- Modeled analogously as before in line 31
75         of listing G.12 on page 161. -->
</RegularDimension>
</StarSchema>

```

G.2.3 Order Line Mart

This section complements section 5.2.3.2 on page 106. Listing G.14 depicts the ETL modeling for the order line mart, which is similar to the order mart discussed in the previous section.

Listing G.14: Star schema for populating the order line mart (WusanML).

```

<?xml version="1.0" encoding="ISO8859_1"?>
2 <StarSchema>
4     <!-- Fact Table -->
5     <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
6         Set table="ORDER_LINE_FACT" and role="order_line". -->
8     <!-- Attribute Selections -->
    <DegenerateDimension>
10     <DataDictionary numberOfFields="7">
        <DataField name="Order_Line_ID" optype="categorical" dataType=↵
        →"string" />
12     <DataField name="Order_Line_Unit_List_Price" optype="↵
        →continuous" dataType="double" isCyclic="0">
        <Interval closure="openOpen" leftMargin="-Infinity" ↵
        →rightMargin="Infinity" />
14     </DataField>
        <DataField name="Order_Line_Quantity" optype="continuous" ↵
        →dataType="integer" isCyclic="0">
        <Interval closure="closedOpen" leftMargin="0" rightMargin=↵
        →"Infinity" />
16

```



```

18     </DataField>
19     <DataField name="Order_Line_Unit_Sale_Price" optype="\
    →continuous" dataType="double" isCyclic="0">
20         <Interval closure="openOpen" leftMargin="-Infinity" \
    →rightMargin="Infinity" />
21     </DataField>
22     <DataField name="Order_Line_Status" optype="categorical" \
    →dataType="string" />
23     <DataField name="Order_Line_Tax_Amount" optype="continuous" \
    →dataType="double" isCyclic="0">
24         <Interval closure="openOpen" leftMargin="-Infinity" \
    →rightMargin="Infinity" />
25     </DataField>
26     <DataField name="Order_Line_Amount" optype="continuous" \
    →dataType="double" isCyclic="0">
27         <Interval closure="openOpen" leftMargin="-Infinity" \
    →rightMargin="Infinity" />
28     </DataField>
29 </DataDictionary>
30 </DegenerateDimension>
31
32 <!-- Assure Referential Integrity -->
33 <DegenerateDimension>
34     <DataDictionary numberOfFields="5">
35         <DataField name="Order_ID" optype="categorical" dataType="\
    →string" />
36         <DataField name="Order_Line_Session_ID" optype="categorical" \
    →dataType="string" />
37         <DataField name="Product_ID" optype="categorical" dataType="\
    →string" />
38         <DataField name="Order_Line_Assortment_ID" optype="categorical\
    →" dataType="string" />
39         <DataField name="Order_Line_Subassortment_ID" optype="\
    →categorical" dataType="string" />
40     </DataDictionary>
41
42     <!-- Transformation as before in line 37
    of listing G.13 on page 162 with one ReplaceNullValues mapping for\
    → each attribute. -->
43 </DegenerateDimension>
44
45 <!-- Date Dimension -->
46 <RegularDimension role="Order_Line_Date">
47
48     <!-- Modeled analogously as before in line 19
    of listing G.12 on page 161. -->
49 </RegularDimension>
50
51 <!-- Time Dimension -->
52 <RegularDimension role="Order_Line_Time">
53
54     <!-- Modeled analogously as before in line 31
    of listing G.12 on page 161. -->
55 </RegularDimension>
56
57 </StarSchema>

```

G.2.4 Clickstream and Session Marts

This section complements section 5.2.3.3 on page 106. Listing G.15 depicts the ETL modeling for the session mart, which is similar to the marts discussed in the previous sections.

Listing G.15: Star schema for populating the session mart (WusanML).

```

2 <StarSchema>
4 <!-- Fact Table -->
4 <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
6     Set table="SESSION_FACT" and role="Session". -->
6
8 <!-- Attribute Selections -->
8 <DegenerateDimension>
10     <DataDictionary numberOfFields="4">
10         <DataField name="Session_ID" optype="categorical" dataType="\
12             →string" />
12         <DataField name="Session_Cookie_ID" optype="categorical" \
14             →dataType="string" />
12         <DataField name="Session_Visit_Count" optype="continuous" \
14             →dataType="integer" isCyclic="0">
14             <Interval closure="closedOpen" leftMargin="0" rightMargin=\
16                 →"Infinity" />
14         </DataField>
14         <DataField name="Session_First_Processing_Time" optype="\
16             →continuous" dataType="double" isCyclic="0">
16             <Interval closure="closedOpen" leftMargin="0" rightMargin=\
18                 →"Infinity" />
16         </DataField>
18     </DataDictionary>
20 </DegenerateDimension>
20
22 <!-- User Agent Degenerate Dimension -->
22 <DegenerateDimension>
24
24     <!-- Data Dictionary -->
24     <DataDictionary numberOfFields="4">
26         <DataField name="User_Agent_Name" optype="categorical" \
26             →dataType="string" />
26         <DataField name="User_Agent_Version" optype="categorical" \
28             →dataType="string" />
28         <DataField name="User_Agent_OS" optype="categorical" dataType=\
28             →"string" />
28         <DataField name="Session_User_Agent" optype="categorical" \
30             →dataType="string" />
30     </DataDictionary>
30
32 <!-- ETL Transformation -->
32 <Transformation>
34     <Step>
34         <Mapping className="com.prudsys.pdm.Transform.\
36             →MultipleToMultipleMapping" removeSourceAttributes="\
36             →false">
36             <AttributeList />
36             <AttributeList />
38         <Mapping className="wusan.pdm.Transform.OneToMultiple.\
38             →SplitUserAgent" removeSourceAttributes="false">
38             <AttributeList>

```

```

40         <Name>Session User Agent</Name>
41     </AttributeList>
42     <AttributeList>
43         <Name>User Agent Name</Name>
44         <Name>User Agent Version</Name>
45         <Name>User Agent OS</Name>
46     </AttributeList>
47 </Mapping>
48 </Mapping>
49 </Step>
50 </Transformation>
51
52 <!-- Vector Filter -->
53 <VectorFilter variableString="x" filterString="f">
54     <AttributeFilter />
55     <AttributeFilter />
56     <AttributeFilter>
57         <FilterExpression>bot</FilterExpression>
58     </AttributeFilter>
59 </VectorFilter>
60 </DegenerateDimension>
61
62 <!-- Assure Referential Integrity -->
63 <DegenerateDimension>
64     <DataDictionary numberOfFields="2">
65         <DataField name="Session_First_Content_ID" optype="categorical"
66             →" dataType="string" />
67         <DataField name="Session_Customer_ID" optype="categorical"
68             →dataType="string" />
69     </DataDictionary>
70
71     <!-- Transformation as before in line 37 of listing G.13 on page 162
72         →with one ReplaceNullValues mapping for each attribute. -->
73 </DegenerateDimension>
74
75 <!-- Referrer Dimension -->
76 <RegularDimension role="Session_First_Referrer">
77
78     <!-- Insert raw ETL transformation of listing G.6 on page 156 here
79         →with source attribute "Session_First_Referrer" in line 13. -->
80
81     <TableStream selectCacheSize="2000" table="REFERRER_DIMENSION"
82         →insertCacheSize="2000" batchSize="1000" keyGenClass="wusan.
83         →pdm.Input.Relational.MD5PrimaryKey">
84         <MissingValueEncoding>
85             <MissingValue>null</MissingValue>
86             <MissingValue>?</MissingValue>
87         </MissingValueEncoding>
88     </TableStream>
89 </RegularDimension>
90
91 <!-- Date Dimension -->
92 <RegularDimension role="Session_First_Request_Date">
93
94     <!-- Modeled analogously as before in line 19
95         of listing G.12 on page 161. -->
96 </RegularDimension>

```

```

92 <!-- Time Dimension -->
    <RegularDimension role="Session_First_Request_Time">
94
        <!-- Modeled analogously as before in line 31
96             of listing G.12 on page 161. -->
    </RegularDimension>
98 </StarSchema>

```

Contrary to the data marts modeled previously, the above listing makes use of a vector filter in the user agent degenerate dimension in line 52. First, not only does this degenerate dimension select an attribute from the source file, it also includes a transformation that analyzes the Session User Agent attribute and derives the three target attributes User Agent Name, User Agent Version, and User Agent OS. Second, it includes a vector filter conforming to section D.7 on page 137, which, if it matches, prevents the current transformed vector from being inserted into the fact table.

Here, the transformation sets the User Agent OS to “bot” if it recognizes a Web robot. This means that a session is not inserted into the fact table if initiated by a Web robot. This combination of a suitable transformation and a vector filter directly addresses the open issue in item 2 on page 40 for Web robots that reveal their identity.⁹ Principally, a vector filter can be added to any dimension; as soon as one vector filter matches, no vector is inserted into the fact table. This feature makes the LOORDSM even more powerful in view of the preprocessing phase, since the filtering approach can be extended to nested star schemas.

Listing G.16: Star schema for populating the clickstream mart (WusanML).

```

<StarSchema>
2
    <!-- Fact Table -->
4    <!-- Insert fact table of line 5 in listing G.8 on page 158 here.
        Set table="CLICKSTREAM_FACT", role="clickstream", and keyGenClass
        →="wusan.pdm.Input.Relational.MD5PrimaryKey". -->
6
    <!-- Attribute Selections -->
8    <DegenerateDimension>
        <DataDictionary numberOfFields="3">
10        <DataField name="Request_Processing_Time" optype="continuous" \
            →dataType="integer" isCyclic="0">
            <Interval closure="closedOpen" leftMargin="0" rightMargin=\
            →"Infinity" />
12        </DataField>
        <DataField name="Request_Sequence" optype="continuous" \
            →dataType="integer" isCyclic="0">
            <Interval closure="closedOpen" leftMargin="0" rightMargin=\
            →"Infinity" />
14        </DataField>
        <DataField name="Request_Template" optype="categorical" \
            →dataType="string" />
16        </DataDictionary>
    </DegenerateDimension>
18
    <!-- Assure Referential Integrity -->
    <DegenerateDimension>
20        <DataDictionary numberOfFields="5">
22

```

⁹The transformation actually makes use of an extensive list of keywords and strings known to be used by current Web robots as part of their identifiers. This list must be updated on a regular basis.

```

24     <DataField name="Product_Object_ID" optype="categorical" \
        →datatype="string" />
26     <DataField name="Request_Assortment_ID" optype="categorical" \
        →datatype="string" />
28     <DataField name="Request_Subassortment_ID" optype="categorical" \
        →datatype="string" />
30     <DataField name="Content_ID" optype="categorical" datatype="\
        →string" />
32     <DataField name="Customer_ID" optype="categorical" datatype="\
        →string" />
34     </DataDictionary>

36     <!-- Transformation as before in line 37 of listing G.13 on page 162 \
        →with one ReplaceNullValues mapping for each attribute. -->
38     </DegenerateDimension>

40     <!-- Referrer Dimension -->
42     <RegularDimension role="Request_Referrer">

44         <!-- Modeled analogously as before in line 73
46             of listing G.15 on page 166. -->
48     </RegularDimension>

50     <!-- Session Mart -->
52     <HybridDimension role="Session">
54         <TableStream selectCacheSize="2000" table="SESSION_FACT" \
56             →insertCacheSize="2000" batchSize="1000" keyGenClass="wusan.\
58             →pdm.Input.Relational.TrivialPrimaryKey">
60             <MissingValueEncoding>
62                 <MissingValue>null</MissingValue>
                    <MissingValue>?</MissingValue>
                </MissingValueEncoding>
            </TableStream>
        </HybridDimension>

        <!-- Date Dimension -->
        <RegularDimension role="Request_Date">

            <!-- Modeled analogously as before in line 19
                of listing G.12 on page 161. -->
            </RegularDimension>

            <!-- Time Dimension -->
            <RegularDimension role="Request_Time">

                <!-- Modeled analogously as before in line 31
                    of listing G.12 on page 161. -->
            </RegularDimension>
        </StarSchema>

```

This listing is similar to the WusanML model of the session mart depicted in listing G.15 on page 166, except for the second hybrid dimension in line 41. This WusanML tag couples the entire session mart to the customer mart and connects both ETL processes (compare figure 29 on page 90). As the session mart's user agent degenerate dimension implements the vector filter described before, not only the sessions initiated by Web robots are filtered but also, due to the coupling of the data marts, every single click of such a session. Executing the ETL process of the coupled data marts is similar to listing G.9 on page 159 with the difference that,

prior to creating the clickstream mart, the session mart must be instantiated. This results in its WusanML description being stored in an internal repository that is read when the clickstream mart is created. Then, invoking the `etl` method of the clickstream mart executes the *coupled* ETL process.

G.3 Tapping the Data Warehouse for Data Mining

In this section, it is assumed that the data warehouse has been made operational for OLAP analyses as discussed before. In order to realize the recommendation engine outlined in section 5.1 on page 95, data mining models must be built on the created data marts and OLAP queries must be executed to generate the top-selling lists.¹⁰ Both methodical requirements are fully supported by WUSAN through its embedded CWM interface for data mining and through Mondrian's MDX API.¹¹ This section discusses alternatives for tapping the data warehouse for data mining. Section G.3.1 investigates how OLAP and data mining can be generally combined and section G.3.2 on the facing page considers a novel stream class based on an OLAP drill-through of an MDX query.

G.3.1 Combining Data Mining and OLAP

As for the data mining aspect in definition 3.5 on page 32, it is necessary to recall footnote 17 on page 32, which summarized the different options of combining data mining and OLAP, that is, the data mining and data warehousing approaches. Basically, conducting data mining analyses with WUSAN requires a stream on which the data mining algorithms operate. Streams are inherited from the XELOPES data mining library and were discussed in detail in section 4.2.2 on page 68.

Investigating the “*cubing then mining*” approach, which was mentioned in that very footnote cited in the previous paragraph, in the end amounts to exploring options for creating streams on the data warehouse. Basically, the `MiningQuerySqlStream` in figure 48 on page 135 can be employed to create streams on the database tables of the data storage layer^{12, 13}. However, this stream implies that the user has in-depth knowledge about the table structure of the data warehouse and is knowledgeable in SQL. On the other hand, it is worthwhile to leverage the multi-dimensional query capabilities of MDX in order to define a stream on the OLAP layer¹⁴. WUSAN's `MiningQueryMdxStream`, which is introduced in section G.3.2 on the next page, pursues this idea.

The “*mining then cubing*” approach mentioned in the footnote cited above is application specific and amounts to loading the discovered rules into the data warehouse to be able to constrain them in a multi-dimensional way. Nonetheless, this approach calls for extensive

¹⁰Recall the recommendation strategy in section 5.1.3 on page 97 and the resulting methodical requirements discussed in section 5.1.4 on page 99.

¹¹Compare item 1 on page 54. As the introductory book of Spofford [2001] discloses, MDX is a very powerful language for calculating complex multi-dimensional measures. Moreover, Sweiger et al. [2002] and Kimball and Merz [2000] reveal that applying the data warehousing approach to Web usage data enables unprecedented analytical capabilities for data warehousing in general. Consequently, through its architecture depicted in figure 15 on page 53, WUSAN also adequately covers the Web reporting aspect of definition 3.5 on page 32, yet leaving the question unanswered regarding which measures are particularly beneficial for WUA.

¹²See figure 16 on page 60.

¹³It is important to note that, unlike the XELOPES `MiningSqlStream`, the `MiningQuerySqlStream` assembles its meta-data from WUSAN's internal meta-data repository, hence producing more precise meta-data than its super-class, which auto-detects its meta-data.

¹⁴See figure 16 on page 60.

research efforts that investigate the alternatives for applying OLAP techniques to data mining results, especially in a domain-specific context.

Finally, the “*cubing while mining*” approach is promising for WUSAN in view of applying taxonomies during data mining [compare Thess and Bolotnicov, 2004, sections 6.3.3.5 and 6.10.2.3], since OLAP dimensions contain hierarchies that can be exploited for creating taxonomies. As the hierarchies are created during data warehouse design (compare section G.4 on the following page), their exploitation as taxonomies is a straightforward task, since the hierarchies are accessible through the Mondrian API. Nevertheless, further research is required on how taxonomies are beneficial for Web usage mining.¹⁵

G.3.2 A Stream Based on Drill-Through

Figure 59 depicts the UML class diagram of the `MiningQueryMdxStream` class, which extends the `MiningQuerySqlStream` class in figure 48 on page 135. This stream takes an MDX query, the result set of which consists of one single cell only as input, and executes a drill-through on that particular cell. The drill-through delivers all the vectors that contribute to the cell’s measure value.¹⁶

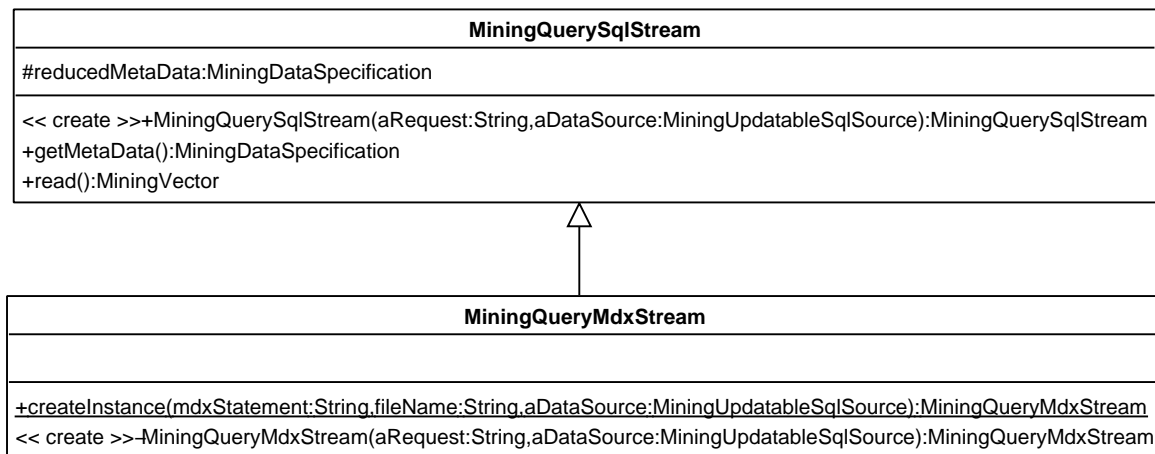


Figure 59: Multi-dimensional stream based on a drill-through.

In fact, the `createInstance` method sends the MDX statement to the OLAP server, which assembles an SQL query required to perform the drill-through on the RDBMS. Apparently, the `MiningQueryMdxStream` is a `MiningQuerySqlStream` based on the drill-through SQL statement.

With this stream, data for data mining tasks can be selected from the data warehouse by constraining the dimensions of a data mart. This can be done with the JPivot component as illustrated in figure 60 on the next page. With JPivot, users can graphically constrain a cube until the desired drill-through cell is displayed. Then, by pressing the MDX button, the MDX statement that creates the displayed result set is revealed. This string can then be employed as an argument for the MDX stream sought.

¹⁵The question regarding how taxonomies are beneficial for Web reporting is deceptive: it is a basic capacity of OLAP to calculate a measure for any level of a hierarchy, that is, hierarchies (or taxonomies) are considered by default in OLAP-based Web reporting.

¹⁶Drill-throughs are discussed in detail in Spofford [2001, chapter 9].

WUSAN Data Warehouse (KDD Cup 2000 Data)



MDX Query Editor

The MDX Editor window displays the following query:

```
select {[Date].[All Dates].[2000].[Quarter 1].[March]} ON columns,
       {[Time.ByTimeOfDay].[All Time.ByTimeOfDay].[afternoon]} ON rows
from [Clickstream]
```

Below the editor, a pivot table is shown with the following data:

	↑Date
↑Time	↑March
↑afternoon	↓17 194

Figure 60: Assembling an MDX statement with JPivot.

Remark. Although the current implementation of the `MiningQueryMdxStream` supports a one-dimensional result set only, this approach could be extended to arbitrary result sets. For example, the result set of figure 60 could be expanded by a second cell referring to February. Then, the `MiningQueryMdxStream` could combine the drill-through SQL statements of both cells as a basis for retrieving the desired vectors. However, this approach of simply combining drill-through SQL statements is not reasonable if a result sets contains cells at different levels of a hierarchy. In this case, a drill-through makes sense only at the highest level of the hierarchy, as the drill-throughs of the subordinate levels are contained in those of the highest level. Hence, the highest levels must be detected through result set analysis using the Mondrian API in order to determine the minimum set of cells required for assembling the overall drill-through SQL statement.

G.4 Configuration of the Data Warehouse

At this point, it is assumed that the RDBMS has been populated with the LOORDSM as discussed in section G.2 on page 160 and the previous sections and that performance optimizations under the terms of section 5.3 on page 109 have been completed, that is, the data warehousing component in figure 15 on page 53 is principally ready for operation.

As the figure illustrates, the ROLAP engine and a visualization module operate on top of the data warehouse. Both constituents, the Mondrian OLAP server and the JPivot visualization, were discussed in section 3.3.3.4 on page 54. In order to map the database tables of the RDBMS to the logical multi-dimensional model of the OLAP server, an XML configuration file (referred to as a *schema file*), the syntax of which is discussed by Hyde [2005], must be created and provided to Mondrian.¹⁷ This file realizes the link of the data storage layer and the OLAP layer in figure 16 on page 60, which was referred to as the second mapping in section 5.2.1 on page 101. As the syntax of the schema file is intuitively comprehensible, an in-depth syntax discussion can be skipped.

Since the entire schema file is quite lengthy, subsequent listings illustrate only parts of it in order to demonstrate its principal configuration. The table names refer to the database tables created in the previous section. Generally, the attributes referred to in the schema file represent only a subset of the attributes actually available in the database tables. The XML tags of the schema file correspond to the commonly used notions in dimensional modeling.

The date dimension in listing G.17 is not cube specific and is hence modeled independently of any cube. It is referenced by almost all cubes of the data warehouse, for instance, the clickstream cube in listing G.20 on page 175 and the order line cube in listing G.21 on page 176. The date dimension models three hierarchies that can be altered during analysis for different date perspectives.

Listing G.17: Defining the date dimension in Mondrian's schema file.

```

1 <!-- Date Dimension -->
  <Dimension name="Date">
3
4   <!-- Standard hierarchy -->
5   <Hierarchy hasAll="true" primaryKey="WUSAN_ID">
6     <Table name="DATE_DIMENSION" schema="WUSAN" />
7     <Level name="Year" column="Year" type="Numeric" uniqueMembers="\
      →true" />
8     <Level name="Quarter" column="Quarter_Label" type="String" \
      →ordinalColumn="Quarter_Numeric" uniqueMembers="false" />
9     <Level name="Month" column="Month_Label" type="String" \
      →ordinalColumn="Month_Numeric" uniqueMembers="false" />
10    <Level name="Day" column="Day" type="Numeric" uniqueMembers="false\
      →">
11      <Property name="Weekday" column="Weekday_Label" type="String"/\
      →>
12    </Level>
13  </Hierarchy>
14
15  <!-- Hierarchy by week of month -->
16  <Hierarchy hasAll="true" primaryKey="WUSAN_ID" name="ByWeekOfMonth" >
17    <Table name="DATE_DIMENSION" schema="WUSAN" />
18    <Level name="Year" column="Year" type="Numeric" uniqueMembers="\
      →true" />
19    <Level name="Quarter" column="Quarter_Label" type="String" \
      →ordinalColumn="Quarter_Numeric" uniqueMembers="false" />
20    <Level name="Month" column="Month_Label" type="String" \
      →ordinalColumn="Month_Numeric" uniqueMembers="false" />

```

¹⁷This syntax follows the notions of dimensional modeling. Hence, it is again necessary to recall footnote 32 on page 104.

```

21     <Level name="Week_of_Month" column="Week_of_Month_Label" type="\
        ↳String" ordinalColumn="Week_of_Month_Numeric" uniqueMembers="\
        ↳false" />
        <Level name="Weekday" column="Weekday_Label" type="String" \
        ↳ordinalColumn="Weekday_Numeric" uniqueMembers="false" />
23 </Hierarchy>

25 <!-- Hierarchy by week of year -->
    <Hierarchy hasAll="true" primaryKey="WUSAN_ID" name="ByWeekOfYear">
27     <Table name="DATE_DIMENSION" schema="WUSAN" />
        <Level name="Year" column="Year" type="Numeric" uniqueMembers="\
        ↳true" />
29     <Level name="Week_of_Year" column="Week_of_Year_Label" type="\
        ↳String" ordinalColumn="Week_of_Year_Numeric" uniqueMembers="\
        ↳false" />
        <Level name="Weekday" column="Weekday_Label" type="String" \
        ↳ordinalColumn="Weekday_Numeric" uniqueMembers="false" />
31 </Hierarchy>
</Dimension>

```

The time dimension in listing G.18 is defined analogously to the date dimension, and it is also referenced by most cubes.

Listing G.18: Defining the time dimension in Mondrian's schema file.

```

<!-- Time Dimension -->
2 <Dimension name="Time">

4     <!-- Standard hierarchy -->
    <Hierarchy hasAll="true" primaryKey="WUSAN_ID">
6         <Table name="TIME_DIMENSION" schema="WUSAN" />
            <Level name="Hour" column="Hour" type="Numeric" uniqueMembers="\
            ↳false" />
8             <Level name="Minute" column="Minute" type="Numeric" uniqueMembers=\
            ↳"false" />
            <Level name="Second" column="Second" type="Numeric" uniqueMembers=\
            ↳"false" />
10    </Hierarchy>

12    <!-- Hierarchy by time of day -->
    <Hierarchy hasAll="true" primaryKey="WUSAN_ID" name="ByTimeOfDay">
14        <Table name="TIME_DIMENSION" schema="WUSAN" />
            <Level name="Time_of_Day" column="Time_of_Day_Label" type="String" \
            ↳ ordinalColumn="Time_of_Day_Numeric" uniqueMembers="true" />
16        <Level name="Hour" column="Hour" type="Numeric" uniqueMembers="\
            ↳false" />
            <Level name="Minute" column="Minute" type="Numeric" uniqueMembers=\
            ↳"false" />
18        <Level name="Second" column="Second" type="Numeric" uniqueMembers=\
            ↳"false" />
    </Hierarchy>
20 </Dimension>

```

The referrer dimension in listing G.19 on the facing page is also defined analogously to the dimensions before and is referenced by the clickstream cube in listing G.20 on the next page.

Listing G.19: Defining the referrer dimension in Mondrian's schema file.

```

2 <!-- Referrer Dimension -->
  <Dimension name="Referrer">
4     <!-- Standard hierarchy -->
      <Hierarchy hasAll="true" primaryKey="WUSAN_ID">
6         <Table name="REFERRER_DIMENSION" schema="WUSAN" />
          <Level name="Host_Name" column="host_name" type="String" \
            →uniqueMembers="true">
8             <Property name="IP_address" column="IP_address" type="String"/\
              →>
              <Property name="Country" column="country" type="String"/>
10          </Level>
          <Level name="Path" column="path" type="String" uniqueMembers="\
            →false" />
12      </Hierarchy>
    </Dimension>

```

Listing G.20 depicts the definition of a rudimentary clickstream cube that references the dimensions previously defined. Furthermore, this listing defines additional dimensions that are accessible only to the defining cube. Each cube requires at least one measure, which is chosen as a simple count aggregation for the cube.

Listing G.20: Defining the clickstream mart in Mondrian's schema file.

```

1 <!-- Clickstream Cube -->
  <Cube name="Clickstream">
3     <Table name="CLICKSTREAM_FACT" schema="WUSAN"/>
5     <!-- Referrer Dimension -->
      <DimensionUsage name="Referrer" source="Referrer" foreignKey="Request_\
        →Referrer"/>
7
9     <!-- Date Dimension -->
      <DimensionUsage name="Date" source="Date" foreignKey="Request_Date"/>
11
13     <!-- Time Dimension -->
      <DimensionUsage name="Time" source="Time" foreignKey="Request_Time"/>
15
17     <!-- Content Dimension -->
      <DimensionUsage name="Content" source="Content" foreignKey="Content_ID\
        →"/>
19
21     <!-- Session ID -->
      <Dimension name="Session_ID">
        <Hierarchy hasAll="true">
          <Level column="Session" name="Session_ID" type="String"/>
        </Hierarchy>
      </Dimension>
23
25     <!-- Customer ID -->
      <Dimension name="Customer_ID">
        <Hierarchy hasAll="true">
          <Level column="Customer_ID" name="Customer_ID" type="String"/>
        </Hierarchy>
      </Dimension>
29
31     <!-- Measure -->

```

```

33 <Measure column="WUSAN_ID" aggregator="count" name="Count_Clicks" \
    →visible="false" />
</Cube>

```

Measures were addressed in item 4 on page 92. For the LOORDSM, they refer to numeric columns in fact tables that are computed during the ETL process or thereafter. Measures in Mondrian are more general, as they are not restricted to reading numeric columns from the fact table. Mondrian provides the `CalculatedMember` tag, which can be employed to generate complex calculated measures from measures defined with the `Measure` tag. An example of this approach is demonstrated in line 28 in listing G.21, which models the order line cube. The measure `Average Items` computes averages on the order level, not on the order line level. Calculated measures are defined with MDX, that is, whatever can be computed with MDX can be defined as a calculated measure.¹⁸

Listing G.21: Defining the order line mart in Mondrian's schema file.

```

2 <Cube name="OrderLine">
  <Table name="ORDER_LINE_FACT" schema="WUSAN" />
4   <!-- Date, Time, and Product Dimensions -->
  <DimensionUsage source="Date" name="Date" foreignKey="Order_Line_Date" \
    → />
6  <DimensionUsage source="Time" name="Time" foreignKey="Order_Line_Time" \
    → />
  <DimensionUsage source="Product" name="Product" foreignKey="Product_ID" \
    →" />
8
  <!-- Order ID -->
10 <Dimension name="Order_ID">
  <Hierarchy hasAll="true">
12   <Level column="Order_ID" name="Order_ID" type="String" />
  </Hierarchy>
14 </Dimension>
16 <!-- Order Line ID (required for drill-through) -->
  <Dimension name="Order_Line_ID">
18   <Hierarchy hasAll="true">
    <Level column="Order_Line_ID" name="Order_Line_ID" type=" \
      →String" />
20   </Hierarchy>
  </Dimension>
22
  <!-- Measures -->
24 <Measure column="Order_Line_ID" aggregator="count" name="Count_Items" \
    → />
  <Measure column="Order_ID" aggregator="distinct_count" name="Count_ \
    →Order" />
26 <Measure column="Order_Line_Unit_Sale_Price" aggregator="sum" name=" \
    →Sale_Price" formatString="#,##0.00_ $" />
28
  <!-- Calculated Measures -->
  <CalculatedMember name="Average_Items" dimension="Measures" visible=" \
    →true" formula="[Measures].[Count_Items]/Count([Order_ID]. \
    →CurrentMember.Children, EXCLUDEEMPTY)" />

```

¹⁸Compare the introduction to MDX by Spofford [2001].

```

30     <CalculatedMemberProperty name="FORMAT_STRING" value="#,#0.0" />
    </CalculatedMember>
32     <CalculatedMember name="Average_Sale_Price" dimension="Measures" \
        →visible="true" formula="[Measures].[Sale_Price]/Count([Order_ID]. \
        →CurrentMember.Children,EXCLUDEEMPTY)">
        <CalculatedMemberProperty name="FORMAT_STRING" value="#,#0.00_ $" \
        →/>
34     </CalculatedMember>
36 </Cube>

```

Figure 61 visualizes the order mart with JPivot (compare the WUSAN architecture in figure 15 on page 53). The chosen view of the order line cube depicts the five measures defined above for the chosen dates with a slicer on the time of day, that is, all figures refer to lunchtime (12:00 pm - 1:59 pm): (i) the measure `Count Items` counts all order items, (ii) the measure `Count Orders` counts all orders, (iii) the measure `Sale Price` totals the sale prices of all order items, (iv) the calculated measure `Average Items` computes the average number of items per order, and (v) the calculated measure `Average Sale Price` computes the average total sale price per order. As the defined measures demonstrate, the order line cube allows analyses not only on the order item level but also on the aggregated order level. Alternatively, a separate order cube based on the `ORDER_FACT` table can be defined. This table represents the materialized aggregations of the order line mart.

With MDX, complex measure computations can be formulated compactly. Yet, the notation is quite complex, which makes the definition of more sophisticated measures a challenge, the more so as an editor that checks MDX syntax is not available in the WUSAN prototype.

WUSAN Data Warehouse (KDD Cup 2000 Data)



	Kennzahlen				
Date	Count Items	List Price	Sale Price	Average Items	Average Sale Price
+2000	174	1.756,13 \$	1.745,20 \$	1,8	18,18 \$
+March	145	1.536,43 \$	1.525,50 \$	1,8	18,83 \$

Slicer: [Time of Day=lunchtime]

Figure 61: Order line mart visualized with JPivot (German localization).

The power of defining and deploying multi-dimensional calculated measures cannot be overestimated. As this thesis centers on the ETL process for WUA, it is beyond its scope to discuss WUA-, EC-, and ECRM-specific measures for clickstream, session, or sales cubes or for combinations of cubes defined in the data warehouse. However, the WUSAN architecture in figure 15 on page 53 as a whole yields the chance to deploy common specific measures for WUA, EC, and ECRM, for example, the measures discussed by Schonberg et al. [2000], Cutler and Sterne [2000], and Bertot et al. [1997], with manageable effort.

Bibliography

- Lou Agosta. Market Overview Update: ETL. White Paper RPA-032002-00021, Giga Information Group, March 2002. URL http://www.informatica.com/news/awards/giga_etl.pdf. Access date: 06/15/2005.
- Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proceedings of the 20th International Conference on Very Large Databases, VLDB 1994*, pages 487–499, Santiago de Chile, Chile, September 1994. Morgan Kaufmann.
- Rakesh Agrawal and Ramakrishnan Srikant. Mining Sequential Patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proceedings of the Eleventh International Conference on Data Engineering, ICDE 1995*, pages 3–14, Taipei, Taiwan, March 1995. IEEE Computer Society Press.
- Werner Altmann, René Fritz, and Daniel Hinderink. *TYPO3 Enterprise Content Management*. Open Source Press, 1 edition, 2004.
- AMAZON. URL <http://www.amazon.com/>. Access date: 01/04/2005.
- Raphael Amit and Christoph Zott. Value Creation in E-Business. *Strategic Management Journal*, 22(6-7):493–520, June-July 2001.
- ANALOG. URL <http://www.analog.cx/>. Access date: 05/05/2005.
- Sarabjot Singh Anand, Maurice Mulvenna, and Karine Chevalier. On the Deployment of Web Usage Mining. *Lecture Notes in Computer Science*, 3209:23–42, September 2004.
- Anastasia I. Andritsou and Nikos B. Pronios. Mobility Convergence in Heterogeneous (Fixed and Mobile) Networks. In *Proceedings of the 40th European Telecommunications Congress*, pages 96–100, Barcelona, Spain, August 2001.
- Suhail Ansari, Ron Kohavi, Llew Mason, and Zijian Zheng. Integrating E-Commerce and Data Mining: Architecture and Challenges. In Nick Cercone, Tsau Young Lin, and Xindong Wu, editors, *Proceedings of the 2001 IEEE International Conference on Data Mining, ICDM 2001*, pages 27–34, San José, CA, USA, December 2001. IEEE Computer Society Press.
- APACHE. *Apache HTTP Server Project*. Apache Software Foundation. URL <http://httpd.apache.org/>. Access date: 04/30/2005.
- ASCENTIAL. URL <http://www.ascential.com>. access date: 06/01/2004.
- Paul Ashley, Satoshi Hada, Günter Karjoth, and Matthias Schunter. E-P3P Privacy Policies and Privacy Authorization. In Sushil Jajodia and Pierangela Samarati, editors, *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, pages 103–109, Washington, DC, USA, November 2002. ACM Press.
- Gunnar Auth and Eitel von Maur. A Software Architecture for XML-Based Metadata Interchange in Data Warehouse Systems. *Lecture Notes in Computer Science*, 2490:1–14, January 2002.

- AW-STATS. URL <http://awstats.sourceforge.net/>. Access date: 05/05/2005.
- Yannis Bakos. The Emerging Role of Electronic Marketplaces on the Internet. *Communications of the ACM*, 41(8):35–42, August 1998.
- Russ Banham. Amazon Finally Clicks. Amazon’s Lesson in Focus and Flexibility. *CFO IT*, Spring Issue, March 2004. URL http://www.cfo.com/article.cfm/3012377/1/c_3046608?f=insidecfo. Access date: 01/04/2005.
- Ranieri Baraglia and Paolo Palmerini. SUGGEST: A Web Usage Mining System. In *Proceedings of IEEE International Conference on Information Technology: Coding and Computing, Special Session on Web and Hypermedia Systems.*, pages 282–287, Las Vegas, NV, USA, April 2002. IEEE Computing Society Press.
- Armenak Barsegyan, Mikhail Kupryanov, Valentin Stepanenko, and Ivan Kholod. Методы и модели анализа данных: OLAP и Data Mining. BHV, St. Petersburg, Russia, 1 edition, 2004.
- BBB-ONLINE. URL <http://www.bbbonline.org/>. Access date: 11/14/2004.
- Paola Benassi. TRUSTe: An Online Privacy Seal Program. *Communications of the ACM*, 42(2):56–59, February 1999.
- Bettina Berendt and Myra Spiliopoulou. Analysis of Navigation Behaviour in Web Sites Integrating Multiple Information Systems. *VLDB Journal*, 9(1):56–75, March 2000.
- Bettina Berendt, Bamshad Mobasher, Myra Spiliopoulou, and Jim Wiltshire. Measuring the Accuracy of Sessionizers for Web Usage Analysis. In Vipin Kumar and Robert Grossman, editors, *Proceedings of the First SIAM International Conference on Data Mining, SDM 2001*, pages 7–14, Chicago, IL, USA, April 2001. SIAM Press.
- Bettina Berendt, Bamshad Mobasher, Miki Nakagawa, and Myra Spiliopoulou. The Impact of Site Structure and User Environment on Session Reconstruction in Web Usage Analysis. In Osmar R. Zaïane, editor, *Proceedings of the 4th WebKDD Workshop, WebKDD 2002*, Edmonton, Canada, August 2002.
- Hal Berghel. Digital village: Caustic cookies. *Communications of the ACM*, 44(5):19–22, May 2001.
- Pavel Berkhin, Jonathan D. Beche, and Dee Jay Randall. Interactive Path Analysis of Web Site Traffic. In Doheon Lee, Mario Schkolnick, Foster Provost, and Srikant Ramakrishnan, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001*, pages 414–419, San Francisco, CA, USA, August 2001. ACM Press.
- Howard G. Bennett and Marcy D. Kuhn. The Emergence of Electronic Customer Relationship Management. In Thomas B. Fowler and John G. Leigh, editors, *The 2002 Telecommunications Review*, pages 91–96. Mitretek Systems, 2001. URL <http://www.mitretek.org/home.nsf/Publications/TelecommReview1999>. Access date: 05/26/2005.
- Alex Berson, Stephen Smith, and Kurt Thearling. *Building Data Mining Applications for CRM*. McGraw-Hill, New York, NY, USA, 1 edition, 2000.

- John Carlo Bertot, Charles McClure, William E. Moen, and Jeffrey Rubin. Web Usage Statistics: Measurement Issues and Analytical Techniques. *Government Information Quarterly*, 14(4):373–395, 1997.
- BIZGRES. Bizgres. PostgreSQL for Business Intelligence and Data Warehousing. URL <http://www.bizgres.org/>. Access date: 08/28/2005.
- Rich Bowen, Daniel L pez Ridruejo, and Allan Liska. *Apache Administrator’s Handbook*. Sams Publishing, Indianapolis, IN, USA, 1 edition, 2002.
- John S. Breese, David Heckerman, and Carl Kadie. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Gregory F. Cooper, editor, *Proceedings of the Fourteenth Conference in Uncertainty in Artificial Intelligence, UAI 1998*, pages 43–52, Madison, WI, USA, July 1998. Morgan Kaufmann.
- Manfred Bruhn. Strategische Ausrichtung des Relationship Marketing. In Adrian Payne and Reinhold Rapp, editors, *Handbuch Relationship Marketing, Konzeption und erfolgreiche Umsetzung*, pages 45–57. Vahlen, 2003.
- Jan W. Buzydlowski, Il-Yeol Song, and Lewis Hassell. A Framework for Object-Oriented On-Line Analytic Processing. In *Proceedings of the 1st ACM International Workshop on Data Warehousing and OLAP*, pages 10–15, Washington, DC, USA, November 1998. ACM Press.
- Paulo Carreira and Helena Galhardas. Efficient Development of Data Migration Transformations. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, SIGMOD ’04*, pages 915–916, Paris, France, June 2004. ACM Press.
- Lara D. Catledge and James Pitkow. Characterizing Browsing Strategies in the World Wide Web. *Computer Networks and ISDN Systems*, 27(6):1065–1073, 1995.
- Soumen Chakrabarti. *Mining the Web: Analysis of Hypertext and Semi Structured Data*. Morgan Kaufmann, San Francisco, CA, USA, 1 edition, 2002.
- Soumen Chakrabarti, Byron E. Dom, S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, Andrew Tomkins, David Gibson, and John Kleinberg. Mining the Web’s Link Structure. *Computer*, 32(8):60–67, August 1999.
- Elsie Chan and Paula M. C. Swatman. Electronic Commerce: A Component Model. In *Proceedings of the Third Annual COLLECTeR Conference on Electronic Commerce*, Wellington, New Zealand, November 1999.
- Kuo-chung Chang, Joyce Jackson, and Varun Grover. E-Commerce and Corporate Strategy: An Executive Perspective. *Information & Management*, 40(7):663–675, August 2003.
- Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and R diger Wirth. CRISP-DM 1.0. Step-by-Step Data Mining Guide. White Paper, CRISP-DM Consortium, 2000. URL <http://www.crisp-dm.org>. Access date: 05/04/2005.
- Scott Chapman and Gurpreet Dhillon. Privacy and the Internet: The Case of DoubleClick, Inc. In Gurpreet Dhillon, editor, *Social Responsibility in the Information Age: Issues and Controversies*, pages 75–88. Idea Group Publishing, 2002.

- Alisa Chatranon, Jason C. H. Chen, P. Pete Chong, and Ye-Sho Chen. Customer Relationship Management (CRM) and E-Commerce. In *Proceedings of the First International Conference on Electronic Business*, Hong Kong, China, December 2001.
- Zhixiang Chen, Ada Wai-Chee Fu, and Frank Chi-Hung Tong. Optimal Algorithmus for Finding User Access Sessions from Very Large Web Logs. In Ming-Syan Chen, Philip S. Yu, and Bing Liu, editors, *Advances in Knowledge Discovery and Data Mining. Proceedings of the 6th Pacific-Asia Conference, PAKDD 2002*, pages 290–296, Taipei, Taiwan, May 2002. Springer.
- Julian Chu and Gina P. Morrison. Enhancing the Customer Shopping Experience: 2002 IBM/NRF “Store of the Future” Survey. Technical Report, IBM Institute for Business Value, Somers, NY, USA, 2003. URL <http://www-1.ibm.com/industries/retail/doc/content/bin/GE510-3261-01F.pdf>. Access date: 03/02/2005.
- Tim Coltman, Timothy M. Devinney, Alopi S. Latukefu, and David F. Midgley. Keeping E-Business in Perspective. *Communications of the ACM*, 45(8):69–73, August 2002.
- Robert Cooley. *Web Usage Mining: Discovery and Application of Interesting Patterns from Web Data*. PhD thesis, University of Minnesota, Minneapolis, MN, USA, 2000.
- Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Web Mining: Information and Pattern Discovery on the World Wide Web. In *Proceedings of the 9th International Conference on Tools with Artificial Intelligence, ICTAI 1997*, pages 558–567, Newport Beach, CA, USA, November 1997. IEEE Computer Society Press.
- Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data Preparation for Mining World Wide Web Browsing Patterns. *Journal of Knowledge and Information Systems*, 1(1):5–32, February 1999.
- Matt Cutler and Jim Sterne. E-Metrics - Business Metrics for the New Economy. White Paper, NetGenesis Corporation, 2000. URL <http://www.targeting.com/emetrics.pdf>. Access date: 08/24/2005.
- Amit Das, Christina Wai Lin Soh, and Patrick Chang Boon Lee. A Model of Customer Satisfaction with Information Technology Service Providers: An Empirical Study. In *Proceedings of the 1999 ACM SIGCPR Conference on Computer Personnel Research*, pages 190–193, New Orleans, LA, USA, 1999. ACM Press.
- DMG-PMML. *Predictive Model Markup Language*. Data Mining Group. URL <http://www.dmg.org/>. Access date: 02/02/2005.
- DATADIRECT. Designing Performance-Optimized JDBC Applications. White Paper, DataDirect Technologies, 2004. URL <http://www.datadirect.com/developer/jdbc/docs/jdbcdesign2.pdf>. Access date: 02/05/2005.
- DATANAUTICS. Web Mining Whitepaper. Driving Business Decisions in Web Time. White Paper, Datanautics, Inc., March 2000. URL http://www.datanautics.com/forms/literature/wp_webmining.pdf. Access date: 05/08/2005.
- Stanley M. Davis. *Future Perfect*. Addison Wesley, Reading, MA, USA, 1 edition, 1987.

- John Day and Hubert Zimmermann. The OSI Reference Model. In Richard Jerry Linn and M. Ümit Uyar, editors, *Conformance Testing Methodologies and Architectures for OSI Protocols*, pages 38–44. IEEE Computer Society Press, 1995.
- Barbara Dinter, Carsten Sapia, Gabriele Höfling, and Markus Blaschka. The OLAP Market: State of the Art and Research Issues. In *Proceedings of the First ACM International Workshop on Data Warehousing and OLAP, DOLAP 1998*, pages 22–27, Washington, DC, USA, November 1998. ACM Press.
- Gunter Dueck. *Die β -inside Galaxie*. Springer, Berlin, Germany, 1 edition, 2001.
- Gunter Dueck. *E-Man. Die neuen virtuellen Herrscher*. Springer, Berlin, Germany, 2 edition, 2002.
- EBAY. URL <http://www.ebay.com/>. Access date: 01/04/2005.
- EC-MATRIX. *Electronic Commerce Matrix*, March 2002. URL <http://fox.wikis.com/wc.dll?Wiki~ECommerceMatrix~SoftwareEng>. Access date: 12/30/2004.
- EC-REPORT. The European e-Business Report. A Portrait of e-Business in 10 Sectors of the EU Economy. Third Synthesis Report of the e-Business W@tch, European Commission, 2004. URL <http://www.ebusiness-watch.org/resources/documents/eBusiness-Report-2004.pdf>. Access date: 05/25/2005.
- Magdalini Eirinaki and Michalis Vazirgiannis. Web Mining for Web Personalization. *ACM Transactions on Internet Technology*, 3(1):1–27, February 2003.
- Richard Emberson. Aggregate Tables, July 2005. URL http://mondrian.sourceforge.net/head/aggregate_tables.html. Access date: 08/24/2005.
- ETI. URL <http://www.eti.com>. Access date: 06/01/2004.
- Patti F. Evans. E-Business News: The Current State of Online Retail., 2004. URL http://www.vectec.org/researchcenter/onlineretail_dec04.html. Access date: 01/01/2005.
- Liam Fahey, Rajendra Srivastava, Joyce S. Sharon, and David E. Smith. Linking E-Business and Operating Processes: The Role of Knowledge Management. *IBM Systems Journal*, 40(4):889–907, 2001.
- Usama Fayyad, Georges G. Grinstein, and Andreas Wierse. *Information Visualization in Data Mining and Knowledge Discovery*. Morgan Kaufmann, San Francisco, CA, USA, 1 edition, 2001.
- Richard A. Feinberg, Rajesh Kadam, Leigh Hokama, and Ikusk Kim. The State of Electronic Customer Relationship Management in Retailing. *International Journal of Retail and Distribution Management*, 30(10):470–481, October 2002.
- Jeffrey E. F. Friedl. *Mastering Regular Expressions*. O’Reilly, Sebastopol, CA, USA, 2 edition, 2002.
- Ann L. Fruhling and Lester A. Digman. The Impact of Electronic Commerce on Business-Level Strategies. *Journal of Electronic Commerce Research*, 1(1):13–22, February 2000.

- GOOGLE-NEWS. URL <http://news.google.com/>. Access date: 04/26/2005.
- GOOGLE-SEARCH. URL <http://www.google.com/>. Access date: 04/26/2005.
- Narasimhaiah Gorla. Features to Consider in a Data Warehousing System. *Communications of the ACM*, 46(11):111–115, November 2003.
- Johannes Grabmeier and Andreas Rudolph. Techniques of Cluster Algorithms in Data Mining. *Data Mining and Knowledge Discovery*, 6(4):303–360, October 2002.
- Robyn Greenspan. Moderate, Steady CRM Growth Through 2006, July 2003a. URL <http://www.ecrmguide.com/article.php/2230361>. Access date: 03/05/2005.
- Robyn Greenspan. Shoppers Utilize Multi-Channel Choices, March 2003b. URL <http://www.clickz.com/stats/sectors/retailing/article.php/2077861>. Access date: 03/08/2005.
- Robyn Greenspan. E-Commerce Penetration on the Rise, February 2004. URL <http://www.clickz.com/stats/sectors/retailing/article.php/3317811>. Access date: 07/09/2004.
- Seth Grimes. Open and Shut. *Intelligent Enterprise*, 8(1):12, January 2005.
- Rüdiger Grimm and Alexander Rossnagel. Can P3P Help to Protect Privacy Worldwide? In Shahram Ghandeharizadeh, Shih-Fu Chang, Stephen Fischer, Joseph Konstan, and Klara Nahrstedt, editors, *Proceedings of the 2000 ACM Workshops on Multimedia*, pages 157–160, Los Angeles, CA, USA, November 2000. ACM Press.
- Robert Grossman, Stuart Bailey, Ashok Ramu, Balinder Malhi, Philip Hallstrom, Ivan Pulleyn, and Xiao Qin. The Management and Mining of Multiple Predictive Models Using the Predictive Modeling Markup Language. *Information and Software Technology*, 41(9):589–595, June 1999.
- Robert Grossman, Mark Hornick, and Gregor Meyer. Data Mining Standards Initiatives. *Communications of the ACM*, 45(8):59–61, August 2002.
- Varun Grover and James T. C. Teng. E-Commerce and the Information Market. *Communications of the ACM*, 44(4):79–86, April 2001.
- Marty Hall and Larry Brown. *Core Servlets and Java Server Pages. Volume 1 – Core Technologies*. Prentice Hall, Upper Saddle River, NJ, USA, 2 edition, 2003.
- Jiawei Han. OLAP Mining: An Integration of OLAP with Data Mining. In Stefano Spaccapietra and Fred J. Maryanski, editors, *Data Mining and Reverse Engineering: Searching for Semantics, Proceedings of the Seventh Conference on Database Semantics, DS-7*, pages 3–20, Leysin, Switzerland, October 1997. Chapman & Hall.
- Jiawei Han and Micheline Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 1 edition, 2001.
- Sven Harmsen. Strategy in the Context of eCommerce. Directed Studies 42590, School of Business, Carleton University, Ottawa, Canada, 2001. URL <http://www.svenharmen.de/PDFs/590-eStrategy.pdf>. Access date: 12/30/2004.

- Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*. Springer, New York, NY, USA, 1 edition, 2001.
- Richard Hawkins, Johan Helsingius, Jiro Kokuryo, Robin Mansell, Lynn Margherio, Michael McCracken, Luc L. G. Soete, and Philip Swan. The Economic and Social Impact of Electronic Commerce. Preliminary findings and Research Agenda. Survey, OECD, 1999. URL <http://www.oecd.org/dataoecd/3/12/1944883.pdf>. Access date: 11/13/2004.
- Jay Henderson, Rich Berkman, and Marc Jacobson. Data Collection for Web Site Analysis. White Paper, Customer Centric Solutions, 2002. Preliminary Draft v. 014.
- Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating Collaborative Filtering Recommender Systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.
- Jochen Hipp, Ulrich Güntzer, and Gholamreza Nakhaeizadeh. Algorithms for Association Rule Mining - A General Survey and Comparison. *SIGKDD Explorations*, 2(1):58–64, June 2000.
- Hajo Hippner. CRM – Grundlagen, Ziele und Konzepte. In Hajo Hippner and Klaus D. Wilde, editors, *Grundlagen des CRM. Konzepte und Gestaltung*, pages 13–41, Wiesbaden, Germany, 2004. Gabler.
- Esther Hochsztain, Socorro Millán, B. Pardo, José M. Peña, and Ernestina Menasalvas. A Framework to Integrate Business Goals in Web Usage Mining. In Ernestina Menasalvas Ruiz, Javier Segovia, and Piotr S. Szczepaniak, editors, *First International Atlantic Web Intelligence Conference, AWIC 2003*, pages 28–36, Madrid, Spain, May 2003. Springer.
- Cay S. Horstmann and Gary Cornell. *Core Java 2. Volume I – Fundamentals*. Sun Microsystems Press, Santa Clara, CA, USA, 7 edition, 2005a.
- Cay S. Horstmann and Gary Cornell. *Core Java 2. Volume II – Advanced Features*. Sun Microsystems Press, Santa Clara, CA, USA, 7 edition, 2005b.
- Xiaohua Hu and Nick Cercone. A Data Warehouse/Online Analytic Processing Framework for Web Usage Mining and Business Intelligence Reporting. *International Journal of Intelligent Systems*, 19(7):585–606, July 2004.
- Sterling Hughes and Andrei Zmievski. *PHP Developer's Cookbook*. Sams Publishing, Indianapolis, IN, USA, 1 edition, 2001.
- Julian Hyde. How to Design a Mondrian Schema, April 2005. URL <http://mondrian.sourceforge.net/schema.html>. Access date: 08/24/2005.
- HYPKNOWSYS. URL <http://www.hypknowsys.org/>. Access date: 06/14/2005.
- IM-VISUALIZATION. *DB2 Intelligent Miner Visualization*. IBM Corporation. URL <http://www-306.ibm.com/software/data/iminer/visualization/>. Access date: 06/14/2005.
- IM-VISUALIZATION-HANDBOOK. *Using the Intelligent Miner Visualizers*. IBM Corporation, 2 edition, August 2004. URL <http://publibfp.boulder.ibm.com/epubs/pdf/h1267371.pdf>. Access date: 06/13/2005.

- INFORMATICA. URL <http://www.informatica.com>. Access date: 06/01/2004.
- A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, September 1999.
- JAKARTA-TOMCAT. URL <http://jakarta.apache.org/tomcat/>. Access date: 06/12/2005.
- Liu Jian-guo, Huang Zheng-hong, and Wu Wei-ping. Web Mining for Electronic Business Application. In *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT 2003*, pages 872–876, Chengdu, China, August 2003. IEEE Computer Society Press.
- Karuna P. Joshi, Anupam Joshi, and Yelena Yesha. On Using a Warehouse to Analyze Web Logs. *Distributed and Parallel Databases*, 13(2):161–180, March 2003.
- JPIVOT. URL <http://jpivot.sourceforge.net/>. Access date: 2004-04-19.
- JSR-69. JSR 69: Java OLAP Interface (JOLAP). URL <http://www.jcp.org/en/jsr/detail?id=69>. Access date: 06/05/2005.
- JSR-73. JSR 73: Data Mining API. URL <http://www.jcp.org/en/jsr/detail?id=73>. Access date: 06/05/2005.
- KDDCUP-DATA. KDD Cup 2000 Data and Questions, 2000. URL <http://www.ecn.purdue.edu/KDDCUP/data/>. Access date: 08/09/2005.
- Sean Kelly. Mining Data to Discover Customer Segments. *Interactive Marketing*, 4(3):235–242, January/March 2003.
- KETTLE. Kettle. Turning Data Into Business. URL <http://www.kettle.be/>. Access date: 01/18/2006.
- Ralph Kimball and Joe Caserta. *The Data Warehouse ETL Toolkit*. Wiley, Indianapolis, IN, USA, 1 edition, 2004.
- Ralph Kimball and Richard Merz. *The Data Webhouse Toolkit. Building the Web-Enabled Data Warehouse*. Wiley, Indianapolis, IN, USA, 1 edition, 2000.
- Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit. The Complete Guide to Dimensional Modeling*. Wiley, Indianapolis, IN, USA, 2 edition, 2002.
- Alfred Kobsa. Personalized Hypermedia and International Privacy. *Communications of the ACM*, 45(5):64–67, May 2002.
- Ron Kohavi. Mining E-Commerce Data: The Good, the Bad, and the Ugly. In Foster Provost, Ramakrishnan Srikant, Mario Schkolnick, and Doheon Lee, editors, *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2001*, pages 8–13, San Francisco, CA, USA, August 2001. ACM Press.
- Ron Kohavi and Foster Provost. Applications of Data Mining to Electronic Commerce. *Data Mining and Knowledge Discovery*, 5(1/2):5–10, January/April 2001.

- Ron Kohavi, Carla E. Brodley, Brian Frasca, Llew Mason, and Zijian Zheng. KDD-Cup 2000 Organizer's Report: Peeling the Onion. *SIGKDD Explorations*, 2(2):86–98, December 2000.
- Ron Kohavi, Neal J. Rothleder, and Evangelos Simoudis. Emerging Trends in Business Analytics. *Communications of the ACM*, 45(8):45–48, August 2002.
- Ron Kohavi, Llew Mason, Rajesh Parekh, and Zijian Zheng. Lessons and Challenges from Mining Retail E-Commerce Data. *Machine Learning*, 57(1/2):83–115, October 2004.
- Pranam Kolari and Anupam Joshi. Web Mining: Research and Practice. *Computing in Science & Engineering*, 6(4):49–53, July/August 2004.
- Raymond Kosala and Hendrik Blockeel. Web Mining Research: A Survey. *SIGKDD Explorations*, 2(1):1–15, June 2000.
- Neal Leavitt. Data Mining for the Corporate Masses? *Computer*, 35(5):22–24, May 2002.
- Albert L. Lederer, Dinesh A. Mirchandani, and Kenneth Sims. Electronic Commerce: A Strategic Application? In Magid Igbaria, editor, *Proceedings of the 1996 ACM SIGCPR/SIG-MIS Conference on Computer Personnel Research*, pages 277–287, Denver, CO, USA, April 1996. ACM Press.
- Heejin Lee, Robert M. O'Keefe, and Kyounglim Yun. The Growth of Broadband and Electronic Commerce in South Korea: Contributing Factors. *The Information Society*, 19(1):81–93, January-March 2003.
- Juhnyoung Lee, Mark Podlaseck, Edith Schonberg, and Robert Hoch. Visualization and Analysis of Clickstream Data of Online Stores for Understanding Web Merchandising. *Data Mining and Knowledge Discovery*, 5(1/2):59–84, January/April 2001.
- Josh Lerner and Jean Tirole. Some Simple Economics of Open Source. *Journal of Industrial Economics*, 50(2):197–234, June 2002.
- Michael Lewis. The Influence of Loyalty Programs and Short-Term Promotions on Customer Retention. *Journal of Marketing Research*, 41(3):281–292, August 2004.
- LOG4J. URL <http://logging.apache.org/log4j/docs/index.html>. Access date: 08/30/2005.
- David Lucking-Reiley and Daniel F. Spulber. Business-to-Business Electronic Commerce. *Journal of Economic Perspectives*, 15(1):55–68, Winter 2001.
- Yiming Ma, Bing Liu, and Ching Kian Wong. Web for Data Mining: Organizing and Interpreting the Discovered Rules Using the Web. *SIGKDD Explorations*, 2(1):16–23, June 2000.
- Thilo Maier. A Formal Model of the ETL Process for OLAP-Based Web Usage Analysis. In Bamshad Mobasher, Bing Liu, Brij Masand, and Olfa Nasraoui, editors, *Web Mining and Web Usage Analysis. Proceedings of the Sixth International Workshop on Knowledge Discovery from the Web, WebKDD 2004*, pages 23–34, Seattle, WA, USA, August 2004.
- Thilo Maier and Thomas Reinartz. Evaluation of Web Usage Analysis Tools. *Künstliche Intelligenz*, (1):65–68, January 2004.

- Amit Kumar Manjhi. Web Caching. Technical Report, Indian Institute of Technology, Kanpur, India, 2000. URL <http://www.cse.iitk.ac.in/~dheeraj/reports/caching.pdf>. Access date: 05/10/2005.
- Marcello Mariucci. Enterprise Application Server Development Environments. Technical report, University of Stuttgart, Stuttgart, Germany, 2000.
- Tim Martyn. Reconsidering Multi-Dimensional Schemas. *SIGMOD Record*, 33(1):83–88, March 2004.
- Tom McCall. Worldwide Business-to-Business Internet Commerce to Reach \$8.5 Trillion in 2005. Analysts to Discuss Current Market Climate at Upcoming iEB Conference, 2001. URL http://www4.gartner.com/5_about/press_room/pr20010313a.html. Access date: 01/02/2005.
- Jim Melton and Andrew Eisenberg. SQL Multimedia and Application Packages (SQL/MM). *SIGMOD Record*, 30(4):97–102, December 2001.
- Bart Meltzer and Robert Glushko. XML and Electronic Commerce: Enabling the Network Economy. *SIGMOD Record*, 27(4):21–24, December 1998.
- Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Boca Raton, FL, USA, 1 edition, 1996.
- Robin van Meteren and Maarten van Someren. Using Content-Based Filtering for Recommendation. In *Proceedings of the ECML 2000 Workshop: Machine Learning in New Information Age*, pages 47–56, Barcelona, Spain, May 2000.
- MICROSOFT-NET. *.NET: Driving Business Value with the Microsoft Platform*. Microsoft Corporation. URL <http://www.microsoft.com/Net/>. Access date: 05/01/2005.
- MICROSOFT-DTS. *Data Transformation Services (DTS)*. Microsoft Corporation. URL <http://www.microsoft.com/sql/evaluation/features/datatran.asp>. Access date: 06/01/2004.
- MICROSOFT-OLE-DB. *OLE DB*. Microsoft Corporation. URL <http://www.microsoft.com/data/oledb>. Access date: 06/06/2005.
- Bamshad Mobasher. Web Usage Mining and Personalization. In Munindar P. Singh, editor, *The Practical Handbook of Internet Computing*. Chapman & Hall/CRC, 2004. Chapter 15.
- Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic Personalization Based on Web Usage Mining. *Communications of the ACM*, 43(8):142–151, August 2000.
- Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective Personalization Based on Association Rule Discovery from Web Usage Data. In Ee-Peng Lim, editor, *Proceedings of the 3rd International Workshop on Web Information and Data Management, WIDM 2001*, Atlanta, GA, USA, 2001. ACM Press.
- MONDRIAN-OLAP. URL <http://mondrian.sourceforge.net/>. Access date: 04/19/2004.

- Trevor T. Moores and Gurpreet Dhillon. Do Privacy Seals in E-Commerce Really Work? *Communications of the ACM*, 46(12):265–271, December 2003.
- MYSQL. URL <http://www.mysql.com/>. Access date: 05/01/2005.
- Olfa Nasraoui. World Wide Web Personalization. In John Wang, editor, *Encyclopedia Of Data Warehousing and Mining*. Idea Group Publishing, 2005.
- Paul Newbold, William L. Carlson, and Betty Thorne. *Statistics for Business and Economics*. Prentice Hall, Upper Saddle River, NJ, USA, 5 edition, 2003.
- OMG. Object Management Group. URL <http://www.omg.org>. Access date: 06/04/2005.
- OMG-CWM. *Common Warehouse Metamodel Specification*. Object Management Group, 2001. URL <http://www.omg.org/docs/ad/01-02-01.pdf>. Access date: 06/18/2003.
- OECD-PRIVACY. OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 1980. URL http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_1_1_1_1,00.html. Access date: 11/13/2004.
- Michael Olan. Unit Testing: Test Early, Test Often. *Journal of Computing Sciences in Colleges*, 19(2):319–328, December 2003.
- OPENI. Open Intelligence. Open Source Web Application for OLAP Reporting. URL <http://openi.sourceforge.net/>. Access date: 08/28/2005.
- OS-COMMERCE. URL <http://www.oscommerce.com/>. Access date: 05/08/2005.
- Leonard Paas and Ton Kuijlen. Towards a General Definition of Customer Relationship Management. *Journal of Database Marketing*, 9(1):51–60, September 2001.
- Zidrina Pabarskaite. Decision Trees for Web Log Mining. *Intelligent Data Analysis*, 7(2): 141–154, 2003.
- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.
- Shan L. Pan and Jae-Nam Lee. Using e-CRM for a Unified View of the Customer. *Communications of the ACM*, 46(4):95–99, April 2003.
- Adrian Payne. The Multi-Channel Integration Process in Customer Relationship Management. White Paper, Cranfield School of Management, Cranfield University, Cranfield, UK, February 2003a. URL <http://www.insightexec.com/cgi-bin/library.cgi?action=detail&id=1957>. Access date: 03/08/2005.
- Adrian Payne. A Strategic Framework for Customer Relationship Management. White Paper, Cranfield School of Management, Cranfield University, Cranfield, UK, May 2003b. URL <http://www.insightexec.com/cgi-bin/library.cgi?action=detail&id=2254>. Access date: 03/05/2005.

- Gordon Paynter, Len Trigg, Eibe Frank, and Richard Kirkby. Attribute-Relation File Format (ARFF), 2002. URL <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>. Access date: 04/11/2005.
- Joe Peppard. Customer Relationship Management (CRM) in Financial Services. *European Management Journal*, 18(3):312–327, June 2000.
- Don Peppers and Martha Rogers. *The One-To-One Manager. Real-World Lessons in Customer Relationship Management*. Currency Doubleday, New York, NY, USA, 1 edition, 1999.
- PHP. URL <http://www.php.net>. Access date: 12/12/2004.
- Gregory Piatetsky-Shapiro, Ron Brachman, Tom Khabaza, Willi Kloesgen, and Evangelos Simoudis. An Overview of Issues in Developing Industrial Data Mining and Knowledge Discovery Applications. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD 1996*, pages 89–95, Portland, Oregon, USA, August 1996. AAAI Press.
- Dimitrios Pierrakos, Georgios Paliouras, Christos Papatheodorou, and Constantine D. Spyropoulos. Web Usage Mining as a Tool for Personalization: A Survey. *User Modeling and User-Adapted Interaction*, 13(4):311–372, November 2003.
- Frank Piller and Detlef Schoder. Mass Customization und Electronic Commerce. *Zeitschrift für Betriebswirtschaft*, 69(10):1111–1136, 1999.
- James Pitkow. In Search of Reliable Usage Data on the WWW. In *Proceedings of the 6th International World Wide Web Conference, WWW 1997*, pages 451–463, Santa Clara, CA, USA, April 1997.
- John D. Poole. Model-Driven Architecture: Vision, Standards and Emerging Technologies. In *Proceedings of the Workshop on Metamodeling and Adaptive Object Models. European Conference on Object-Oriented Programming, ECOOP 2001*, Budapest, Hungary, June 2001. URL <http://www.cwmforum.org/Model-Driven%20Architecture.pdf>. Access date: 06/05/2005.
- John D. Poole, Dan Chang, Douglas Tolbert, and David Mellor. *Common Warehouse Metamodel. An Introduction to the Standard for Data Warehouse Integration*. Wiley, Indianapolis, IN, USA, 1 edition, 2002.
- John D. Poole, Dan Chang, Douglas Tolbert, and David Mellor. *Common Warehouse Metamodel. Developer's Guide*. Wiley, Indianapolis, IN, USA, 1 edition, 2003.
- Michael E. Porter. *Competitive Advantage: Creating and Sustaining Superior Performance*. Free Press, New York, NY, USA, 1 edition, 1985.
- Olivier Povel and Christophe Giraud-Carrier. SwissAnalyst: Data Mining without the Entry Ticket. In Max Bramer and Vladan Devedzic, editors, *Artificial Intelligence Applications And Innovations (IFIP 18th World Computer Congress, TC12 First International Conference on Artificial Intelligence Applications and Innovations AIAI-2004)*, pages 393–406. IFIP, Kluwer, 2004.
- PRICELINE. URL <http://www.priceline.com/>. Access date: 01/04/2005.

- John R. Punin, Mukkai S. Krishnamoorthy, and Mohammed J. Zaki. LOGML: Log Markup Language for Web Usage Mining. *Lecture Notes in Computer Science*, 2356:88–112, January 2002.
- Raymond Pyle. Electronic Commerce and the Internet. *Communications of the ACM*, 39(6): 22–23, June 1996.
- QXL. URL <http://www.qxl.com/>. Access date: 01/04/2005.
- R-PROJECT. The R Project for Statistical Computing. URL <http://www.r-project.org/>. Access date: 02/01/2005.
- Erhard Rahm and Thomas Stöhr. Data-Warehouse-Einsatz zur Web-Zugriffsanalyse. In Erhard Rahm and Gottfried Vossen, editors, *Web und Datenbanken. Konzepte, Architekturen, Anwendungen*, pages 335–362. Dpunkt Verlag, Heidelberg, Germany, 1 edition, 2002.
- Raghu Ramakrishnan and Johannes Gehrke. *Database Management Systems*. McGraw-Hill, Boston, MA, USA, 3 edition, 2003.
- Joseph Reagle and Lorrie Faith Cranor. The Platform for Privacy Preferences. *Communications of the ACM*, 42(2):48–55, February 1999.
- Frederick J. Riggins. A Framework for Identifying Web-Based Electronic Commerce Opportunities. *Journal of Organizational Computing and Electronic Commerce*, 9(4):297–310, 1999.
- Ronald L. Rivest. The MD5 Message-Digest Algorithm, 1992. URL <http://theory.lcs.mit.edu/~rivest/Rivest-MD5.txt>. Access date: 07/09/2004.
- Nicholas C. Romano Jr. and Jerry Fjermestad. Electronic Commerce Customer Relationship Management: A Research Agenda. *Information Technology and Management*, 4(2-3):233–258, April-July 2003.
- RULEQUEST. C5.0: An Informal Tutorial, 2004. URL <http://www.rulequest.com/see5-unix.html>. Access date: 04/11/2005.
- Laura Rush. E-Commerce Growth Spurred by Maturation, January 2004. URL <http://www.clickz.com/stats/markets/retailing/article.php/3303311>. Access date: 07/09/2004.
- Lourdes Salcedo, Olivier Beauvillain, Ian Fogg, Mark Mulligan, and Julian Smith. Market Forecast Report. European Commerce, 2003-2009, January 2004. URL <http://www.jupiterresearch.com/bin/item.pl/research:vision/91/id=95879>. Access date: 01/01/2005.
- Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Riedl. Recommender Systems for Large-Scale E-Commerce: Scalable Neighborhood Formation Using Clustering. In *Proceedings of the Fifth International Conference on Computer and Information Technology (ICCIT 2002)*, 2002.
- Ralf Schaarschmidt, Jan Nowitky, and Jens Lufter. Clickstream Warehousing für e-CRM: Neue Herausforderungen an die Datenhaltung? In Hans Ulrich Buhl, Andreas Huther, and Bernd Reitweisner, editors, *Information Age Economy, WI 2001*, pages 117–131, Augsburg, Germany, 2001. Physica-Verlag.

- J. Ben Schafer, Joseph A. Konstan, and John Riedl. E-Commerce Recommendation Applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, January/April 2001.
- Marcus Schögel and Achim Sauer. Multi-Channel Marketing – Die Königsdisziplin im CRM. *Thesis*, 19(1):26–31, 2002.
- Marcus Schögel, Inga Schmidt, and Achim Sauer. Multi-Channel Management im CRM – Prozessorientierung als zentrale Herausforderung. In Hajo Hippner and Klaus D. Wilde, editors, *Management von CRM Projekten. Handlungsempfehlungen und Branchenkonzepte*, pages 107–122. Gabler, 2004.
- Marcus Schögl and Inga Schmidt. E-CRM – Management von Kundenbeziehungen im Umfeld neuer Informations- und Kommunikationstechnologien. In Marcus Schögl and Inga Schmidt, editors, *E-CRM. Mit Informationstechnologien Kundenpotenziale nutzen*, pages 29–83. Symposium Publishing, 2002.
- Edith Schonberg, Thomas Cofino, Robert Hoch, Mark Podlaseck, and Susan L. Spraragen. Measuring Success. *Communications of the ACM*, 43(8):53–57, August 2000.
- Jörg Schumacher and Matthias Meyer. *Customer Relationship Management strukturiert dargestellt*. Springer, Heidelberg, Germany, 1 edition, 2004.
- Benjamin Scribner. CRM and the Retail Industry. White Paper, Benjamin Scribner Strategic Consulting, Washington, DC, USA, November 2001. URL http://www.bensplace.com/Employer/publications/CRM_Retail_Industry.pdf. Access date: 05/18/2005.
- Benjamin Scribner. CRM Primer. White Paper, Benjamin Scribner Strategic Consulting, Washington, DC, USA, September 2002. URL http://www.bensplace.com/Employer/publications/CRM_primer.pdf. Access date: 03/06/2005.
- Cyrus Shahabi and Farnoush Banaei-Kashani. A Framework for Efficient and Anonymous Web Usage Mining Based on Client-Side Tracking. In Ron Kohavi, Brij M. Masand, Myra Spiliopoulou, and Jaideep Srivastava, editors, *WebKDD 2001 – Mining Web Log Data Across All Customer Touch Points. Third International Workshop*, pages 113–144, San Francisco, CA, USA, August 2001. Springer.
- Emil Sit and Kevin Fu. Web Cookies: Not Just a Privacy Risk. *Communications of the ACM*, 44(9):120, September 2001.
- Sarah Spiekermann, Jens Grossklags, and Bettina Berendt. E-Privacy in 2nd Generation E-Commerce: Privacy Preferences versus actual Behavior. In Michael P. Wellman and Yoav Shoham, editors, *Proceedings of the 3rd ACM Conference on Electronic Commerce*, pages 38–47, Tampa, FL, USA, October 2001. ACM Press.
- Myra Spiliopoulou. Web Usage Mining for Web Site Evaluation. *Communications of the ACM*, 43(8):127–134, August 2000.
- Myra Spiliopoulou, Bamshad Mobasher, Bettina Berendt, and Miki Nakagawa. A Framework for the Evaluation of Session Reconstruction Heuristics in Web Usage Analysis. *INFORMS Journal on Computing*, 15(2):171–190, Spring 2003.

- George Spofford. *MDX Solutions With Microsoft SQL Server Analysis Services*. Wiley, Indianapolis, IN, USA, 1 edition, 2001.
- Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *SIGKDD Explorations*, 1(2):12–23, January 2000.
- Jaideep Srivastava, Jau-Hwang Wang, Ee-Peng Lim, and San-Yih Hwang. A Case for Analytical Customer Relationship Management. In Ming-Shan Cheng, Philip S. Yu, and Bing Liu, editors, *Advances in Knowledge Discovery and Data Mining. Proceedings of the 6th Pacific-Asia Conference, PAKDD 2002*, pages 14–27, Taipei, Taiwan, May 2002. Springer.
- Jaideep Srivastava, Prasanna Desikan, and Vipin Kumar. Web Mining – Concepts, Applications & Research Directions. In Hillol Kargupta, Anupam Joshi, Krishnamoorthy Sivakumar, and Yelena Yesha, editors, *Data Mining: Next Generation Challenges and Future Directions*, pages 51–71. AAAI Press, 2004.
- Merlin Stone, Matt Hobbs, and Mahnaz Khaleeli. Multichannel Customer Management: The Benefits and Challenges. *Journal of Database Marketing*, 10(1):39–52, September 2002.
- Michael Stonebraker. Too Much Middleware. *SIGMOD Record*, 31(1):97–106, March 2002.
- Patrick Sue and Paul Morin. A Strategic Framework for CRM. White Paper, LGS Group Inc., February 2001. URL <http://www.insightexec.com/cgi-bin/library.cgi?action=detail&id=1385>. Access date: 03/10/2005.
- SUN-J2EE. *Java 2 Platform, Enterprise Edition (J2EE)*. Sun Microsystems. URL <http://java.sun.com/j2ee/>. Access date: 05/01/2005.
- Mark Sweiger, Mark R. Madsen, Jimmy Langston, and Howard Lombard. *Clickstream Data Warehousing*. Wiley, Indianapolis, IN, USA, 1 edition, 2002.
- Pang-Ning Tan and Vipin Kumar. Discovery of Web Robot Sessions Based on their Navigational Patterns. *Data Mining and Knowledge Discovery*, 6(1):9–35, January 2002.
- Michael Thess. Xeli’s Intro. Introduction to Xelopes. White Paper, Prudsys AG, May 2004. URL <http://www.xelopes.de>. Access date: 07/09/2004.
- Michael Thess and Michael Bolotnicov. *XELOPES Library Documentation Version 1.2.5*. Prudsys AG, November 2004. URL <http://www.xelopes.de>. Access date: 07/09/2004.
- John Thorp. *The Information Paradox. Realizing the Business Benefits of Information Technology*. McGraw-Hill, revised edition, New York, NY, USA 2003.
- Luba Torlina, Peter Seddon, and Brian Corbitt. Attributable Characteristics of Goods and Services in Electronic Commerce. In *Global Networked Organizations. Proceedings of the Twelfth International Bled Electronic Commerce Conference*, Bled, Slovenia, June 1999.
- Juan Trujillo and Sergio Luján-Mora. A UML Based Approach for Modeling ETL Processes in Data Warehouses. *Lecture Notes in Computer Science*, 2813:307–320, January 2003.
- TRUSTE. URL <http://www.truste.org/>. Access date: 11/14/2004.

- Efraim Turban and David King. *Introduction to E-Commerce*. Prentice Hall, Upper Saddle River, NJ, USA, 1 edition, 2003.
- TYPO3. URL <http://www.typo3.com/>. Access date: 05/08/2005.
- USCB-SURVEY. *Quarterly Retail E-Commerce Sales*. United States Census Bureau. URL <http://www.census.gov/mrts/www/ecom.html>. Access date: 12/30/2004.
- WEKA. *Weka Machine Learning Project*. The University of Waikato. URL <http://www.cs.waikato.ac.nz/~ml/index.html>. Access date: 06/06/2005.
- Panos Vassiliadis, Zografoula Vagenas, Spiros Skiadopoulos, Nikos Karayannidis, and Timos Sellis. ARKTOS: Towards the Modeling, Design, Control and Execution of ETL Processes. *Information Systems*, 26(8):537–561, December 2001.
- Panos Vassiliadis, Alkis Simitsis, and Spiros Skiadopoulos. Conceptual Modeling for ETL Processes. In *Proceedings of the 5th ACM International Workshop on Data Warehousing and OLAP*, pages 14–21, McLean, VA, USA, November 2002. ACM Press.
- Panos Vassiliadis, Alkis Simitsis, Panos Georgantas, and Manolis Terrovitis. A Framework for the Design of ETL Scenarios. In Johann Eder and Michele Missikoff, editors, *Advanced Information Systems Engineering. 15th International Conference, CAiSE 2003*, pages 520–535, Klagenfurt, Austria, June 2003. Springer.
- Panos Vassiliadis, Alkis Simitsis, Panos Georgantas, Manolis Terrovitis, and Spiros Skiadopoulos. A Generic and Customizable Framework for the Design of ETL Scenarios. *Information Systems*, 2005. Article in press.
- John Verzani. *Using R for Introductory Statistics*. Chapman & Hall/CRC, Boca Raton, FL, USA, 1 edition, 2005.
- Alexander J. Vincent. *JavaScript Developers’s Dictionary*. Sams Publishing, Indianapolis, IN, USA, 1 edition, 2002.
- Eugene Volokh. Personalization and Privacy. *Communications of the ACM*, 43(8):84–88, August 2000.
- Nir Vulkan. *The Economics of Electronic Commerce: A Strategic Guide to Understanding and Designing the Online Marketplace*. Princeton University Press, Princeton, NJ, USA, 1 edition, 2003.
- W3C-HTTP. *HTTP – Hypertext Transfer Protocol*, 2000. URL <http://www.w3.org/Protocols/>. Access date: 12/12/2004.
- W3C-P3P. *Platform for Privacy Preferences (P3P) Project*, 2001. URL <http://www.w3.org/P3P/>. Access date: 11/14/2004.
- W3C-XHTML. *XHTML 1.0 The Extensible Hypertext Markup Language (Second Edition). A Reformulation of HTML 4 in XML 1.0*, 2002. URL <http://www.w3.org/TR/xhtml1/>. Access date: 12/12/2004.
- W3C-XML. *Extensible Markup Language (XML) 1.0 (Third Edition)*, 2004. URL <http://www.w3.org/TR/REC-xml/>. Access date: 12/12/2004.

- E. Garrison Walters. *The Essential Guide to Computing. The Story of Information Technology*. Prentice Hall, Upper Saddle River, NJ, USA, 1 edition, 2001.
- Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Report 2004/199, Cryptology ePrint Archive, 2004. URL <http://eprint.iacr.org/2004/199.pdf>. Access date: 07/23/2005.
- WEBTRUST. URL <http://www.cpawebtrust.org/>. Access date: 11/14/2004.
- J. Christopher Westland and Theodore H. K. Clark. *Global Electronic Commerce: Theory and Case Studies*. The MIT Press, Cambridge, MA, USA, 1 edition, 1999.
- WHAT-IS. URL <http://whatis.techtarget.com/>. Access date: 04/19/2004.
- WIKIPEDIA. URL http://en.wikipedia.org/wiki/Ideal_type. Access date: 06/18/2005.
- Klaus D. Wilde and Hajo Hippner. Web Mining. Informationen für das E-Business. Absatzwirtschaft Studie, Katholische Universität Eichstätt-Ingolstadt, Ingolstadt, Germany, 2002. URL <http://www.crm-competence.com/>. Access date: 05/05/2005.
- Linda Wilkins, Paula M. C. Swatman, and Tanya Castleman. What's in a Name? Conceptual Issues in Defining Electronic Commerce. In *Proceedings of the 8th European Conference on Information Systems, ECIS 2000*, Vienna, Austria, July 2000.
- Ian H. Witten and Eibe Frank. *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, CA, USA, 2 edition, 2005.
- Sherman Wood. Optimizing Mondrian Performance, July 2005. URL http://mondrian.sourceforge.net/head/optimizing_performance.html. Access date: 08/24/2005.
- Osmar R. Zaïane, Man Xin, and Jiawei Han. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In *Proceedings of the Advances in Digital Libraries Conference, ADL 1998*, pages 19–29, Santa Barbara, CA, USA, April 1998. IEEE Computer Society.
- Vladimir Zwass. Electronic Commerce: Structures and Issues. *International Journal of Electronic Commerce*, 1(1):3–23, Fall 1996.
- Detlev Zwick and Nikhilesh Dholakia. Contrasting European and American Approaches to Privacy in Electronic Markets: Property Right versus Civil Right. *Electronic Markets*, 11(2): 116–120, February 2001.

INDEX

- AbstractDimension, 86, 87
- aggregate tables, 110
- analytical CRM, 18
- application phase, 129
- association rule analysis, 28
- assortment dimension, 158
- atom, 75
 - semantic atom, 82
- attribute, 62, 63
 - categorical attribute, 63, 65
 - discrete attribute, 63
 - numeric attribute, 63, 65
 - ordinal attribute, 63, 65
- B2B EC, 8
- B2C EC, 9
- bot, *see* Web robot
- C2B EC, 117
- C2C EC, 117
- cache, 109
- caching problem, 36, 39
- categorical attribute, 63, 65
- CategoricalAttribute, 65
- classification, 31
- clickstream mart, 106, 166, 175
- closed loop, 24, 101
- cluster analysis, 31
- collaborative CRM, 18
- collection limitation principle, 118
- Common Warehouse Meta-Model, *see* CWM
- composition
 - horizontal composition, 78
 - vertical composition, 75
- content dimension, 159
- cookies, 36, 38
- cost leadership strategy, 12
- CRISP-DM, 45
- CRM, 14
 - analytical CRM, 18
 - collaborative CRM, 18
 - electronic CRM, *see* ECRM
 - functional chain, 16, 124
 - operational CRM, 18
 - positive feedback cycle of CRM, 15
 - process alignment framework, 127
 - taxonomy, 18
- cross-selling, 15
- cubing
 - then mining, 170
 - while mining, 171
- customer
 - acquisition, 15
 - mart, 160
 - profitability, 15
 - relationship management, *see* CRM
 - retention, 16
 - satisfaction, 16
 - segmentation, 16
- CWM, 46
 - data mining package, 49
 - OLAP package, 49
 - transformation package, 49
- data collection, 34
- data dictionary, 50
- data mart, 104
 - clickstream mart, 106, 166, 175
 - customer mart, 160
 - order line mart, 106, 164, 176
 - order mart, 104, 162
 - session mart, 106, 166
- data matrix, 64, 66, 68
- data mining
 - application phase, 129
 - association rule analysis, 28
 - classification, 31
 - cluster analysis, 31
 - embedded data mining, 44
 - layer, 60
 - path analysis, 30
 - regression, 31
 - sequence analysis, 29
 - statistical analysis, 28
 - training phase, 129
- data source, 133
- data storage layer, 59

- data transformations, *see* transformations
- database streams, 71, 133
- date dimension, 149, 173
- decomposable, 142
- decomposition
 - horizontal decomposition, 75
 - vertical decomposition, 74
- degenerate
 - dimension, 84, 89
 - star schema, 89, 158
- DegenerateDimension, 87, 88
- denormalization, 102
- deployment phase, 32, 44
- differentiation strategy, 12
- dimension, 84, 86
 - assortment dimension, 158
 - collapsed dimension, 110
 - content dimension, 159
 - date dimension, 149, 173
 - degenerate dimension, 84, 89
 - lost dimension, 110
 - product dimension, 160
 - referrer dimension, 156, 175
 - regular dimension, 84, 88
 - table, 104, 149
 - time dimension, 153, 174
- Dimension, 87, 88
- dimensional modeling, 82
- discrete attribute, 63
- drill-through, 171

- e-business, *see* EC
- EC, 7
 - business-to-business, *see* B2B EC
 - business-to-consumer, *see* B2C EC
 - channel, 19
 - consumer-to-business, *see* C2B EC
 - consumer-to-consumer, *see* C2C EC
 - matrix, 8
- ECRM, 20
 - research areas, 21
- EDI, 7
- electronic CRM, *see* ECRM
- electronic commerce, *see* EC
- electronic data interchange, *see* EDI
- embedded data mining, 44
- ETL, 56
 - layer, 60
 - process, 90
 - transformation, *see* transformation
- fact table, 84, 89
- filter streams, 71, 133
- flat file streams, 70, 133
- focus strategy, 12
- foreign key constraints, 110
- forward auction, 117
- frequent item set, 99
 - graph, 100
- functional chain of CRM, 16, 124

- gradual ETL approach, 91
- GUI, 105

- horizontal
 - composition, 78
 - decomposition, 75
- HybridDimension, 87, 89

- indecomposable, 75, 142
- indexes, 110
- individual participation principle, 118
- information supply chain, 47
- instantaneous ETL approach, 91
- invalid values, 67
- inverse meta-data, 67

- JDM API, 49
- JOLAP, 49
- JPivot, 55, 177

- KDD Cup 2000, 95

- LAMP environment, 38
- LogFileStream, 71, 134
- logging, 38
- LOORDSM, 82, 87, 111

- mapping
 - multiple-to-multiple, 77, 142
 - multiple-to-one, 76, 142
 - one-to-multiple, 76, 142
 - one-to-one, 76, 141
- mass customization, 10
- MD5, 89
- MDX, 102, 170, 172
- measure, 92, 176
- memory streams, 70, 132
- meta-data, 62, 64, 66

- date-dimension, 150
- inverse meta-data, 67
- referrer dimension, 156
- role meta-data, 88
- time dimension, 153
- transformation, 73
- mining
 - attribute, *see* attribute
 - basis, 129
 - then cubing, 170
 - vector, *see* vector
- MiningArffStream, 70, 134
- MiningArrayStream, 70, 133
- MiningAttribute, 65
- MiningC50Stream, 71, 134
- MiningCollectionStream, 70, 132
- MiningCsvStream, 70, 134
- MiningDataSpecification, 51, 65
- MiningFileStream, 70, 134
- MiningFilterStream, 71, 136
- MiningInputStream, 69, 131
- MiningQueryMdxStream, 171
- MiningQuerySqlStream, 71, 135
- MiningSqlStream, 71, 135
- MiningStreamTransformer, 74
- MiningTableSqlStream, 71, 88, 109, 135
- MiningTokenizerStream, 70, 134
- MiningTransformationActivity, 79, 145
- MiningTransformationStep, 79, 144
- MiningTransformationStream, 71, 136
- MiningTransformer, 74, 77
- MiningUpdatableSqlSource, 71, 137
- MiningVectorFilterStream, 71, 136
- missing values, 62, 63, 70
- Mondrian, 55
- multi-channel
 - environment, 19
 - integration, 18, 126
 - management, 126
- MultidimensionalStream, 72, 136
- multiple-to-multiple mapping, 77, 142
- multiple-to-one mapping, 76, 142
- MultipleToMultipleMapping, 77, 143
- network packet sniffers, 40
- numeric attribute, 63, 65
- NumericAttribute, 65
- OLAP layer, 60
- OLAP server, 44
- one-to-multiple mapping, 76, 142
- one-to-one mapping, 76, 141
- OneToMultipleMapping, 76, 142
- OneToOneMapping, 76, 141
- openness principle, 118
- operational CRM, 18
- order line mart, 106, 164, 176
- order mart, 104, 162
- ordinal attribute, 63, 65
- OrdinalAttribute, 65
- P3P, 120
- page-tagging, 41
- path analysis, 30
- pattern analysis phase, 31, 44
- pattern discovery phase, 28, 44
- performance issues, 109
- personalization, 33
 - real-time personalization, 97
- physical product, 10
- PMML, 50
 - data dictionary, 50
- PmmlPresentable, 86, 141
- positive feedback cycle of CRM, 15
- preprocessing phase, 27, 42
- primary key mapping, 83
- PrimaryKeyGenerator, 89, 110
- privacy, 118
- product
 - dimension, 160
 - physical product, 10
 - view history, 98
 - virtual product, 10
- projection, 85
- purpose specification principle, 96, 118
- recommendation engine, 108
- recommender systems, 96
- referrer dimension, 156, 175
- regression, 31

- regular dimension, 84, 88
- regular transformation, 73
- RegularDimension, 87, 88
- remote agent, 41
- reverse auction, 117
- reverse proxy, 41
- ROLAP, 54
- role meta-data, 88

- schema file, 60, 102, 173
 - clickstream mart, 175
 - date dimension, 173
 - order line mart, 176
 - referrer dimension, 175
 - time dimension, 174
- sequence analysis, 29
- session, 27, 35
 - mart, 106, 166
 - timeout, 39
 - tracking, 38
- special transformation, 74
- SQL/MM, 50
- standards, 46
 - CWM, 46
 - JDM API, 49
 - JOLAP, 49
 - PMML, 50
 - SQL/MM, 50
- star schema, 84, 89
 - assortment dimension, 158
 - clickstream mart, 168
 - content dimension, 159
 - customer mart, 161
 - degenerate star schema, 89, 158
 - order line mart, 164
 - order mart, 162
 - product dimension, 160
 - session mart, 166
- StarSchema, 87, 89, 104
- statistical analysis, 28
- stream, 66, 69
 - prototype, 131
- streams, 68–72
 - database streams, 71, 133
 - filter streams, 71, 133
 - flat file streams, 70, 133
 - memory streams, 70, 132
- StreamTokenizer, 70

- template, 97
- time dimension, 153, 174
- training phase, 129
- transformation, 40, 73
 - activity, 79, 145
 - composition of transformations, 78
 - concatenation of transformations, 77
 - ETL transformation, 40
 - meta-data transformation, 73
 - modeling, 72–81
 - nesting of transformations, 77
 - raw ETL transformation, 85
 - date dimension, 152
 - referrer dimension, 156
 - time dimension, 155
 - real-valued transformation, 73
 - regular transformation, 73
 - special transformation, 74
 - step, 79, 144
 - vector transformation, 73, 76
- TrivialPrimaryKey, 110
- trust, 118

- UpdatableStream, 70, 132
- URL rewriting, 39
- use limitation principle, 96, 118
- user session, *see* session

- vector, 64
 - filter, 137, 168
 - transformation, 73, 76
- VectorFilter, 138
- vertical
 - composition, 75
 - decomposition, 74
- virtual product, 10

- WAMP environment, 38
- Web application server, 37
 - logs, 37
- Web content mining, 26
- Web mining, 25
 - taxonomy, 26
- Web personalization, *see* personalization
- Web robot, 40, 168
- Web server logs, 35
- Web structure mining, 26
- Web usage analysis, 32
 - process, 32

Web usage mining, 27
 methods, *see* data mining
 process, 27, 43
WEKA, 51
WrapperDimension, 87, 89
WUSAN architecture, 51, 53
 analysis component, 54
 data access component, 52
 data warehousing component, 54
 population component, 52
WusanML, 80, 104, 147
 serialization, 86, 141
XELOPES, 51, 55