Preselection of Electronic Services by Given Business Service Based on Measuring Semantic Heterogeneity within the Application Area of Logistics

Der Wirtschaftswissenschaftlichen Fakultät der Universität Leipzig eingereichte DISSERTATION zur Erlangung des akademischen Grades Doktor rerum politicarum Dr. rer. pol. vorgelegt

von Rolf Kluge
geboren am 15.10.1980, in Leipzig

Gutachter: Prof. Dr. Bogdan Franczyk
Prof. Dr. Dieter Ehrenberg

Eingereicht am 30. Januar 2012
Verteidigt am 17. Juli 2012
Veröffentlicht am 05. November 2012
Bibliographic data

Title: Preselection of Electronic Services by Given Business Service Based on Measuring Semantic Heterogeneity within the Application Area of Logistics

Keywords: service preselection, service discovery, service matching, service correspondence, business service, electronic service, service heterogeneity, service-oriented architecture, SOA, logistics, logistics service engineering and management

Language: English
Author: Rolf Kluge
City: Leipzig
University: University of Leipzig
Faculty: Faculty of Economics and Management
Year: 2012
Pages: 316 (225)
Figures: 80
Tables: 59
References: 303
Appendix: 14
Preamble and dedication

This thesis had been written between 2008 and 2012. The first ideas for this thesis had been developed in 2007. In October 2006 I joined the Logistics-Service-Bus (LSB) project team at University of Leipzig (www.lsb-plattform.de). The objective of the research project was the conceptual development of a service-oriented IT-platform, which supports the business of small and medium size logistics companies in Middle Germany. In order to build a proper platform requirements of logistics companies in Middle Germany had been collected and analyzed. For this purpose, a structured method including questionnaire, analyzing techniques, and a tool had been developed by the project team. Eight logistics companies had been surveyed and analyzed in detail between October 2006 and October 2007. During the survey one of the managers of a logistics company told me that he would use services from a service-oriented IT-platform if the following questions are clearly answered: What does the platform provide? Does my company benefit from it? And, does the service fit for my company’s business? Out of this questions a rough idea about the matching of business- and IT-service was born. This idea had been published in a workshop paper in 2008 [Kluge et al. 2008] and pursued until completion.

During my thesis I was enrolled as a Ph. D. student at University of Leipzig Germany and at Macquarie University in Sydney Australia. I would like give my special thanks to my doctoral thesis supervisor and academic mentor Prof. Dr. Bogdan Franczyk, who gave me the opportunity writing this thesis and supported me with proper advises and fruitful discussions. Further, I would like to thank Prof. Dr. Leszek A. Maciaszek, who gave me (in conjunction with Prof. Dr. Bogdan Franczyk) the opportunity for joining the cotutelle program, which allowed me being enrolled at Macquarie University in Sydney. During my time at Macquarie University from December 2008 until July 2009 Prof. Dr. Leszek A. Maciaszek was a perfect host and provided me everything I needed. Thanks!

Further, I would like to thank Dr. André Ludwig and Dr. Thomas Hering. Dr. Thomas Hering was project leader of LSB I from 2006 to 2009 and promoted my thesis topic in the early stages. Later, Dr. André Ludwig, who was project leader of LSB II and InterLogGrid from 2009 to 2011, and co-founder of the logistics service engineering and management (LSEM) research group, supported my thesis directly. Furthermore, I would like to thank the LSEM research team Dr. Steffi Donath, Dr. Roman Belter, Christoph Augenstein, Christopher Klinkmüller, Robert Kunkel, Holger Müller, Stefan Mutke, and Martin Roth for all the discussions, the hard work and the fun we had. Thanks!

My infinite thanks are dedicated to my wife Katrin Kluge. She gave me power, confidence and a dose of pressure whenever I needed it. Further, I would like to thank my parents Christine Kluge and Dr. Gert Kluge, who supported me during my whole period of education. Thanks!
Table of contents

Preamble and dedication ........................................................................................................ III

List of figures ............................................................................................................................ VIII

List of tables .............................................................................................................................. XII

List of abbreviations ................................................................................................................ XV

1 Introduction .......................................................................................................................... 1
   1.1 Research area and research topic .................................................................................. 1
   1.2 Research methodology ................................................................................................. 2
   1.3 Structure of the thesis ................................................................................................. 5

2 Background, motivation and solution pathway ................................................................. 9
   2.1 Background .................................................................................................................. 9
   2.2 Problem description and motivation .......................................................................... 22
   2.3 Objectives and basic idea .......................................................................................... 25
   2.4 Assumptions ............................................................................................................... 26
   2.5 Requirements ............................................................................................................. 28
   2.6 Conceptual approach and solution pathway ............................................................ 29
   2.7 State of the art and related work .............................................................................. 31
   2.8 Summary .................................................................................................................... 33

3 Service representation ......................................................................................................... 34
   3.1 Requirements for service representation .................................................................... 34
   3.2 Models and ontologies ............................................................................................... 36
   3.3 Existing service representation approaches .............................................................. 48
   3.4 Conceptual model for service representation ............................................................ 71
   3.5 Summary .................................................................................................................... 80

4 Service modeling ................................................................................................................ 82
   4.1 Methodologies, methods, techniques, processes, activities, and tasks ..................... 82

- IV -
4.2 Ontology engineering methodologies ........................................... 84
4.3 Method for BSM and ESM modeling ........................................... 91
4.4 Case study conceptualization .................................................... 94
4.5 Tool support for service modeling ............................................. 99
4.6 Summary .................................................................................. 103

5 Service matching ........................................................................ 104
  5.1 Terminology and definitions .................................................... 104
  5.2 Service matching levels .......................................................... 107
  5.3 Basic matching algorithms ...................................................... 108
  5.4 Matching strategies ............................................................... 144
  5.5 BSM-ESM matching solution .................................................. 162
  5.6 Summary ................................................................................ 168

6 Visualization ............................................................................. 169
  6.1 Visualization of data and information ....................................... 169
  6.2 Data and information visualization methods .............................. 171
  6.3 Visualization method evaluation .............................................. 175
  6.4 Visualization method application and analysis ......................... 179
  6.5 Summary ................................................................................ 188

7 Tool support .............................................................................. 189
  7.1 Service preselection tool architecture ...................................... 189
  7.2 Service preselection tool functionality ...................................... 191
  7.3 Summary ................................................................................ 201

8 Validation .................................................................................. 202
  8.1 Use case scenarios ................................................................. 203
  8.2 Use case services .................................................................... 203
  8.3 Application and analysis ........................................................... 210
  8.4 Summary ................................................................................ 218
9 Conclusion and outlook .........................................................219
  9.1 Summary ........................................................................219
  9.2 Main contribution .........................................................220
  9.3 Discussion ......................................................................221
  9.4 Further research .........................................................223

References ..................................................................................226

Appendix A ................................................................. 245
Appendix B ................................................................. 246
Appendix C ................................................................. 247
Appendix D ................................................................. 248
Appendix E ................................................................. 250
  BS Transport Management - OrderInquiry ................................250
Appendix F ................................................................. 255
  ES CRM – CustomerEntry ..................................................255
Appendix G ................................................................. 257
  ES ERP – SendInvoice ......................................................257
Appendix H ................................................................. 262
  ES SAP TM – InvoiceManagement .....................................262
Appendix I ................................................................. 266
  ES SAP TM – PlanTransportationActivities ..........................266
Appendix J ................................................................. 270
  ES SAP TM – OrderReceipt Order Management .....................270
Appendix K ................................................................. 277
  ES SAP TM – Order Management renamed .........................277
Appendix L..................................................................................................................284
  ES SAP TM – Order Management restructured .............................................284

Appendix M..............................................................................................................290
  ES SAP TM – Order Management missing feature .....................................290

Appendix N..............................................................................................................296

Curriculum vitae .................................................................................................298

Selbstständigkeitserklärung .............................................................................300
List of figures

Figure 1.1 Information Systems Research Framework [Hevner/March 2003] 6
Figure 1.2 Structure of this thesis 8
Figure 2.1 Roles and interactions in SOA (cf. [Khoshafian 2006, Papazoglou/Heuvel 2007] and others) 12
Figure 2.2 LSEM service lifecycle 18
Figure 2.3 Typical evaluation process (cf. [Markovic/Karrenbrock 2007]) 24
Figure 2.4 Conceptual approach 30
Figure 2.5 Relation keyword based search, the approach, and semantic service discovery 32
Figure 3.1 UML Class diagram example 39
Figure 3.2 MOF four layered meta-data architecture 41
Figure 3.3 Ontology example modeled with Protégé [Protege] 43
Figure 3.4 Ontology types (cf. [Guarino 1997]) 45
Figure 3.5 Molecular modeling examples [Shostack 1993] 49
Figure 3.6 Service Blueprinting Example 51
Figure 3.7 SADT activity [Congram/Epelman 1995] 52
Figure 3.8 OBELIX service value ontology 54
Figure 3.9 OBELIX service offering ontology 54
Figure 3.10 OBELIX service element visualization 55
Figure 3.11 WSDL 2.0 specification 56
Figure 3.12 Web Service semantics with WSDL-S (cf. [World Wide Web Consortium (W3C) 2005]) 58
Figure 3.13 Main elements of OWL-S (cf. [Martin et al. 2004]) 59
Figure 3.14 USDL modules (arrows represent dependencies) [Barros et al. 2011d] 65
Figure 3.15 Relation between SCMM, BSM and ESM according to MOF 71
Figure 4.1 Relation between methodology, method, technique, process, activity and task [Gómez-Pérez et al. 2004] 83
Figure 4.2 Reference activities 85
Figure 4.3 METHONTOMETRY activities and life cycle
[cf. Gómez-Pérez et al. 2004] 87
Figure 4.4 Essential process model in ontology development
(cf. [Gómez-Pérez et al. 2004]) 87
Figure 4.5 Tasks of the conceptualization activity according to METHONTOMETRY [Fernández-López et al. 1997] 91
Figure 4.6 Conceptualization tasks for BSM/ESM modeling 94
Figure 4.7 Ecore kernel elements 100
Figure 4.8 SCMM in EMF tree-based Ecore editor (partly) 101
Figure 4.9 Order management ESM using tree-based Ecore editor (partly) 103
Figure 5.1 Service matching process (cf. [Euzenat/Shvaiko 2007]) 104
Figure 5.2. Internal structure example 130
Figure 5.3 Part of the XML Schemal datatype hierarchy
Figure 5.4 Fragment of a service model 136
Figure 5.5 Similarity based on relational structure (example) 143
Figure 5.6 Sequential composition of matchers 145
Figure 5.7 The matching process including similarity matrix and alignment extraction 146
Figure 5.8 Sequential composition of matching processes though similarity matrix 146
Figure 5.9 Parallel composition of matchers 147
Figure 5.10 Parallel composition of matchers based on similarity matrix 147
Figure 5.11 Two typical ontologies showing referential cycles (simplified, i.e. no relation names due lack of space) 153
Figure 5.12 Iterative computation of similarity function 153
Figure 5.13 Possible node-edge combinations including weights for similarity flooding algorithm 155
Figure 6.1 Periodic table of visualization methods [Lengler/Eppler] 171
Figure 6.2 Bar chart for a single BS-ES matching results set on service level and service concept level 181
Figure 6.3 Bar chart for a single BS-ES matching results set on service entity level 182
Figure 6.4 Scatter plot comparison of two BS-ES matching result set on service level, service concept level, and service entity level 183
Figure 6.5 Continuum comparison of multiple BS-ES matching result sets on service level 185
Figure 6.6 Radar chart comparison of multiple BS-ES combinations on service concept level 186
Figure 6.7 Radar chart comparison of multiple BS-ES combinations on service entity level 187
Figure 7.1 Tool architecture 190
Figure 7.2 Three areas of the service preselection tool 192
Figure 7.3 Loading and selecting SCMM based models 193
Figure 7.4 Tree View 194
Figure 7.5 Table view 195
Figure 7.6 Bar view (Overall) 196
Figure 7.7 Bar view (AbstractService) 197
Figure 7.8 Scatter plot (Overall) 198
Figure 7.9 Scatter plot (OntologyConcept) 198
Figure 7.10 Radar chart (Overall) 199
Figure 7.11 Radar chart (OntologyConcept) 200
Figure 7.12 Weight and threshold configuration view 201
Figure 8.1 Business service: OrderInquiry 205
Figure 8.2 Electronic service: CustomerEntry 205
Figure 8.3 Electronic service: SendInvoice 206
Figure 8.4 Electronic service: SendCustomerInvoice 207
Figure 8.5 Electronic service: PlanTransportationActivities 207
Figure 8.6 Electronic service: OrderReceive 208
Figure 8.7 Electronic service: OrderReceive (renamed) 209
Figure 8.8 Electronic service: OrderReceive (restructured) 210
Figure 8.9 Electronic service: OrderReceive (modified features) 210
Figure 8.10 Overall matching result visualization (ES from different domains) 212
Figure 8.11 Matching results on service concept level
BS Inquiry : ES CustomerEntry (green color)
Figure 8.12 Overall matching result visualization (different ES from same domain) 212

Figure 8.13 Matching results on service concept level
BS Inquiry : ES SendCustomerInvoice (green color)
BS Inquiry : ES PlanTransportationActivities (blue color)
BS Inquiry : ES OrderReceipt (red color) 214

Figure 8.14 Matching results on OntologyConcept level
BS Inquiry : ES SendCustomerInvoice (green color)
BS Inquiry : ES PlanTransportationActivities (blue color)
BS Inquiry : ES OrderReceipt (red color) 214

Figure 8.15 Scatter plot view of matching results of
BS Inquiry : ES OrderReceipt (x-axis) and
BS Inquiry : ES OrderReceipt renamed (y-axis) 215

Figure 8.16 Scatter plot view of matching results of
BS Inquiry : ES OrderReceipt (x-axis) and
BS Inquiry : ES OrderReceipt restucture (y-axis) 216

Figure 8.17 Matching results on OntologyConcept level
BS Inquiry : ES OrderReceipt (red color)
BS Inquiry : ES OrderReceipt mod. feature (blue color) 218
List of tables

Table 1.1 Design-Science Research Guidelines [Hevner et al. 2004] 3
Table 2.1 Criteria for business and electronic services 16
Table 3.1 Languages of existing service representation approaches 67
Table 3.2 Comparison of basic functional aspects and their occurrence in existing service representation approaches (1/2) 70
Table 3.3 Comparison of basic functional aspects and their occurrence in existing service representation approaches (2/2) 70
Table 4.1 Service definitions and taxonomy 95
Table 4.2 Stakeholder definition 96
Table 4.3 Action definition 96
Table 4.4 Resource definition 97
Table 4.5 OntologyConcept definition 97
Table 4.6 ObjectProperty definition 98
Table 4.7 DataProperites definition 99
Table 5.1 String equality example 112
Table 5.2 Hamming distance dissimilarity (as a similarity) example 112
Table 5.3 Substring test example 113
Table 5.4 Substring similarity example 114
Table 5.5 Normalized Levenshtein distance (as a similarity) example 115
Table 5.6 Jaro-Winkler measure example 116
Table 5.7 Cosine similarity example 118
Table 5.8 Variants of the term "transport order" (adopted from [Maynard 2000]) 119
Table 5.9 Token based cosine synonym similarity example 124
Table 5.10 Example of token based cosine synonym similarity with domain filter and suffix stemming 125
Table 5.11 Gloss overlap example 127
Table 5.12 Weighted gloss overlap 128
Table 5.13 Geometric mean values with the combination of TBCS and WGO 129
Table 5.14 Conversion proximity between row and column datatype 131
Table 5.15 Datatype compatibility matrix

Table 5.16 Data property comparison based on datatypes

Table 5.17 Cardinality compatibility [Lee et al. 2002]

Table 5.18 Cardinality similarity measure

Table 5.19 Arithmetic means of datatype comparison and cardinality similarity

Table 5.20 Features for comparing service entities

Table 5.21 Results of the normalized structural topological dissimilarity for the taxonomy in Figure 5.4

Table 5.22 Results of the normalized Wu-Palmer similarity for the taxonomy in Figure 5.4

Table 5.23 Results of the upward cotopic similarity for the taxonomy in Figure 5.4

Table 5.24 Maximum aggregation

Table 5.25 Weight product aggregation with $w_1 = w_2 = 0.5$

Table 5.26 Normalized Manhattan distance as a similarity based on dimension mentioned above

Table 5.27 Normalized Euclidean distance as a similarity based on dimension mentioned above

Table 5.28 Normalized weight sum based on Manhattan distance (as a similarity) with $w_{\text{label}} = 3/4$, $w_{\text{comments}} = 1/4$

Table 5.29 Similarity flooding first iteration ($\sigma^1$)

Table 5.30 $\sigma^1$ similarity flooding (normalized)

Table 5.31 $\sigma^{20}$ similarity flooding (normalized)

Table 5.32 OLA initial concept similarity values ($\sigma^C_I$)

Table 5.33 OLA initial property similarity values ($\sigma^P_I$)

Table 5.34 $\sigma^C_I$ OLA concept similarity results

Table 5.35 $\sigma^P_I$ OLA property similarity results

Table 5.36 $\sigma^{10}_I$ OLA concept similarity results

Table 5.37 $\sigma^{10}_P$ OLA property similarity results

Table 5.38 Matching techniques on service entity property level including weights for service entity matching

Table 6.1 Selected visualization methods
Table 6.2 Nested table for single BS-ES matching result set on service level, service concept level and service entity level 180
Table 6.3 Sample result set of two BS-ES combinations 183
Table 6.4 Sample overall matching result ten BS-ES combinations 185
Table 6.5 Sample service concept matching result of five BS-ES combinations 186
Table 6.6 Sample maximum results from ConcreteService matching between one BS and five ES 187
Table 8.1 Overall and service concept level matching results (ES from different domains) 211
Table 8.2 Overall and service concept level matching results (different ES from same domain) 213
Table 8.3 Overall and service concept level matching results (similar ES from logistics domain) 216
List of abbreviations

AI      artificial intelligence
BPMN   Business Process Model and Notation
BS     business services
BSD    Business Service Deployment
BSE    Business Service Engineering
BSO    Business Service Operation
BST    Business Service Termination
BSM    business service model
CIM    Computation Independent Model
CMM    Capability Maturity Model
CMMI   Capability Maturity Model Integration
DAML-S DARPA Agent Markup Language for Services
DARPA  Defense Advanced Research Projects Agency
DL     Description logic
ebXML  Electronic Business using eXtensible Markup Language
EMF    Eclipse Modeling Framework
ES     electronic service
ESD    Electronic Service Development
ESE    Electronic Service Engineering
ESM    electronic service model
ESO    Electronic Service Operation
EST    Electronic Service Termination
IDE    integrated development environments
IOPE   inputs, outputs, preconditions and effects
KIF    Knowledge Interchange Format
KM     Knowledge Machine
LSEM   logistics service engineering and management
MDA    Model Driven Architecture
MOF    Meta Object Facility
OBELIX Ontology-Based ELe ctronic Integration of compleX products and value chains.
OCL    Object Constraint Language
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OKBC</td>
<td>Open Knowledge Base Connectivity</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>OWL DL</td>
<td>OWL Description Logic</td>
</tr>
<tr>
<td>PERT</td>
<td>Program Evaluation and Review Technique</td>
</tr>
<tr>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>SADT</td>
<td>Structured Analysis and Design Technique</td>
</tr>
<tr>
<td>SCMM</td>
<td>service concept meta-model</td>
</tr>
<tr>
<td>SOA</td>
<td>service-oriented architecture</td>
</tr>
<tr>
<td>SPICE</td>
<td>Software Process Improvement and Capability Determination</td>
</tr>
<tr>
<td>SUMO</td>
<td>Suggested Upper Merged Ontology</td>
</tr>
<tr>
<td>SWRL</td>
<td>Semantic Web Rule Language</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>UNSPSC</td>
<td>United Nations Standard Products and Services Code</td>
</tr>
<tr>
<td>USDL</td>
<td>Unified Service Description Language</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>WSMF</td>
<td>Web Service Modeling Framework</td>
</tr>
<tr>
<td>WSML</td>
<td>Web Service Modeling Language</td>
</tr>
<tr>
<td>WSMO</td>
<td>Web Service Modeling Ontology</td>
</tr>
<tr>
<td>WSMX</td>
<td>Web Service Execution Environment</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
<tr>
<td>XMI</td>
<td>XML Metadata Interchange</td>
</tr>
</tbody>
</table>
1 Introduction

This chapter provides the thesis' general introduction. First, an overview of the research area is given and one particular topic within this area is introduced in subchapter 1.1. Next, subchapter 1.2 discusses the research methodology. Lastly, the structure of the thesis is presented in subchapter 1.3.

1.1 Research area and research topic

Companies are constantly faced with changing market conditions and threats, new competitive pressures in terms of cost, time, flexibility, and ever changing regulations that require compliance. On this basis, companies outsource internal applications to external service providers in order to focus on the growth of their core activities and competencies. Outsourcing includes business functions such as packaging or customer support, but also infrastructural functions in particular computing systems such as enterprise software applications. This led to new markets and business models for providing business and computing functions.

With the advent of service orientation an important design paradigm was established that significantly eases outsourcing. Service orientation utilizes services as basic abstractions of business and computing functions. Services are defined as self-contained, loosely coupled entities that encapsulate a limited piece of functionality. Services are reusable, are able to be composed, and they provide a well-defined external interface. Service orientation as a design paradigm for business and computing functions enables enterprises to integrate externally provided services into internal processes easily and promises a number of benefits – among them flexible re-configuration, dynamic binding, easy access to heterogeneous resources and processes, transparency across implementation details and last but not least a relatively stable set of standards for aspects such as interface, orchestration, choreography or contracting.

Depending on the abstracted functions, services can be differentiated into business and electronic services. A business service (BS) encapsulates business concerns and capabilities. An electronic service (ES) encapsulates computing systems, information systems and software applications.

Research on service orientation has been conducted in various directions. An area which received insufficient support so far is the alignment of business and electronic services. Business and electronic services are naturally interrelated. Electronic services provide fundamental support for business services by means of data, information and their processing. However, in environments with a high number of business services and usually a high number of electronic services the decision on which electronic service provides the most suitable support for a certain business service is not a trivial task.
Given one particular business services and a high number of electronic services, this thesis is dedicated to the question: How to find electronic services that support that business service? This question represents the research topic addressed by the thesis. However, details about this topic are presented later on in chapter 2.

1.2 Research methodology

In order to comply with the requirements on adequate research quality, the thesis is conducted by the design science research paradigm [Hevner et al. 2004]. Information systems research is characterized by the disciplines behavioral science and design science. By considering the confluence of people, organizations, and technology both paradigms are foundational in information systems research. However, behavioral science and design science have different roots and aims. The behavioral science paradigm has its roots in natural science research. It seeks in empirically developing justified theories and hypotheses that explain or predict human and organizational phenomena by considering design, implementation, analysis and use of information systems. Those theories provide researchers and practitioners information about the interactions of people, technology and organizations that have to be managed for effective application of information systems [Hevner/March 2003]. In contrast, design science with its roots in engineering and in “the science of the artificial” [Simon 1996] seeks in the creation of artifacts. Artifacts are innovations that are embodied by ideas, practices, products or technical capabilities, which help to design, implement, analyze or use information systems effectively [Hevner/March 2003]. [Hevner et al. 2004] contrasts both paradigms by “The behavioral science paradigm seeks to find ‘what is true’. In contrast, the design science paradigm seeks to create ‘what is effective’”.

Research of this thesis follows the design science paradigm. Thus, this thesis seeks in the creation of artifacts. [Hevner/March 2003] defines four types of artifacts: constructs, models, methods and instantiations.

- Constructs provide language for definition and communication of problems and solutions.
- Models represent real world situations by using constructs that aid for understanding problem and solution space. This enables a clear understanding and effects exploration of design decisions.
- Methods define processes of solutions. A method can be a formal algorithm that explicitly defines the search process, an informal description of ‘best practice’ approaches, or something between provided in a semi-formal format.
- Instantiations show how to implement methods, models or constructs in a working system and show how a concrete artifact is suitable for its intended purpose.

Although, instantiations should be close to real world, [Hevner et al. 2004] distinguish design science from system development. System development is restricted to apply existing knowledge to a problem. Instead, design science addresses un-
solved problem and seeks in finding new solutions for solving problems more efficiently.

In order to testify design science research [Hevner et al. 2004] introduces seven guidelines.

<table>
<thead>
<tr>
<th>Guideline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Design as an Artifact</td>
<td>Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.</td>
</tr>
<tr>
<td>2. Problem Relevance</td>
<td>The objective of design science research is to develop technology-based solutions to important and relevant business problems.</td>
</tr>
<tr>
<td>3. Design Evaluation</td>
<td>The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.</td>
</tr>
<tr>
<td>4. Research Contribution</td>
<td>Effective design-science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.</td>
</tr>
<tr>
<td>5. Research Rigor</td>
<td>Design science research relies upon the application of rigorous methods in both the construction and evaluation of the design artifact.</td>
</tr>
<tr>
<td>6. Design as a Search Process</td>
<td>The search for an effective artifact requires utilizing available means to reach desired ends while satisfying laws in the problem environment.</td>
</tr>
<tr>
<td>7. Communication of Research</td>
<td>Design science research must be presented effectively both to technology-oriented as well as management-oriented audiences.</td>
</tr>
</tbody>
</table>

Table 1.1 Design-Science Research Guidelines [Hevner et al. 2004]

Researchers should acquire these guidelines when building and applying artifacts according to design-science research. However, [Hevner et al. 2004] “advise against mandatory or rote use of the guidelines”. Instead, researchers “must use their creative skills and judgment to determine when, where, and how to apply each of the guidelines in a specific research project” [Hevner et al. 2004].

Within this thesis these guidelines are addressed as follows:

**Guideline 1: Design as an Artifact**

The thesis results are a set of artifacts, i.e. constructs, models, methods and instantiation. Among others, business and electronic services, which are constructs for problem and solution communication, are introduced in chapter 2. Further, a model that represent real-world situations is addressed by the service concept meta-model (SCMM) in chapter 3. Methods and techniques are provided in chapter 4, which addresses a service modeling method, in chapter 5, which examines service matching algorithms, in chapter 6, which introduces visualization techniques for matching results, and in chapter 7, which examines tool support for the overall approach. Last not least, instantiations of constructs, models and methods are addressed by the validation chapter (i.e. chapter 8), which presents validation through instantiation and prove of applicability of the artifacts mentioned above.
Guideline 2: Problem Relevance

The selecting of electronic services is addressed by service-oriented paradigm using the term service discovery [Papazoglou/Georgakopoulos 2003]. The basic model of service-orientation dedicates an entity called service registry (also known as service broker) for that purpose. Basic mechanisms, such as UDDI [Clement et al. 2004], support the search for electronic services using keywords. However, it is generally believed that keyword based discovery can be improved significantly by introducing more powerful matching approaches [Papazoglou/Heuvel 2007]. Promising approaches come from Semantic Web Service area (i.e. WSMO [Roman et al. 2006b], OWL-S [Martin et al. 2004]). However, existing Semantic Web Service discovery approaches lack in practicability (cf. chapter 9.2). The research topic of preselecting electronic services by given business services is relevant, since it addresses an unsolved research field within the service-oriented research area (cf. [Papazoglou/Heuvel 2007]).

Guideline 3: Design Evaluation

The presented artifacts are demonstrated and validated by use case scenarios and a prototypical software tool that supports application of the overall approach (cf. chapter 8). Use case scenarios are developed within InterLogGrid together with major industrial partners (cf. [InterLogGrid 2011]). However, the concepts and mechanisms underlying this thesis are topics of ongoing research, which have not been used in large scale so far. So, validating the approach cannot be based on existing data, and, thus, has to be rather explorative and based on use cases, which do not exist in a productive environment yet.

Guideline 4: Research Contribution

The approach of preselection of electronic services by given business services fits in to the gap of keyword based service discovery and semantic Web service discovery. It is more precise than keyword based discovery. Semantic Web Services can be more precisely, but lack in practicability (cf. chapter 2.7). The thesis presents an approach of measuring semantic heterogeneity between business services and electronic using ontology matching algorithms. Matching algorithms result in matching values that can be aggregated, analyzed and visualized, and, thus, facilitate decision making during the evaluation of electronic services for a specific business service. And, hereby the thesis contributes a novel approach in service discovery. (Thesis contribution is discussed in detail in chapter 9)

Guideline 5: Research Rigor

The overall approach presented in this thesis is based on well-established concepts and mechanisms. It uses standards and best practices as far as possible and extends them in an appropriate way. For instance, the service concept meta-model is derived from existing well-known service models (cf. chapter 3). Matching algorithms are taken from ontology matching (cf. chapter 5). And lastly, just to name a few, well-known visualization techniques are applied for visualizing matching
result (cf. chapter 6). Thus, only rigorous models, methods and techniques are used in the construction and evaluation of the design artifacts.

Guideline 6. Design as a Search Process

The overall approach can be divided into several parts. Each of these parts had been developed in iteration with experimental applications and subsequent improvements. For instance, the service concept meta-model was derived from existing service models. Each existing service model had been examined several times resulting in a meta-model, which is viable for business and electronic service description. Another example is the evaluation of matching algorithms. Various matching algorithms from ontology matching area had been evaluated in detail, in order to derive a proper matching algorithm combination. So, the proposed solution is a result of a search process that had been taken place in different areas.

Guideline 7. Communication of Research

All artifacts that result from this thesis were communicated in different publications. The first glance of the thesis idea had been presented in [Kluge et al. 2008] and [Kluge et al. 2009]. The overall approach had been published in [Kluge et al. 2010] and [Kluge 2010]. Parts of the approach, such as visualization, had been presented in [Klinkmüller et al. 2010]. Further, the approach was part of several milestone reports of InterLogGrid and Logistics Service Bus research project, such as [Kluge 2009, Augenstein et al. 2010a, Augenstein et al. 2010b]. Audience of these publications were researchers and practitioners with technology-oriented and management-oriented background.

1.3 Structure of the thesis

The structure of the thesis follows the information systems (IS) research framework, which "aids in understanding, executing, and evaluating IS research" [Hevner/March 2003]. The IS research framework is depicted in Figure 1.1. The framework contains three vertical areas: environment, knowledge base, and IS research itself. The environment (cf. left hand side of Figure 1.1) comprises people, organizations and technology. This area describes the environment where business needs are derived from. Research based on business needs brings relevance to IS research (cf. subchapter 1.2 Guideline 2). The knowledge base (cf. right hand side of Figure 1.1) provides raw material for IS research. IS research results in artifacts that should be based on well-established theories, frameworks, instruments, constructs, models, methods etc. (cf. subchapter 1.2 Guideline 5). Only well-established artifacts are "applicable knowledge". IS research itself seeks in developing and building artifacts that fulfill business needs and is based on rigor knowledge. Further, artifacts should be justified and evaluated by case studies, simulations, experimental, analytical methods etc. Evaluation may lead to well established artifacts, which are applicable in business environment and can be
This thesis starts with a background section in chapter 2, which introduces the research area. Afterwards motivation and objectives are presented. This includes environment description from which “business needs” are derived from. Further, basic artifacts are introduced that work as “applicable knowledge” for following research. Based on that, the basic constituents and the overall approach are presented in subchapter 2.5. Basic constituent introduce thesis’ artifacts the first time. Hence, chapter 2 touches each area, i.e. environment, knowledge base, and artifacts under development, of the IS research framework. Each artifact mentioned in 2.5 is developed and build in chapters 3 to 7. Each of these chapters start with examining existing knowledge (i.e. knowledge base) and deriving proper artifacts out of it. Thus, applicable knowledge is acquired before developing artifacts. After chapter 7, chapter 8 validates and justifies all results in conjunction.

This thesis can be separated into four parts as depicted in Figure 1.2. Each part consists of one or more chapters with two or more subchapters. The first part ad-
addresses introduction and motivation and comprises chapter 1 and 2. The second part addresses models, methods and techniques needed for the overall approach and spans from chapter 3 to chapter 6. The third part is dedicated to application and validation, which includes chapter 7 and 8. Lastly, there is one chapter that draws a conclusion and gives an outlook for future research. Each of these chapters is introduced briefly in the following.

**Introduction and motivation**

- Chapter 1 provides the thesis’ general introduction including an introduction of the research area, research topic, research methodology, and structure of the thesis.
- Chapter 2 provides introduction, background, motivation, requirements, and the solution pathway. Thus, it spans background knowledge necessary for understanding the research area and topic, objectives that motivate this thesis, it addresses general conditions, prerequisites and assumptions under which the thesis will be examined, and it provides general requirements from which the overall conceptual approach is derived from. The overall approach provides solution pathway of the thesis. Each of the following chapters is dedicated to one particular part of the conceptual approach.

**Models, methods and techniques for the overall approach**

- As the initial part of the conceptual approach, chapter 3 is dedicated to the representation of business and electronic service. In order to find a proper representation form, distinct criteria are defined, first. Next, existing service representation approaches are examined, compared, and evaluated in detail. As a result there is a meta-model for service representation called service concept meta-model (SCMM). The SCMM will be used for further considerations in the following chapters.
- After the definition of the SCMM, chapter 4 addresses the question "How to model according to that SCMM?" Methods, techniques, and tools are need for that purpose. After a detailed examination of existing approaches, chapter 4 provides a method for service modeling according to the SCMM. The validity of this method is proven by a case study. The case study is supported by a tool presented in chapter 4 as well.
- As one of the main parts of the conceptual approach chapter 5 examines matching between business services and electronic services. Existing algorithms from ontology matching area are examined in detail for that purpose. The applicability of each of the algorithms is proven by examples from the logistics domain. At the end, this chapter provides a proper matching algorithm, which is applicable for SCMM based service models.
- Chapter 6 addresses visualization of matching results. First, there is a general introduction to visualization of data and information. Next, existing visualization methods are examined and evaluated by distinct requirements. Last, the application of visualization methods is shown and analysis suggestions are given.
Application and validation

- In order to apply all parts of the conceptual approach in combination tool support is needed. A tool helps modeling according to the SCMM, applies matching algorithms and provides visualization for analyzing matching results properly. Such a tool is introduced and presented in detail in chapter 7.

- Chapter 8 provides validation as required by [Hevner et al. 2004]. Here validation is rather explorative. Thus, based on three use cases and several business and electronic service descriptions the overall approach is applied using the tool mentioned previously. As a result this chapter will prove whether the approach meet its expectations.

Conclusion and outlook

- The final chapter, i.e. chapter 9, provides conclusion and outlook. The conclusion comprises a summary, explanation of main contribution and a critical discussion about thesis’ results. Further, open topics and future work are presented.
2 Background, motivation and solution pathway

This chapter provides background knowledge at first. Second, motivation and objectives of the thesis are introduced. Further, assumptions and requirements are derived from these objectives. Afterwards, the thesis’ conceptual approach is presented. The conceptual approach provides a pathway for all following chapters. This chapter ends with a summary.

2.1 Background

This thesis is written within the Logistics Service Engineering and Management (LSEM) research group. LSEM seeks in leveraging the engineering and the management of logistics services by applying service oriented paradigm for logistics. The background chapter introduces service orientation first, including definitions, characterizations and delimitations. Second, there is a brief introduction of the logistics domain. At the end, this subchapter introduces the research framework of LSEM and relates the thesis to this framework.

2.1.1 Service orientation

Service orientation is a design paradigm that constitutes services as building blocks for business and technological functions. Service orientation has been extensively described and discussed in literature by researchers and practitioners, i.e. [Alonso et al. 2004, Krafzig et al. 2004, Marks/Bell 2006, Erl 2007, Papazoglou/Heuvel 2007, Paruchuri 2007]. As such, the design paradigm service orientation provides design principles around the service concept, a distinct architectural model [Papazoglou/Heuvel 2007], as well as various other models, methods, technologies, frameworks, and tools. Main constituents of service orientation are explained below.

Service

Services are central artifacts of service orientation. In literature, there are several service definitions. Some of them are presented below:

- „Services exist as physically independent software programs with distinct design characteristics that support the attainment of the strategic goals associated with service-oriented computing. Each service is assigned its own distinct functional context and is comprised of a set of capabilities related to this context. Those capabilities suitable for invocation by external consumer programs are commonly expressed via a published service contract (much like a traditional API).” [Erl 2007]
- „A service is an exposed, self-contained, and platform-independent piece of functionality with well-defined interface that can be dynamically located and invoked.” ([Bonati et al. 2006], p. 120)
• “A service is a mechanism to enable access to one or more capabilities, where
the access is provided using a prescribed interface and is exercised consistent
with constraints and policies as specified by the service description.” ([OASIS
Open 2006], p. 12)
• “A service is a logical representation of a repeatable business activity that has a
specified outcome, such as ‘check customer credit’, ‘provide weather data’, or
‘consolidate drilling reports’. It is self-contained, may be composed of other
services, and is a ‘black box’ to its consumers.” ([The Open Group - SOA
Working Group 2007, The Open Group SOA Working Group 2007], p. 6)
• “Services are self-describing, open components that support rapid, low-cost
composition of distributed applications. Services are offered by service provid-
er-organizations that procure the service implementations, supply their service
descriptions, and provide related technical and business support.” ([Papazoglou/Georgakopoulos 2003], p. 26)
• “Services are the act and result of making available some resources belonging to
an entity called service provider, to an entity called the consumer, who, as a re-
sult, acquires a capability. […] Consumer […] experience the actual service

There is a variety of service definitions and, so far, there is no all-encompassing
widely accepted one. The definitions range from considering a service as a “soft-
ware program” to “a service as the result of making available some resources for a
customer”. This is due to the fact that service orientation is a paradigm that ad-
dresses business and technological aspects at same time. Beside the definition,
there are service principles. Service principles characterize the nature of a service
and are, in contrast to the definitions, widely accepted. According to [Marks/Bell
2006] and [Erl 2007] these principles are:

(1) Service contract: A service must have a service contract that expresses its
purpose and its capabilities. Further, the contract separates the service functional-
ity from its specific implementation. So, it presents functionality to the outside
world, e.g. to a consumer, while hiding detail about implementation.

(2) Service discoverability: Service should be discoverable, which means that a
service should be well designed and that service contracts should be published and
visible to the intent audiences. This implies that service contracts are available and
searchable in a service repository.

(3) Service loose coupling: The coupling principle refers to the relationship be-
tween the service contract and its implementation. Loose coupling means that the
service implementation should be independent from the service contract. In that
way, a specific implementation can be modified, evolved or even replaced without
changing the contract and, more importantly, without affecting the way a consum-
er interacts with the service.

(4) Service statelessness: Since service state information can compromise avail-
bility of a service and undermine the scalability, services should be designed state-
less ideally. However, since most of the services need state information for operation, one should design services as less statefull as possible.

(5) Service reusability: Reusable means that a service must not be designed for a single purpose of one single consumer only. Instead, a service must be designed in order to serve several purposes and use cases, and beyond, even use case one might not have thought about during design time. Since service development can be time-consuming and costly, reusability is needed in order to ensure economies of scale.

(6) Service autonomy: In order ensure that a service carries out its capabilities reliable, consistent and predictable a service should have a significant control over its resource and environment. Service autonomy is important when designing reusable services that are frequently shared.

(7) Service composability: Service should be composable, i.e. services should be designed in a way that they can be incorporated into other (composite) services. Further, service should be able to be assembled into orchestrated process flows.

(8) Service abstraction: Service abstraction emphasizes the need to hide as much information about underlying details as possible. This ties into many aspects of service-orientation, e.g. the relationship of service contract and service implementation, and the relationship between a composite service and its underlying services.

(9) Service interoperability: Basically, this principle refers to the interoperation of services between each other and their environment through standardized interfaces and communication protocols enforced by a body of policies. Service interoperability is fundamental to each of the principles mentioned above.

Since there is no common definition of services, these principles build the foundation for understanding services in course of service orientation.

Service-oriented architecture

Service orientation is driven by a distinct architectural model that defines constituents and roles. Similar to the service definitions, there are several definitions for service-oriented architecture (SOA):

- “Service-oriented architectures is an architectural style for a community of providers and consumers of services” [Object Management Group (OMG) 2006b]
- “Service-oriented architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.” [OASIS Open 2006]
- “Service-oriented architectures define the overall structure that is used to support service-oriented programming for service-oriented applications in the service-oriented enterprise” [Khoshafian 2006]
- “Service-oriented architectures (SOA) is an […] approach that addresses the requirements of loosely coupled, standards-based, and protocol independent distributed computing.” [Papazoglou/Heuvel 2007]
“Service-oriented architecture represents a distinct form of technology architecture designed in support of service-oriented solution logic, which is comprised of services and service compositions shaped by and designed in accordance with service-orientation.” [Erl 2007]

“SOA is a conceptual business architecture where business functionality, or application logic, is made available to SOA users, or consumers, as shared, reusable services on an IT network” [Marks/Bell 2006]

SOA definitions range from a business and management centric interpretation (cf. first three definitions) to a software and computing centric view (cf. last three definitions). The reason is that service orientation is considered as a management and computing paradigm at same time. Depending on the authors’ scope and the audience, business or computing aspects are more emphasized in literature. Most SOA literature provide both interpretations within the introduction part, but focus management or software aspects in detail later on.

Even though SOA literature may vary, most of the authors agree with the SOA role model. As already suggested by some of the definitions above, there is a service provider, a service consumer and a service broker. These roles and the relationship between them are illustrated in Figure 2.1.

![Figure 2.1 Roles and interactions in SOA (cf. [Khoshafian 2006, Papazoglou/Heuvel 2007] and others)](image)

As the name suggests, the service provider provides the service. The provider offers a service contract that can be requested by a service consumer. Before providing a service, the service provider has to publish the service via a service broker. When a service consumer requests a service, the service provider has to ensure that the service will be carried out. The fulfillment can be done by the service provider itself. However, the service provider can work as an aggregator providing a composite service, which might be fulfilled by other service providers. How the service is fulfilled, i.e. the service implementation, is transparent to the service consumer. The service consumer, also known as service requester [World Wide Web Consortium (W3C) 2004c] or service customer, searches for a service by requesting the service broker. After having found a proper service the consumer has to bind the service. Binding means integrating the service into the requesters’ environment. It may comprise negotiation about specific aspects of the service.
contract as well. Afterwards, the service can be used and will be carried out by the service provider. The service broker is also known as service registry [Khoshaﬁan 2006] or service repository. A service broker acts as registrar, which enables service providers to register provided services in a registry and allows service consumer to search and discover registered services. Service brokers ensure discoverability and reusability in service orientation.

Beside the agreement about the SOA role model and that SOA comprises business and computing aspects at same time, it is widely accepted that SOA is individual for each company. Further, since SOA cannot be deﬁned by hard facts, authors tend to deﬁned SOA by statements about “what it is not”. For instances, [Marks/Bell 2006] states “SOA is not a product. SOA is not a solution. SOA is not a technology.” and [McKendrick 2008] provides ten rules that testify if it is not a SOA. Particularly the distinction between SOA business aspects and SOA technological aspects involves people from different domains. Thus, authors tend to use the term service-oriented computing (SOC) when talking about technological aspects and the term service-oriented enterprise (SOE) when addressing business and organizational aspect.

**Service-oriented computing**

Service-oriented computing (SOC) is a “computing paradigm that utilizes services as the constructs to support the development of rapid low-cost composition of distributed applications. With this, SOC represents the combination of a wide spectrum of technologies including distributed systems, software engineering, information systems, computer languages, Web-based computing, and XML technologies rather than being a new technology.” [Papazoglou 2007]. SOC means designing, implementing, and running a computing platform based on the SOA design paradigm. SOC does not dictate the technology for implementation. However, several technological concepts had been established and are predominate within SOC, such as Enterprise Service Bus (ESB) [Papazoglou/Heuvel 2007], Web Services [World Wide Web Consortium (W3C) 2004c] and the WS-* speciﬁcations [Weerawarana et al. 2005].

**Service-oriented enterprise**

[Khoshaﬁan 2006] deﬁnes service-oriented enterprise as “… standards-based integration paradigm” that provides a “new way of building enterprises that are extended, virtual, real-time, and resilient” and a “new way of thinking about applications, partnerships, and outsourcing.” SOE is termed as a thinking culture where each participant sees himself as a service provider and service consumer at same time. According to [Khoshaﬁan 2006], this is different from the past in two ways: First, since today’s companies serve the needs of several parties (such as trading partners, employees, shareholders, governments, and communities), service orientation ﬁts better for today’s concerns. Service-oriented thinking helps organizations to focus on their core competences. Second, a huge change emerged through service orientation as the computing paradigm, which enhances the way of automating processes. Service-oriented computing works as an enabler for
some businesses (cf. [Cherbakov et al. 2005, Khoshafian 2006, Henneberger 2008])

Further details about services, SOA, SOC and SOE can be taken from literature as suggested by the references above. Next, the thesis application domain is introduced.

2.1.2 Application domain logistics

For use cases presentation and validation purposes this thesis is dedicated to the logistics domain. Logistics is defined as the market-oriented, integrated planning, design, processing, and controlling of the flow of goods and information between two companies, which are in a supplier-customer relationship. The material flow, also known as physical logistics, comprises transportation, transshipment, and warehousing as well as several other activities, e.g. assembling, refinement, wrapping etc. The flow of information, also known as dispositive logistics, comprises planning, management, and controlling activities, such as scheduling, quantity planning, capacity planning, and order management (cf. [Christopher 2004, Rushton et al. 2010]).

Subjects of consideration within the logistics domain are logistics services. A logistics service is a service of the logistics domain with arbitrary complexity. A logistics service is offered by logistics service provider and consumed by a logistics service consumer. Complexity refers to influencing parameters, which make services more or less complex. There are several parameters that may cause complexity, such as customer demands, market determinations (e.g. statutory regulations) or cargo specific requirements (e.g. transportation of chemical products or food). Logistics service providers are companies that contribute to a physical or dispositive logistics service. This includes traditional logistics companies, i.e. second-, third- and fourth party logistics provider, as well as companies that provide services in terms of information, such as logistics software vendors. Analogously, logistics service consumers are companies that request physical or dispositive logistics services. These are typically consignees or consigners, such as manufacturing companies and shippers. However, even logistics companies such as contract logistics provider (i.e. third-party logistics provider) or logistics service aggregators may act as a logistics service consumer as well. For instance, the business model of a fourth-party logistics (4PL) provider relies on the aggregation of logistics service. That means, a 4PL receives physical and dispositive logistics service from other providers, aggregates these service in a meaningful way and offers them as a composite service at the market. During service fulfillment 4PLs act as a logistics service provider for the end customer, but as a logistics service consumer to the supplying logistics service providers.

Service orientation within logistics domain is addressed by the logistics service engineering and management (LSEM) resource group. Research aims of LSEM are introduced in the following.
2.1.3 LSEM research framework

The logistics service engineering and management (LSEM) group was founded in summer 2006 at University of Leipzig within the research project Logistics Service Bus. The projects objective was building a service-oriented IT platform for logistics participants with a special focus on fourth party logistics provider (4PL). During development the research group formed a research framework and an agenda in order to structure research topics. The research framework lists relevant concepts, methods, and research areas and defines relationships between them. The framework provides an approach that fosters the engineering and management of logistics service, which comprises analyzing, planning, design, implementation, deployment and determination. This thesis is written within LSEM. So, this subchapter provides a brief overview about the LSEM research agenda and relates the thesis topic to LSEM.

2.1.3.1 Service-oriented logistics service

Subject of LSEM consideration are logistics services. Following service orientation, LSEM postulates that each logistics service can be presented by a service (or possibly a set of services) that is according to the service-oriented paradigm. In that way, logistics services are represented as an encapsulated distinct functionality of physical and dispositive logistics. The distinction between a logistics service and its service-oriented representation is that logistics services are not structured and formal described. The service-oriented representation, instead, fulfills this characteristic. It comprises, for instance, a well-defined encapsulated functionality, which is structured and formally described, and includes a well defined interface.

2.1.3.2 Business services and electronic services

As mentioned in section 2.1.1, service orientation addresses both - business and technological aspects. Thus, when talking about service it is common to distinguish business and technical services (et al. [Marks/Bell 2006, Karakostas/Zorgios 2008]). Within LSEM research framework the understanding for business and technical services is adopted and used analogously.

A business services (BS) represents functionality according to the service-oriented paradigm, which supports the execution of business processes within an organization directly. BS describe business capabilities that are offered by a service provider and can be consumed by a service consumer. BS are core business functionalities and, thus, may work as fundamental building blocks of an enterprise architecture (et al. [Zachman 2001, Bredemeyer et al. 2002]). All service-oriented principles apply to BS. So, BS are composable, i.e. several BS can be brought into a process flow (e.g. by using BPMN [Miers/White 2008]) and form a new compo-

---

1 http://www.lsem.de
2 http://www.lsb-plattform.de
site BS. Examples for BS are invoicing, transport planning, order management, warehousing or loading of goods.

Technical services are termed as electronic services (ES) within LSEM. This is due to the fact that technical service may comprise software and hardware as well. The term 'electronic' emphasizes the software and neglects all hardware aspects. Thus, ES represent the software infrastructure that supports the fulfillment of a BS. Typical examples of ES are application software systems, database systems, customer relationship software, enterprise resource planning software, transport management software etc.

Although BS and ES are defined by their scope on business or software aspects, a clear distinction is hard to determine in each case. This is due to the fact that software certainly may contain business aspects and may use business terminology as well. So, in order to make a distinction between both services, LSEM defines six criteria. Below, each criterion is instantiated for BS and ES and explained by examples from the logistics domain (cf. Table 2.1).

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Business service</th>
<th>Electronic service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Providers</td>
<td>logistics service providers, in particular second-, third- and fourth-party logistics</td>
<td>logistics service provider, in particular, software vendors and -companies</td>
</tr>
<tr>
<td>Consumers</td>
<td>consigners, consignees, or third- and fourth party logistics (service aggregators)</td>
<td>almost all logistics companies; in particular second-, third- and fourth-party logistics</td>
</tr>
<tr>
<td>Activity type</td>
<td>physical activities (e.g. transportation, freight loading) and dispositive activities (e.g. ordering, planning, invoicing)</td>
<td>dispositive activities (e.g. ordering, planning, invoicing)</td>
</tr>
<tr>
<td>Implementation</td>
<td>not software, cannot be dedicated to a distinct software system</td>
<td>representation of software, can be dedicated to a distinct software system</td>
</tr>
<tr>
<td>Computational</td>
<td>less complex</td>
<td>more complex</td>
</tr>
<tr>
<td>Terminology</td>
<td>business terms, usually no technical terms</td>
<td>business terms and technical</td>
</tr>
</tbody>
</table>

*Table 2.1 Criteria for business and electronic services*

Business services are provided by logistics service providers, in particular shippers, freight forwarder, warehousing companies, and transshipment providers (i.e. second-, third- and fourth-party logistics). According to the definition of a logistics service, an electronic services provider is a logistics service provider. Howev-
er, providers of electronic services are typically software vendors, i.e. companies that develop, host, and license software systems.

Consumers of ES are almost all logistics companies, since all of them need (more or less) software that supports their business. In contrast, business services consumers are usually consigners (e.g. manufacturing companies), consignees (trading companies), or third- and fourth party logistics, i.e. service aggregators.

There are physical and dispositive logistics services. A BS comprises physical activities, such as transportation, freight loading, etc., and dispositive activities, such as planning, invoicing, and management. Thus, business services can be physical or dispositive. Instead, ES address dispositive activities only, since ES deal with information. However, an ES may possibly track physical activities (e.g. tracking of transport state) or even trigger a physical service (e.g. start loading). However, ES are dispositive in their nature.

Another criterion is the implementation of BS and ES. As encapsulated software functionality, ES are fulfilled by software. Thus, ES can be dedicated to one or many distinct software systems. In contrast, BS are usually carried out by humans, machines, vehicles etc. A business service cannot be dedicated to a distinct software system. Sure, a business service can be supported by software (i.e. an ES), but it is not software. So, the distinction between BS and ES is that the latter is software and the former is not software. It is also a business service if it is not clear if it is software or not, i.e. if one cannot dedicate the service to a distinct software system.

Due to their implementation BS and ES are different in their computational complexity. Since ES are implemented by software they are more likely to be computational complex then BS. BS are possibly fulfilled by humans and, thus, less complex regarding computation.

Last not least, business and electronic service representations contain different types of terminologies. Typically, business service descriptions use terms that are related to the business domain, such as invoice, driver, freight, etc. Within BS descriptions there are usually no technical terms. In contrast, ES contain technical terms, e.g. messaging queuing, runtime, authentication etc., but business terms as well. This is due to the fact that business user need to communicate with a software system. Business users would not be able to use software if the software contains terminology they are not familiar with. For instance, the term ‘invoice’ is a typical business term that is used for a BS description and will certainly be found in each enterprise resource planning (ERP) software product as well.

2.1.3.3 LSEM service lifecycle

Services underlie a temporal dimension, i.e. they are created, provided, fulfilled, and terminated later on. Thus, there is a service lifecycle. LSEM defines a service lifecycle that comprises four mayor phases: (1) Service Engineering, i.e. systematic development of services, (2) Service Deployment, i.e. systematic setup of services, (3) Service Operation, i.e. systematic provision and fulfillment of a service,
Service Termination, i.e. systematic termination of a service. An overview of all phases including sub-phases is shown in Figure 2.2.

The phases above can be applied to BS and ES. The lifecycle phases for BS are Business Service Engineering, Business Service Development, Business Service Operation and Business Service Termination. These phases are defined below.

**Business Service Engineering (BSE)** is defined as the engineering-based (i.e. under consideration of applied models, methods, techniques), and goal-oriented development of (structured service-oriented) BS from existing (unstructured) logistics services. Development is divided into analysis and identification, design, implementation, and test. BSE results in a set of structured BS for logistics that are represented as a Business Service Map and can be used for the following phase.

**Business Service Deployment (BSD)** is defined as the engineering-based (i.e. under consideration of applied models, methods, techniques), and goal-oriented setup and provisioning of BS for a certain use case. This phase is subdivided into analysis, design, implementation, and test and simulation. During these sub-phases there are several activities, such as composition of BS to business processes, BS configuration (e.g. definition of service-level parameters), simulation of BS in order to predict their behavior, selection of adequate ES that support the BS, and binding BS for concrete BS instantiations. As a result the BSD phase delivers a concrete BS instance that fulfills a certain purpose (e.g. a logistics service contract). This BS instance has been composed from other service, is configured, tested and simulated, and bound to certain providers as well as to ES that support the fulfillment of BS.

**Business Service Operation (BSO)** is defined as the systematic usually recurring fulfillment of a BS according to the settings and characteristics that had been defined in BSD. Moreover, BSO comprises monitoring and reporting of running BS instances (e.g. by a target/actual comparison) as well as maintenance and customizing. BSO provides activities for accurate fulfillment and controlled BS operation.
Business Service Termination (BST) is defined as the systematic termination of the operation of a BS. This may be necessary at the end of a service contract, for instance, and requires activities such as stopping operation, release of bindings, replacement of functionality, retirement, and BS archiving. All in all, BST provides activities for terminating one or a set of BS.

Analogous to BS, the ES lifecycle phases are Electronic Service Engineering, Electronic Service Development, Electronic Service Operation, and Electronic Service Termination. These phases are defined in the following.

Electronic Service Engineering (ESE) is defined as the engineering-based (i.e. under consideration of applied models, methods, techniques), and goal-oriented development of (structured service-oriented) ES from existing (unstructured) logistics services, such as proprietary software systems or legacy software systems. Development is divided into analysis and identification, design, implementation, and test. ESE results in a set of structured ES for logistics that are represented as an Electronic Service Map and can be used for the following phase.

Electronic Service Development (ESD) is defined as the engineering-based (i.e. under consideration of applied models, methods, techniques), and goal-oriented setup and provisioning of BS for a certain use case. ESD is subdivided into analysis, design, implementation, test, and simulation. During the sub-phases there are several activities, such as composition of ES in order to support BS as a composite ES, ES configuration (e.g. definition of service-level parameters), simulation of ES in order to predict their behavior, ES binding, and ES instantiation. As a result the ESD phase delivers a concrete ES instance that fulfills a certain purpose.

Electronic Service Operation (ESO) is defined as the systematic usually recurring operation of ES according to the settings and characteristics that had been defined in BSD. Moreover, ESO comprises monitoring and reporting of BS instances (e.g. by a target/actual comparison) as well as maintenance and customizing (if necessary). ESO provides activities for accurate fulfillment and controlled ES operation.

Electronic Service Termination (EST) is defined as the systematic termination of ES operation. Service termination requires activities such as stop operation, release bindings, replacement of functionality, retirement, and ES archiving. All in all, EST provides activities for terminating one or a set of ES.

2.1.3.4 LSEM research agenda and the thesis topic

The research agenda is conducted to the LSEM service lifecycle. Next, research topics that touch the thesis are explained. Further, the relation between the thesis’ topic and the LSEM research agenda is explained as well.

BS and ES representation from unstructured logistics services

Since BS and ES are representations of unstructured logistics services, a systematic approach is needed that transforms existing logistics services into a service-oriented representation. Of course, this approach has to consider the BS and ES specifics. Further, it comprises definition of a proper model (or a set of models), a
method that guides through, and techniques to support. One of the challenges is to clearly define boundaries and interfaces of logistics services. Further, one has to find a proper representation granularity-level\(^3\). As a result of this research topic, there is an approach that provides models, methods, guideline, and best-practices for the transformation of unstructured logistics services into BS/ES.

*Structuring of services by an integrated modeling architecture*

BS and ES provide formalized representations of logistics services. However, service representations may serve several purposes. For instance, one service representation is used for monitoring, another one for simulation purposes, and a third one serves service proposition to the end customer. Thus, there are several representations, i.e. several service models, which provide a specific view each, but refer to the same BS or ES. So, an integrated modeling framework and architecture is needed that manages connections as well as relations between those models. One of the challenges is to structure models in a way that they are manageable. Another challenge is to ensure consistency even after editing models. As a result of this research topic, there is an approach that provides an integrated modeling architecture including tools that support modeling.

*Service repository for service maps*

After the transformation of unstructured logistics services from existing businesses and software systems, there are many BS and ES available. In order to maintain overview about the service one has to structure and visualize them. For that purpose and in analogy to a real map, it is feasible to create a service map that supports navigation through all services. Further, all services (including their service models) should be accessible and discoverable. Thus, a service repository is needed that serves as a database for BS and ES. Setting up a service repository and visualizing a service map is rather a research topic. However, both is required within LSEM and has to be considered.

*LSEM tool chain*

Another research topic, which is related to the whole LSEM research area, is tool support. Since there are many tools and techniques necessary, e.g. for service simulation, service monitoring, service modeling, service visualization, service transformation etc., one has to ensure that all tools are consistent and compatible to each other. For instance, tools should be able to exchange data in order to be able to create a service model with a tool, refine it with another one and use the model for simulation later on. The platform that ensures compatibility of the tool chain within LSEM is the eclipse modeling framework (EMF)\(^4\). Thus, every tool that will be developed has to be according to the EMF specification. Setting up a tool chain using EMF is rather a research topic.

---

\(^3\) This problem is discussed et al. in [Marks/Bell 2006] (p. 48f)

\(^4\) [http://www.eclipse.org/modeling/emf/](http://www.eclipse.org/modeling/emf/)
Identification of ES that support BS

ES support the fulfillment of BS. ES provide information, computation functionality and communication functionality that supports physical and planning activities (i.e. BS). However, each ES has different functionalities and may support a particular BS more or less. So, the question arises: Which ES (or which set of ES) support a particular BS best? In order to answer this, one has to know the particular BS and all available ES very well. Further, one has to define evaluation criteria, which serve as indicators that identify ES candidates. Afterwards, one has to evaluate in detail in order to find that particular ES (or set of ES) that supports the BS best. A well-founded decision is cost- and time-consuming. This is even worse, assuming that there are many BS and many ES available. So, the research topic around identification of ES that supports particular BS is about the definition of evaluation criteria and about easing the decision process by using adequate models, methods and tool.

2.1.3.5 LSEM and the thesis

This thesis is embedded in the Business Service Deployment (BSD) phase within the LSEM service lifecycle. BSD happens after Business Service Engineering (BSE) and before Business Service Operation (BSO). Before BSD can be fulfilled, one has to transform unstructured logistics-services into BS and make them available in a service repository. This is fulfilled by BSE. Further, before BSO can be carried out, BS have to be deployed for a certain use case. This comprises BS composition, BS configuration, binding of BS instances, and the selection of ES that support the operation of the BS. In particular, the thesis is dedicated to the ES selection activity of BSD. As a precondition for this activity, the phase Electronic Service Engineering (ESE) has to be carried out, i.e. the transformation of unstructured proprietary software systems, legacy software systems etc. into service-oriented ES. ES descriptions ensure adequate description of software systems within LSEM.

Considering the LSEM research agenda the thesis topic relates to the identification of ES that support BS. However, it is related to other LSEM research topics as well. As mentioned before, one needs proper BS and ES representations for proper evaluation. Thus, research in this area works as a precondition for the thesis. Further, since there are several views on BS and ES and several BS and ES models as well, an integrated modeling architecture supports gaining proper information from BS and ES for evaluation purposes. Moreover, having services represented as a service map and having their descriptions stored in service repository, eases the accessible of BS and ES and thus fosters evaluation too. Considering tools, the LSEM tool chain provides a foundation for all tools that will be applied within the thesis topic.

After the introduction of LSEM and the thesis' relation to LSEM, the problem area and the motivation are explained in the following section.
2.2 Problem description and motivation

After the background section, where service orientation, logistics and the research agenda of LSEM the thesis is related to had been introduced, this section provides problem description and motivation of the thesis.

The thesis is embedded in the LSEM research framework and surrounded by other LSEM research topics. However the problem area of ES evaluation for BS purposes is much broader. Considering BS as representations of real-world services (such as transportation, order management, etc.) and ES as representations of software (such as customer relationship management, enterprise resource planning software, etc.) the topic is related to the alignment of business and IT. Business-IT-Alignment “… deals with the alignment of the IT strategy of an organization with its business strategy. […] It is the desired state in which a business organization is able to use IT efficiently and effectively to achieve its business objectives.” [Viaene 2005] Business-IT-Alignment concerns researchers and practitioners since IT was introduced to organizations and, thus, had been discussed in literature very often (cf. [Krafzig et al. 2004, Polikoff et al. 2005, Gronau/Rohloff 2007]). Considering the implementation of proper software products, i.e. software products that are aligned to companies business, a particular company has decided if an individual software product should be built or if an existing software product should be bought. For decision making, one has to know the requirements the software has to fulfill, has to estimate possible development costs, and evaluate existing software products based on distinct criteria. Here, especially requirements analysis and the selection process are time-consuming and costly. Considering evaluation of existing software, one has to scan the market, has to get familiar with each of the products, ask product vendors or consultants about the product features, and should do some test installations.

Basically, today’s software evaluation has three main challenges: hidden knowledge, complexity, and other motives. Hidden knowledge is due to the fact that there are so many software products on the market that it is impossible to know all details of each of them. On the other hand, there is hidden knowledge about requirements. It is hard to get information about the requirements a company claims to a software product. Since knowledge about requirements and software features are from vital importance, i.e. it is impossible to make an objective decision without this knowledge, eliciting this knowledge is a key factor for software evaluation. However, even one may have sufficient knowledge about requirements and all relevant software products, the complexity of information is unmanageable. This is due to two facts. First, there are many requirements, many software products on the market and many details about each software product leading to a flood of information that becomes unmanageable. Second, there is a variety of

* It refers to a typical make or buy decision (cf. [Probert 1997])
* For example, the software product SAP Transport Management, which is one of many software products, is functionality explained in a book with 646 pages (cf. [Lauterbach et al. 2009]). This testifies the mentioned complexity.
possibilities how requirements and software product information are presented. Requirement are described in text from, with UML diagrams, or collected in a requirements tool (such as DOORS [IBM Corporation 2011]), usually. The functionality of software products is described by manuals, online documentations or books (such as [Lauterbach et al. 2009]). Anyway, most of the information is unstructured. Thus, the complexity of information makes detailed comparison of requirements and software product features impossible. Last not least, there are other motives that drive the decisions about software (cf. [Shah 2010]):

- **Personal motives:** The software selection process is influence by individual opinions and personal feelings. For instance, if someone used to work with a software product in the past, this person tends to favor for that product without knowing the alternatives in detail. Another example, if a person knows the vendor of the software product very well, he or she tend to prefer this product over the products from other vendors.

- **Organizational motives:** Since there are (usually) other software products in use already, organizations tend to select new software products from the same vendor or from the same software family. This is because, decision makers expect that employees are already familiar with a related software product and can get used to the new software easily. Further, it is expected that existing software interacts with a new software product much better if it is from the same vendor.

- **Strategic motives:** Software selection might be influences by strategic goals, i.e. affecting the company’s external image by implementing a particular IT solution. For instance, if a company implements an expensive enterprise software product the company may get a wealthy image. Further, if a company aims at implementing the newest technology, such as RFID (cf. [Finkenzeller 2010]), a company may get the image of a technology pioneer. Furthermore, by implementing the software product that is used by business partner, a company shows readiness for working with this partner.

Although these motives might be feasible in some cases, they are no objective evaluation criteria. Objective criteria for successful evaluation are the functional and non-functional features of a software product, which have to be aligned to the requirements.

Considering BS as representations of functional building blocks of business and ES as representations of functional building blocks of software products there is a chance to overcome the issues of today’s software evaluations, i.e. hidden information, complexity and other motives. By definition BS and ES are formal described by a service description, which comprise purpose and capabilities descriptions as well as functional and non-functional aspects. Thus, the problem of hidden information is solved by proper BS and ES representations. Further, services contracts are independent from their implementation and, thus, independent from a particular vendor. This makes it possible to fade-out other (vendor related) motives and to be focused on objective criteria, i.e. functional and non-functional aspects. Concerning complexity, the issue of unstructured or different structured information can be solved partly only. Service-orientation provides some guide-
lines and principles for that purpose (cf. section 2.1.1). However, there are various possibilities of service representation. Furthermore the complexity in terms of the flood of information is not solved either. Each of the BS and ES representations contains a lot of information. Although BS and ES are discoverable due to service-orientation, one has to define the criteria on which the discovery is based on. Considering selection of ES that supports a particular BS one has to discover each ES and evaluate it in detail, which is still time-consuming and complex. And it becomes even worse when considering ES selection for a whole company with many BS. For example, assuming that there are 50 BS within a company and 1000 ES available at the market there are 50,000 combination possibilities. The evaluation of all of these combinations cannot be accomplished to acceptable costs. A full evaluation is not possible.

The thesis is dedicate to the problem area of information complexity during ES evaluation by supporting the detailed evaluation through a preselection phase that considers a particular BS and reduces the large number of available ES to a significant smaller one. Figure 2.3 illustrates the relation between preselection and an overall evaluation process.

As shown in Figure 2.3 the evaluation process starts from a total amount of ES. Considering the requirements of a particular BS a functional based evaluation is carried out first. The evaluation based on functional criteria seeks in finding functional matches between BS and ES. This results in a smaller amount of ES candidates, which show functional compatibility with the BS. After functional evaluation, the evaluation of ES candidates based on their non-functional features leads to one ES that fits the BS concerns. This thesis is focused on functional based evaluation. Thesis' objectives are explained next.

---

7 The fact that many ES might be used by a BS or one ES is used by many BS is neglected here.
8 Actually, it is also possible that there is no ES or two or more ES after non-functional based evaluation. In the first case, one may have a look into another service repository (or into another service market) or has to develop a new ES from scratch. However, it could be that it makes no sense to support that particular BS by an ES in this case. In the latter case, i.e. non-functional based evaluation...
2.3 Objectives and basic idea

Objectives are based on the problem description stated above. The general objective of the thesis is to provide models, methods, and techniques for preselection of ES for a given BS. Preselection does neither mean full evaluation nor automatic selection. Instead, preselection is about reducing the large amount of ES to a significant smaller amount under consideration of a particular BS. Models, methods, and techniques work as decision support for ES evaluation.

Since manual evaluation of ES is complex, time-consuming and costly, the thesis aims at providing a semi-automated mechanism that works more efficient than manual evaluation. Semi-automation means that there are algorithms, which discover ES by distinct criteria automatically. However, since full-automation is error prone and leads to incorrect results, semi-automation is more feasible. Semi-automation leaves space for manual steps, analysis and interpretation.

Basic idea of preselection is measuring heterogeneity between BS and ES descriptions and conclude about the matching level between BS and ES from that. Considering service descriptions, there are three different types of heterogeneity: syntactic heterogeneity, terminological and semantic heterogeneity, and semiotic heterogeneity. Semantic heterogeneity is when service descriptions are based on different languages, different conceptual models or different formalisms. Terminological or semantic heterogeneity is when service descriptions use different names and labels, but referring to the same entities (e.g. using different languages, Article vs. Artikel, or using synonyms, Article vs. Paper). Semiotic (also known as pragmatic) heterogeneity occurs by differences in interpretation of entities by humans. For instance, a truck is a vehicle, but for one person it is a means of work and for someone else it might be a collector's item. Usually, programmatic heterogeneity occurs by nature, cannot be avoided, but can be limited by defining the domain of interest. However, pragmatic heterogeneity is hard to detect and, thus, is neglected within this thesis. Syntactic heterogeneity can be avoided by defining syntactic rules. In case of service description, this can be done by the definition of a conceptual service model. Service descriptions, which are according to the same conceptual model are based on the same syntactic rules and, thus, do not have any syntactic differences. Considering semantics, heterogeneity of services description occurs by nature. Reasons for that can be differences in the modeling perspective, differences in granularity or differences in coverage. So, semantic heterogeneity cannot be avoided, but it is detectable. Semantic heterogeneity is in focus of this thesis. All in all, the objective can be expressed as follows: Examining proper models, methods, and techniques for measuring semantic heterogeneity between BS and ES descriptions under the condition that syntactic heterogeneity is avoided and pragmatic heterogeneity is neglected. Semantic heterogeneity results in two or more ES, these ES fit the concern of the BS in conjunction and, thus, have to be considered in that way.

* Considering logistics truck has the same meaning within this particular domain usually.
rogeneity will reveal the matching level of particular BS-ES combinations. This will support decision making during ES evaluation.

All examinations and resulting artifacts provided in this thesis aim at being independent from the application domain. However, since the thesis is written within LSEM and since an application domain helps for validation and exemplification, this thesis is dedicated to the logistics domain.

Thesis' objectives and basic idea are stated above. However, there are some conditions and prerequisites that are assumed under which the thesis will be examined. Assumptions are presented next.

2.4 Assumptions

There are seven distinct assumptions the thesis is based on. Each assumption is introduced by a statement and explained in detail below.

A1: Companies business can be represented as a set of business service.

This assumption is foundational for LSEM in general. LSEM assumes that the business of all companies can be structured into functional building blocks and represented as one or many BS. Thus, BS are representations of these functional building blocks, which are structured and described according to the service-oriented paradigm.

A2: Software can be represented as a set of electronic service.

This is a foundational assumption for LSEM too. LSEM assumes that each software system, i.e. infrastructure software, application software, operating system, etc., can be (re-)structured and represented as a one or many ES. ES are the functional building blocks of software, which is structured and described according to the service-oriented paradigm.

A3: A large amount of business services and electronic services are preexisting

In course of this thesis, it is assumed that a large amount of BS and ES are preexisting. Thus, it is neglected that BS and ES have to be acquired and elicited from companies and existing software respectively. The thesis neglects the fact that there might be hard work to “carve out” proper functional building blocks from companies and software products. Further, the question of service granularity\(^\text{10}\) is neglected. It is assumed that BS and ES do have a proper granularity.

\(^{10}\) Service granularity is refers to the overall quantity of functionality encapsulated by a service. (cf. [Erl 2007]) The problem of proper granularity has been discussed several times (et al. [Kulkarni/Dwivedi 2008])
A4: Business services contain business semantics.

BS are representations of functional building blocks of companies business. Thus, a formal description of a BS contains business semantics, i.e. terms and terminology that is understood by people that work within the business domain. Although, this is an obvious fact and not an assumption, it is emphasized here.

A5: Electronic services contain business semantics.

ES are representations of functional building blocks of software. The formal description of an ES may contain business and technical terms. Usually, the description of ES representing infrastructure software, database software or operating system software tend to contain technical terms only. On the other hand, ES representations of application software are more likely to be described by business terms. This is due to the fact that application software is much closer to the business domain and, thus, has to contain business terms in order to be understood by the end user. This might be obvious. However, the assumption “ES contain business semantics” is stated as a requirement in order to sharpen the focus of the thesis. Hence, ES that do contain business semantics are under consideration here. And in turn, ES which do not contain business semantics are neglected. This restriction is necessary in order to justify the last assumption.

A6: Business services and electronic services can be described based on the same conceptual model.

As already stated in the objective section of subchapter 2.3, syntactic heterogeneity of service description can be avoided by the definition of a common conceptual model. A common conceptual model should be feasible for business service and electronic service descriptions at the same time. So, it is assumed that it is possible to describe business services and electronic service based on the same conceptual model.

A7: Electronic services support the operation of a business service if both contain the same or similar business concepts within their descriptions.

The preselection approach is based on the assumption that the higher correspondence between the terminologies of the description BS and ES the higher the probability that an ES supports the operation of a BS. With other words, it is assumed that an ES is more likely to support the operation of a BS, if ES and BS have similar semantic concepts. Hence, the degree of semantic concepts similarities of a particular BS and a particular ES indicates the level of the support of the ES for the BS. This assumption is foundational for all following consideration. Justification is given by two statements and an interpretation of each of them. The first statement is from Shostack: “… what happens in a computer is often analogous to what must happen in order for a [business] service to be successfully rendered”[Shostack 1993]. This statement indicates that there are similarities be-

11 Since, the paper has been written from a marketing perspective, one has to clarify that the term “computer” refers to software (i.e. ES), and that the term service refers services from business perspective (i.e. BS), such as transportation, warehousing etc.
tween processes implemented within a computer program and processes fulfilled by a real-world service. The second statement is from Paruchuri. Paruchuri defines the “SOA Common Information Model” (CIM), which contains a set of information objects and relations with business semantics (i.e. business concepts). Paruchuri states that the definition of a CIM should be domain-specific (focused on a certain business area), but should be strictly independent from a software implementation. Further, he states that a “Real SOA” should have business concepts on business level and on software level as well (cf. [Paruchuri 2007]). Consequently, a good software support is provided only, if the software contains the business concepts of the business domain. Adapted to BS and ES this means that BS and ES should use similar business semantics. As a reverse conclusion, BS and ES fit to each other (in a sense that an ES supports the operation of a BS) if both have similar business concepts. And further, a measurement of the similarity between BS and ES business semantics would quantify the support level of a particular ES for a particular BS.

2.5 Requirements

For preselection of ES for given BS based on measuring semantic heterogeneity a number of generic requirements must be satisfied. Requirements are used as a basis for the conceptual approach (cf. subchapter 2.6). Requirements are described below.

Conceptualization and formalization: Measuring the heterogeneity of subjects such as BS and ES requires that relevant aspects of them are represented explicitly or implicitly. Conceptualization represents an abstract, simplified view of subjects that need to be represented for a certain purpose. It comprises objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them. In addition, formalization is necessary to put conceptualization into a finite form, structure it and express it in terms of language, rules, and axioms. On this basis it ensures that subject conceptualizations are machine-readable and processable. In order to avoid syntactic heterogeneity, conceptualization and formalization has to be based on a common conceptual model.

Measuring: On the basis of formal conceptualizations, measuring heterogeneity levels between BS and ES descriptions is needed. Thus, algorithms are needed in order to conclude about the heterogeneity measures of BS and ES.

Aggregation: Since entities of BS and ES may have different detail levels and different levels of abstraction, heterogeneity is measured on different levels as well. This leads to several result sets, which can be represented as multi-dimensional vectors. In order to conclude about an overall heterogeneity measure, aggregation of results from detail level towards an abstract overall measure is
needed. Aggregation logic should be capable of balancing the importance of entities towards an aggregated value, i.e. by assigning weights.

Analysis and Visualization: The results of determining heterogeneity for a large number of BS and ES may be still complex and due to its numeric and multi-dimensional character difficult to interpret by humans. Thus, a visualization metaphor for representation of large scale heterogeneity measures and formatting them towards easy interpretation by humans for decision making is required. Visualization should maintain the multi-dimensional information level and support a human decision maker by means of colors, shapes, characters etc.

(Semi-)Automation: Since the selection and evaluation of ES for a certain BS can be fulfilled manually as well, the approach should facilitate automation in order to gain cost and time efficiency. However, since full automation may lead to error-prone results the approach should leave space for manual steps and interpretations.

2.6 Conceptual approach and solution pathway

Based on the objectives, assumptions and requirements the conceptual approach is presented in this section. The conceptual approach shows basic constituents and an overall solution from high-level perspective. This works as a pathway for detailed examination in the following chapters. The conceptual approach is depicted in Figure 2.4. Each constituent is depicted in the middle. Requirements are depicted on the right hand side.

For conceptualization and formalization purposes and in order to avoid syntactical heterogeneity a conceptual model for BS and ES service description is necessary. The conceptual model is termed as service concept meta-model (SCMM). The SCMM is presented on top of Figure 2.4. It contains basic concepts that are valid for BS and ES descriptions. Basic concepts can be, for instance, processes, resources, actions, and stakeholder. Further, each concept has concept properties defined in the SCMM as well. A detailed examination of service representation is provided in chapter 3. As a result chapter 3 presents a SCMM that is valid for BS and ES description in course of this thesis.

In order to describe BS and ES a method is needed that derives information from existing BS and ES. That means, BS and ES information need to be modeled based on the SCMM. Models for BS and ES descriptions are termed business service models (BSM) and electronic service models (ESM). As a result, i.e. after applying the method, there are several BSM $x_i$ and ESM $y_j$, where $i$ represents the index of one particular BSM and $j$ represents the index of one particular ESM. All $x_i$ and $y_j$ are instances of the SCMM and, thus, are according to the concepts and properties defined in the SCMM. Instances $x_i$ and $y_j$ contain several BS entities $n_{i,k,o}$ and ES entities $m_{j,l,p}$, where $o$ is the index of one particular entity of the BSM $x_i$, $p$ is the index of one particular entity of the ESM $y_j$, and $k$ and $l$ are the indices of the service concepts defined in the SCMM the entities belong to. A method for
service modeling as well as resulting BSM and ESM including entities are examined and described in detail in chapter 4.

Figure 2.4 Conceptual approach

After conceptualization and formalization, algorithms are needed in order to measure heterogeneity between BSM $x_i$ and ESM $y_j$. Basically, three steps are necessary. First, BS entities $n_{i,k,o}$ and ES entities $m_{j,l,p}$ need to be extracted from BSM and ESM and grouped in order to apply algorithms on comparable entities only. Comparable entities are entities that belong to the same service concept (i.e.
k=l). Entities that belong to different service concepts (i.e. k≠l) are not comparable, i.e. a stakeholder entity and an action entity are not comparable. Second, algorithms for correspondence measurement are applied on service entities. All BS entities \(n_{i,k,o}\) need to be compared to all comparable ES entities \(m_{j,l,p}\) resulting in a correspondence value \(ce_{i,j,k,l,o,p}\) for each \(n_{i,k,o} - m_{j,l,p}\) combination. The correspondence value \(ce_{i,j,k,l,o,p}\) is a quantification of the similarity between a certain BS entity \(n_{i,k,o}\) and a certain ES entity \(m_{j,l,p}\) where both entities belong to the same service concept (i.e. k=l). Since there are many possible entity combinations for each BS-ES-combination automation is needed. Third, aggregation algorithms need to be applied in order to aggregate correspondence value \(ce_{i,j,k,l,o,p}\) on entity level to correspondence values on levels above. Two aggregation levels are thinkable: aggregation on service concept level and aggregation on service level. A correspondence value on service concept level \(cc_{i,j,k,l}\)\(^{12}\) aggregates correspondence values on entity level, i.e. \(ce_{i,j,k,l,1,1}, ..., ce_{i,j,k,l,1,1}, ..., ce_{i,j,k,l,1,1}\), to a single value for each service concept defined in the SCMM. The correspondence value on service level \(cs_{i,j}\) aggregates the correspondence values on concept level, i.e. \(cc_{i,j,1,1,1}, ..., cc_{i,j,1,1,1}, ..., cc_{i,j,1,1,1}\), to one overall correspondence value of a particular \(x_i\), \(y_j\) combination. Since there are many values on entity, concept and service level, automation is needed once again. Each algorithm is expressed by a function \(f(...)\) as shown in Figure 2.4. All correspondence measure related topics, i.e. extraction, matching and aggregation, are addressed in chapter 5 in detail.

Since there are many numeric correspondence values on different level of abstraction, visualization is needed for analysis purposes. This encompasses visualization of correspondence values on service, service concept, and service concept entity level for several BS-ES combination. This is sketched at the bottom of Figure 2.4. However, visualization and analysis will be examined in detail in chapter 6.

Before starting with the first point of the conceptual approach, i.e. service representation, in chapter 3, the next subchapter addresses state of the art and related work.

### 2.7 State of the art and related work

Electronic service evaluation and discovery is an important topic in SOA. The SOA basic model addresses this by an own entity called service registry, which is also known as service broker (cf. Figure 2.1). A service consumer can discover the registry in order to find a service that fits its concerns (cf. [Papazoglou 2007]). Basic mechanisms, such as UDDI [Clement et al. 2004], support the search for electronic services based on keywords only. However, keywords based discovery is rather imprecise. There are other more sophisticated approaches based on semantic Web service (i.e. ES) description, such as WSMO [Roman et al. 2006b] or OWL-S [Martin et al. 2004]. Semantic Web service discovery is much more pre-

---

\(^{12}\) Since only comparable entities are considered, k and l are equal (i.e. k=l).
cise, since it is based on the definition of goals. These goals have to be defined explicitly in terms of functional and non-functional parameters. However, there are divergent opinions about goals, since “service requesters are not expected to have the required background to formalize their goals” [Keller et al. 2005]. This is due to the fact that business people (i.e. potential service requestors) know their business, but usually do not know what to expect from a Web service (i.e. an electronic service). Further, goals are defined by preconditions, assumptions, postconditions and effects (cf. [Roman et al. 2006b]) using logical languages, but business people are not familiar with logical rules. One might think that business services can be similar to goals, but they are not. Goals define expectations on a (sought-after) Web service explicitly. In contrast, BSs describe business cases, which might implicitly contain requirements on an electronic service. With this, the approach for preselection of ES by given BS based on semantic concept correspondence fits in the gap between a keyword based service discovery and service discovery mechanisms based on semantic Web services (cf. Figure 2.5). This approach is more precise than keyword based discovery, since not keywords only, but the terminology and semantic concept are considered. In contrast it is less precise than a sophisticated semantic Web service discovery. However, the approach is reasonable for preselection purposes, i.e. reducing a large number of available electronic services to a significant smaller set, which makes evaluation less complex and less expensive. Furthermore, it does not rely on goals which might be hard to define and which are described in a language business people do not understand.

![Figure 2.5 Relation keyword based search, the approach, and semantic service discovery](image)

The approach is similar to [Markovic/Karrenbrock 2007], where business process models are used for electronic services discovery. The approach of [Markovic/Karrenbrock 2007] rely on one common ontology model, which avoids semantic heterogeneity per-se. However, a common ontology including all ontology concepts that span whole business service areas and electronic service areas at same time is not realistic. The variety and diversity of those areas is too big. Further, [Markovic/Karrenbrock 2007] does not provide any matching or aggregation algorithm, and it does not aims visualization for preselection purposes.
2.8 Summary

This chapter started with a section about background knowledge, which included service orientation, logistics and the LSEM research framework. This thesis is embedded in LSEM and, thus, related to other LSEM research topics. Relation of the thesis within LSEM had been explained. Key research objects, i.e. business service (BS) and electronic service (ES), have been introduced. Afterwards, problem and motivation for the thesis had been described in detail, main objectives had been derived from that. In turn, the basic idea was formulated. Next, a section about assumptions that constraint and narrows the scope of the thesis had been introduced, explained, justified, and discussed. Based on distinct requirements that had been derived from objectives, basic constituents are presented. Basic constituents make up the conceptual approach and work as a pathway for the following chapters. Each constituent as well as its relation to the requirements, its position within the overall approach, and the connection to one of the following chapters had been presented. Last not least this chapter ended with a section about state of the art and related work. The next chapter addresses the first point of the conceptual approach - service representation.
3 Service representation

This chapter is dedicated to the representation of services. The statement “There is no one correct way to model […] - there are always viable alternatives.” [Noy/McGuinness 2001] holds true for service representation. The statement continues: “The best solution almost always depends on the application that you have in mind […]” [Noy/McGuinness 2001]. So, in order to find a proper service representation, one has to define distinct criteria. Thus, service representation requirements for application within this thesis are setup first. Next, two general approaches for information representation, i.e. models and ontologies, are examined for their applicability for service representation. Afterwards, existing service representations approaches are observed. Service representations from business and technical area are considered. Basic service concepts are derived from both areas. At the end, this chapter presents a meta-model for service representation, which is base on existing approaches and fulfills the requirements.

3.1 Requirements for service representation

Requirements for service representation are introduced in the following. Requirements are listed and abbreviated with RRx for identification purposes. RR stands for ‘representation requirement’ and x for a consecutive number. Each requirement has a title and an explanation. The words ‘have to’ (or ‘has to’ respectively) and ‘should’ are used. ‘Have to’ means that the requirement is compulsory, i.e. a service representation that do not fulfill this requirement is not suitable. ‘Should’ means that the requirement is desirable, but not mandatory. A SCMM, which does not fulfill a ‘should’-requirement, can be used for further consideration, even though it may not be optimal or convenient. Each requirements is presented and explained next.

RR1: Service representations for BS and ES have to rely on a conceptual model.

BS and ES have to be described based on a conceptual model. The conceptual model contains basic concepts including properties, cardinalities, types, constraints and other syntactic rules. The definition of these rules is necessary in order to prevent syntactic heterogeneity of BS and ES representations.

RR2: A conceptual model has to be applicable for BS and ES

The conceptual model required in RR1 has to be applicable for BS and ES. So, the conceptual model has to contain basic service concepts that are applicable for BS and ES at same time. A common conceptual model is needed that ensures basic comparability between BS and ES.
RR3: Service representations have to be formal and machine-readable

Algorithms that conclude about semantic heterogeneity between BS and ES will be applied. Since algorithms are implemented using computing functionality, service representations have to be formal and machine-readable. Otherwise, automatic computation would not be possible.

RR4: Service representations have to conform with ontologies

There are existing algorithms for measuring heterogeneity in the area of ontologies matching [Euzenat/Shvaiko 2007]. In order to apply these algorithms, service representations have to be conform with ontologies.

RR5: Service representations should contain sound semantics

Since existing ontology matching approaches rely on sound semantics, i.e. they rely on names, identifiers, descriptions, synonyms, hierarchies (i.e. taxonomies), and relations between concepts, service representations should support these as well. Although it is not mandatory that service representations contain all of these constructs, it is advisable and helpful to use as many as possible.

RR6: Service representation for BS and ES should consider existing service representation approaches.

Since there are many approaches available and since many researchers had already thought about service representations, the service representation used here should consider existing approaches. Although this is not mandatory, i.e. neglecting existing service representation approaches would work for further processing as well, it is advisable to consider existing service representation approaches in order to have a sound basis and being conform to other research endeavors.

RR7: Service representations for BS and ES have to consider functional aspects of services.

Since this thesis is focused on functional aspects only, service representation for BS and ES have to consider functional aspects. Even though this does not exclude other criteria, i.e. non-functional aspects (e.g. quality of services, pricing, legal, reliability, etc.), it is helpful being focused on functional aspects only.

RR8: Service representation should be simple and not too complex.

Service representations for BS and ES should be less complex and less complicated. Even though not all aspects of a service can be modeled, the representation should seek in being simple and applicable. This refers to using not too many perspectives, not to many service concepts, and further no mathematical rules. Mathematical rules such as logical expression are being hardly understood by a layperson. Especially participants from logistics are not familiar with logical rules and logical expression. Thus, the conceptual model and underlying models should be focused on terminological expression instead. Further, the service representation should contain only one model and no sub-models, modules or perspectives. Since sub-models, modules and perspective lead to complexity, the representation should be kept simple.
should be described by one model for each service only. Sure, sub-models can foster modularization, but one tends to put more information than necessary into each of the sub-models. Thus, in order to comply with the simplicity of the service representation there should be one conceptual model only.

The fulfillment of these requirements will lead to a proper service representation for this thesis. Requirements will be referred at given point of interest in the following subchapters. Next, two basic ways of information representations, i.e. models and ontologies, are examined in detail.

### 3.2 Models and ontologies

Service representation is about representing information about services. Basically, there are two ways of information representation: models and ontologies\(^\text{13}\) [Atkinson et al. 2006]. Both are different in their origin, both have different characteristics, both have strength and weaknesses, but both are similar as well. In order to prevent confusion models and ontologies are explained. They are examined and compared in order to decide, which of both is more appropriated for service representation in course of the thesis.

#### 3.2.1 Models

The term model is used in different disciplines, such as information technology, mathematics, fashion industry, and more. Information technology or more precisely the software engineering [Sommerville 2010] discipline is focused here. All other areas are neglected.

According to Stachowiak [Stachowiak 1973] a model is defined as a representation of something. Since this is a very generic definition, Stachowiak [Stachowiak 1973] added three characteristics of a model that should explain the definition in more detail:

- **Mapping characteristic:** A model is based on an original and is a representation of that. The original can be a model as well.
- **Reduction characteristic:** A model shows only these aspects of its original, which are relevant for the creator or user of the model.
- **Pragmatic characteristic:** A model is created for a certain purpose and can be taken as a substituted for the original.

Concerning the first characteristic, mapped aspects can be structures, objects, attributes, relationships, the behavior and more [Kastens/Büning 2008]. However, the decision about these aspects and the occurrence of a model resides to the modeller and to the usage of the model. Individual objectives of stakeholders lead to

\(^{13}\) Service representation comprises service models and service ontologies.
the fact that there is no intersubjectivity of a model. The relationship between
original and model is restricted to the purpose of a particular stakeholder. In order
to cope with the subjectivity of models Stachowiak [Stachowiak 1973] postulates
four questions. These questions should guide and support the modeling process.
The four questions are: From what (original is the model)? For whom (is the mo-
del and what is the benefit for this person)? When (or more precisely, what is the
validity time period of the model)? and What for (or more precisely, what is the
purpose of the model)?

Models can be classified into descriptive models and prescriptive model. If a
model is a mirror of an existing original it is termed as descriptive. Unlike a
descriptive model, a prescriptive model is used as a specification for something to be
created. A descriptive model is similar to a construction plan, whereas a descrip-
tive model is “like a photo” of its original. However, a model can be a descriptive
model and a prescriptive model at the same time. Modeling a database schema
from an existing database for documentation purposes leads to a description mod-
el. The modification of the description model in order to add functionality will
lead to a prescriptive model, since this model will be used as a construction plan.
A model that is first descriptive and later prescriptive (or the other way around) is
called transient model [Ludewig 2003].

Models can be found in all areas of software engineering, e.g. design models (such
as Class Diagrams, Object Diagrams, Entity Relationship Models [Thalheim
2000], etc.) process models (such as Sequence Diagrams, etc.), information dia-
grams, models for user interaction (like Use Case or Interaction Diagrams),
process maturity model (like CMM [Paulk 1995] or SPICE [Van Loon 2007] etc.)
and more14. Most of the models are prescriptive models, since software engineer-
ing is dedicated to software development. Thus, models work as construction
plans in order to support the software development process. A software engineer
starts with an abstract specification, followed by a detailed specification and an
architectural design. Afterwards, the source code is implemented and the software
is deployed and executed. Each of these steps is supported by certain documents
and prescriptive models. However, descriptive models, such as source code docu-
mentation, testing protocols and user manual will be created as well. Descriptive
models are relevant for maintenance, usage, and for documentation purposes, i.e.
when modifications are necessary later on [Ludewig 2003].

The model chain above shows a rough idea of model to be created during software
development. A more detailed consideration about models is made in Model Driv-
en Architecture (MDA). MDA is proposed by the Object Management Group
(OMG) and suggests a model driven software development approach including a
clear distinction between functional and technical aspects.

14 All mentioned model that do not have a reference refer to Unified Modeling Language (UML)
models [Object Management Group (OMG)].
3.2.1.1 Model Driven Architecture

Model Driven Architecture (MDA) is a model driven software development approach proposed by the Object Management Group (OMG) [Object Management Group (OMG) 2003]. MDA suggests using models for building software systems. According to MDA, other software development approaches using models lack in mixing business information and technical information in one model. Instead, MDA proposes a separation of business and technical information and provides different model levels. MDA defines four models levels: Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM) and a code model.

CIM, also called business model or domain model, describes the situation the system will be used as well as the requirements of the system. However, a CIM has to be independent from system implementation, i.e. it should hide as much information about automated data processing the system will fulfill later on. A CIM is useful for presenting what the system is expected to do, helps by understanding the problem area, and works as a shared vocabulary for PIM and PSM. Thus, a CIM should be traceable to the constructs of PIM and PSM and vice versa. A CIM may contain several models that can be created by using Unified Modeling Language (UML) [Object Management Group (OMG) 2011a], e.g. Use Case diagrams. A PIM describes the system, but hides technical details about the platform. A PIM might be described using business process models, such BPMN [Miers/White 2008] or UML sequence diagrams. Based on PIM description, a platform is evaluated. This platform has to fulfill all functional and non-functional requirements. After having chosen a proper platform, a PSM will be created. A PSM describes the system according to the selected platform by considering platform specific details. MDA proposes models based on UML [Object Management Group (OMG) 2011a] and Object Constraint Language (OCL) [Object Management Group (OMG) 2006a] for that purpose [Object Management Group (OMG) 2003]. Last not least, source code, i.e. the code model, has to be implemented that is according to the specification provided by PIM, PSM and CIM, i.e. the code model has to fulfill all system functionalities. MDA aims at achieving platform independence, system independence and programming language independence by separating models according to their business and technical level. From systems' point of view, models on business level, i.e. CIM and PIM, are more abstract then models on technical level, i.e. PSM and code model. Thus, model levels are also considered as abstraction levels.

Beside the definition of abstraction levels MDA proposes the transformation of models (cf. [Object Management Group (OMG) 2008]). Transformation means that elements of the target model are created from the elements form a source model. Usually, transformation is done from an abstract model to a concrete model along the CIM-PIM-PSM-code chain. Transformation can be done manually, with computer assistance or automatically (cf. [Object Management Group (OMG) 2008]).
All in all MDA is a strategy provided by the OMG that fosters the development of platform and vendor independent specifications as well as interoperability and portability of software systems. Meta-modeling plays an important role in MDA [Atkinson/Kühne 2002]. However, meta-models are important for modeling in general. Thus, the meta-model definition, the relation between meta-models and modeling languages as well as the usage of meta-modeling in software engineering are examined next.

3.2.1.2 Meta-models and modeling languages

A meta-model is defined as “model of a model” by [Miller/Mukerji 2003]. This definition is rather imprecisely, but states the basic idea of meta-modeling. A finer-grained definition is provided by Stahl and Völter. [Stahl/Völter 2005] states that a meta-model describes the structure of a possible model including constructs, relationships between as well as restrictions and modeling rules. A model has to stick to these rules in order to be compliant with the meta-model. All models that are built on the basis of a meta-model are somehow similar and can be compared.

A meta-model describes structures in an abstract way. The concrete syntax of that is provided by the modeling-language. Where the abstract syntax (i.e. the meta-model) provides the foundation for the automated and tool-supported processing of a model, the concrete syntax provides concrete constructs in a textual or graphical manner that is used for modeling [Petrasch/Meimberg 2006]. Meta-model and modeling-language have a one-to-many relationship, i.e. a meta-model can be represented by several textual or graphical syntaxes [Stahl/Völter 2005].

For instance, UML class diagram [Object Management Group (OMG) 2011a] is based on a meta-model that defines the basic constituent such as packages, classes and attributes. A model that complies to the UML meta-model contains a Vehicles package with a class Truck containing two attributes called type and year_of_construction, for instance. This model can be represented in different languages, e.g. using a graphical notation or in as XML Metadata Interchange (XMI) [Object Management Group (OMG) 2011b]. A graphical notation is illustrated in Figure 3.1. The same graphical notation represented in XMI is shown in Listing 3.1.

![Figure 3.1 UML Class diagram example](image-url)
The example above shows the relationship between models, meta-models, and languages. However, a meta-model may have a meta-meta-model that provided the structure of the meta-model in turn. Hence, in order to cope with meta-modeling, Object Management Group defined the meta-data architecture called Meta Object Facility (MOF), which is presented in the following.

### 3.2.1.3 Meta Object Facility

Meta Object Facility (MOF) [Object Management Group (OMG) 2005] had been introduced by the Object Management Group (OMG) as a meta-data architecture for modeling. MOF provides a general understanding and a foundation for meta-modeling. Each model that is conform to MOF can be transformed into another model that relies on MOF as well. Further, MOF models can be held in the same model repository and can be created and processed with the same tools. As such MOF, does not provide a meta-data architecture only, but tools and techniques for building and processing models. MOF supports the formal definition of models and their application, such as needed by Model Driven Architecture (MDA). MOF provides four layer meta-data architecture, which is illustrated in Figure 3.2.
The four layers are (from top to down) M3 meta-meta-model layer, M2 meta-model layer, M1 model layer and M0 the information layer:

- **M3**: The meta-meta-model layer defines structure and semantics of the meta-meta-data. As such the meta-meta-model provides metadata for meta-model layer. MOF provides constructs for meta-meta-modeling itself.

- **M2**: The meta-model layer comprises structure and semantics of meta-data. A meta-model is an instance of the meta-meta-model and provides metadata for the model layer. For instance, the definition of the constructs of a UML Class diagram, such as packages, classes and attributes, belongs to this layer.

- **M1**: The model layer comprises structure and semantics of a model. A model is an instance of a meta-model and provides metadata, i.e. the structure, for data in the information layer. A model, which is created by a developer using UML constructs, for instance, belongs to the model layer.

- **M0**: The information layer comprises data using the structure of the model. Instance data of a model belong to this layer.

After the explanation of all relevant aspects regarding MOF, modeling, meta-modeling, and its usage within MDA as well, information representation using ontologies is examined next.

### 3.2.2 Ontologies

The term ontology has its origin in the Greek philosophy [Aristoteles 350 B.C., Platon 375 B.C.]. Today it is still known as the study of the nature of being, existence, and reality. Ontology deals with questions about what entities exists in the world, how these entities may be grouped together, related to each other, and subdivided according to their similarities and differences [Loux 2006]. However, in the middle of the 20th century, the philosophical understanding of ontology had
been adopted by information technology, i.e. more precisely in the area of artificial intelligence [Jackson 1985]. In this area ontologies are understood as artifacts that are used for knowledge representation and aim at making knowledge understandable and interpretable for machines.

The most popular ontology definition in this area is given by Gruber [Gruber 1993]: "An Ontology is an explicit specification of a conceptualization." Since this definition is very broad, researchers extended this definition and highlight certain aspects. A well-known version is provided by Studer [Studer et al. 1998]: "An ontology is a formal, explicit specification of a shared conceptualization. Conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group." This definition highlights the formality and commitment aspect of ontologies. Formal representation is necessary for machine-to-machine communication and reasoning about knowledge. The commitment aspect refers to consensual knowledge, i.e., knowledge that is accepted by a group, which minimizes ambiguity. Both aspects are core of many knowledge representation applications, such as the Semantic Web [Berners-Lee et al. 2001].

Another more implementation-oriented definitions is provided by Maedche [Maedche 2002]: "[…] an ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary. Usually a form of first-order-logic theory is used to represent these assumptions, vocabulary appear as unary and binary predicates, called concepts and relations, respectively." This definition refers to ontology as an artifacts that contain a vocabulary and a set of assumptions about this vocabulary. The vocabulary is represented by a set of concepts and a set of relations between these concepts. Assumptions about the vocabulary are usually represented by logical constraints according to the first-order-logic [Smullyan 1995].

3.2.2.1 Ontology constituents

As suggested by the definition of [Maedche 2002], the main constituents of an ontology is a vocabulary, containing concepts and relations, and a set of assumptions about this vocabulary. Concepts refer to entities that exist. They are also known as classes, sets, collections, frames, or types. Concepts can be structured in hierarchy, i.e., concepts may have sub- and super-concepts. For instance, the concept Truck may have a super-concept Vehicle and the sub-concepts SmallTruck, MediumTruck and BigTruck. A hierarchy of concepts is known as a taxonomy [Bailey et al. 2005]. Further, there are relations between concepts. Relations are also called properties, attributes, or slots [Noy/McGuinness 2001]. As the name suggests relations define relations between concepts and concept properties. For
example, a Vehicle may have a property called hasDriver, which refers to a concept called Driver. Both Vehicle and Driver are related by the property hasDriver. From the properties point of view, the former is called concept domain and the latter concept is called range, i.e. hasDriver refers to the domain Vehicle and the range Driver. Assumptions about the vocabulary are basically restrictions and constraints, such as cardinalities, value ranges, or logical constraints. For example, “a Vehicle has zero or one Driver” is typical cardinality constraint statement, “the age of a Driver is at least 18” is a typical value range statement, and “an empty Truck with a Driver is ReadyForTransportation” is a logical constraint statement. Each of this statement can be claimed as an assumption about the vocabulary. Not mentioned yet\(^\text{15}\), ontologies do contain individuals. An individual represent a specific instance of a concept including all relations and according to the defined constraints. For example, a “Scania R500”\(^\text{16}\) is an individual of BigTruck and “MaxMuster” an individual of Driver. Concepts, properties, and instances are illustrated in Figure 3.3 by example.

![Ontology example](https://example.com/ontology.png)

Figure 3.3 Ontology example modeled with Protégé [Protege]

These are the basic constituents of an ontology. However, there are different forms of representation in terms of ontology languages.

### 3.2.2.2 Ontology languages

Ontologies are represented in formal languages. Formal languages make ontologies machine-readable and understandable to software. An ontology language contains the ontology constituents mentioned above. However, there are different ontology languages. Usually, they are categorized by their expressiveness\(^\text{18}\) into frame-based [Minsky 1975, Nebel 1999] and description-logic [Baader et al. 2003] languages. Moreover, since ontologies became “mainstream” in semantic web area [McGuinness 2003], there are ontology languages based on web languages as well [Lacy 2005]. In the following name-based languages, description logic languages and web ontology languages are introduced briefly.

---

\(^\text{15}\) Even not in the definition of [Maedche 2002].


\(^\text{17}\) Color of the arrows show the type of the relation, i.e. purple means subclass, blue means individual, orange is the relation hasDriver.

\(^\text{18}\) Expressiveness refers to the richness of a language for making statements [Lacy 2005]
Frame-based languages have long history in computer science and artificial intelligence. Examples for those are F-logic [Kifer et al. 1995], OKBC [SRI International 1995], and KM [Clark/Porter 2003]. Frame-based languages use concepts, which are called frames, and concept properties, which are called slots. Furthermore, there are fillers, which are values of frame slots. A frame with fillers can be constituted as an individual. Frames, slots and fillers are the modeling primitives, which make frame-based languages similar to object-oriented modeling [Kifer et al. 1995]. Frame-based languages are less expressive, but easy to handle.

Description logic (DL) languages are more expressive than frame-based languages. DL is a mature knowledge representation technique using concepts, roles (i.e. properties) and individuals. Furthermore, DL uses axioms as fundamental modeling primitives. Axioms are statement about concepts and roles. This is different from frame-based languages, where specifications about frames and slots completely declare concepts and properties. In contrast, DL axioms make statements about concepts and roles, which constraint and specify concepts and roles in more detail. Axioms enrich the expressiveness of DL languages [Grau et al. 2008]. Traditionally, axioms are declared by logical expression using first-order logic [Smullyan 1995]. However, there are languages, such as OWL that declare distinct constructs for axiom statements. Examples for DL ontology languages are KL-ONE [Brachman/Schmolze 1985], KIF [Genesereth 1998], RACER [Haarslev/Müller 2001] and OWL [World Wide Web Consortium (W3C) 2004a].

Semantic Web had been introduces by Berners-Lee as an extension of the Web “in which information is given well-defined meaning, better enabling computers and people to work in cooperation” [Berners-Lee et al. 2001]. The main purpose of the Semantic Web is to provide structured and formal information in order to leverage the drawbacks of traditional Web [Lacy 2005]. Ontologies had been selected for that Semetic Web purposes. Thus, ontology languages that cope with the requirements of the Semantic Web, had been developed. Web Ontology Language OWL emerged as the dominant language for that purpose. OWL is based on RDF syntax [World Wide Web Consortium (W3C) 2004b] and comes in three flavors: OWL Lite, OWL DL and OWL Full. OWL Lite is the simplest version of OWL. OWL Lite contains classes (i.e. concepts), properties (i.e. relations) and constructs for class and properties hierarchies. Further, there is a subset of constraints defined, e.g. cardinalities. OWL DL, i.e. OWL Description Logic, contains the full expressiveness of OWL. All possible OWL constructs of OWL Lite and additional constraints are defined within OWL DL. For instance, there are constructs for declaring classes and properties equivalent or disjoint, there are constructs for declaring properties transitive, symmetric, asymmetric, functional, inverse-functional, reflexive or irreflexive (cf. [Horridge 2011]), and more. However, OWL DL defines some limitations that ensure decidability, i.e. that a reasoner

---

19 A reasoner is a computer program that infers about new knowledge from existing ontology knowledge. Additionally a reasoner is used for consistency check and querying. Known reasoner implementations are KOAN2 [KOAN2], Pellet [Pellet], FacTC [FacTC], and RACER [RACER].
can infer new knowledge from exiting ontology knowledge. One limitation is, for instance, that a class cannot refer to an instance through a property. Last not least, OWL Full provides the full expressiveness of OWL DL, but does not define any limitations. Thus, OWL Full is not decidable, i.e. a reasoner cannot infer about new knowledge from an OWL Full ontology.

3.2.2.3 Ontology types

Ontologies are built with the aim of fostering the reuse of knowledge. Many ontologies with different purposes and scopes had been build so far. One can distinct ontologies that are more general and ontologies that are more specific. Based on this characteristic one can distinct different ontology types. [Guarino 1997] mentions four ontology types: top-level-, domain-, task- and application ontologies. These types are illustrated in Figure 3.4. Rectangles represent ontology types. Arrows represent specialization relationships.

![Ontology types](image)

*Figure 3.4 Ontology types (cf. [Guarino 1997])*

Top-level ontologies (also known as foundational ontologies or upper ontologies) describe very general concepts, such as time, space, matter, action etc. These concepts can be used across many domains and task. Due to the generality of concepts, top-level ontologies are usually not used within applications directly. Instead, they are referred in other ontologies as general knowledge. Examples for top-level ontologies are SUMO [Niles/Pease 2001] and DOLCE [Gangemi et al. 2002].

Domain ontologies describe a vocabulary of a domain, such as medicine, biology, geography, or logistics. Task ontologies describe generic tasks, such as diagnosing, selling, or ordering. Domain and task ontologies refer to top-level ontologies and specialize top-level terms. In this way, domain and task ontologies have a much narrower and more specific scope than top-level ontologies. Domain ontologies in the area of logistics, for instance, had been considered by Leukel [Leukel/Kirn 2008] and Lian [Lian et al. 2007].

Application ontologies specify concepts depending on a particular domain, a particular task, and a specific application context. A use case for an application ontology is, for instance, the order management task in the logistics domain for certain company.
Considering Figure 3.4, the lower ontologies are more specific and have a narrower scope of application. And the upper ones are more general and, thus, have a broader potential of reuse.

After having introduced the essence of models and ontology, both are compared in the following in order to conclude, which of both is more appropriated for service representation in course of this thesis.

### 3.2.3 Models vs. ontologies

Both, models and ontologies, can be used for service representation. Thus, the question arises, which one to choose for the purpose of this thesis. In order to answer this question one has to be clear about differences of models and ontologies, their intended usage, and their strength and weaknesses.

As mentioned models are prominently used in software engineering and ontology are prominently used in the Semantic Web area today. However, both can be used for information representation and both can be used in conjunction as well (e.g. [Roman et al. 2006b, Terrasse et al. 2006, Brockmans/Haase 2008]). Hence, ontologies and models might not be that far away from each other. In the following the most prominent distinction criteria between models and ontology are examined. Referring to Atkinson [Atkinson et al. 2006] these criteria are used in order to draw a conclusion about the usage of models and ontologies within this thesis. Most distinctions between models and ontologies had been made about their purpose and features. They are based on the premise that models and ontologies are inherently different and that there are objective criteria for their distinction. These distinction criteria (DC) are listed and refuted below:

**DC1: Models focus on realization, ontologies do not**

This statement was made implicitly by [Noy/McGuinness 2001]. [Noy/McGuinness 2001] state that ontologies are focused on capturing abstract domain knowledge in terms of concepts and relationships, while models are focused on realization, e.g. used for building software. This might be true for some cases (e.g. for Entity-relationship models [Thalheim 2000]). However, it is not true in general, since there are models that are focused on domain modeling as well. For instance, MDA [Miller/Mukerji 2003] proposes the use of models at all levels of abstraction. There are models, which are intended to be used for domain modeling exclusively, e.g. the Computation Independent Model (CIM), and not for realization purposes. On the other hand there are approaches that suggests the usage of ontologies for construction purposes as well, such as the Semantic based Requirements Engineering [Riechert et al. 2007].

**DC2: Ontologies are for run-time knowledge exploitation, models are not**

This statement refers to the fact that ontologies may contain instances data and “…, currently MDA has not been applied for run-time relevant characteristics” [World Wide Web Consortium (W3C) 2006]. However, this does not mean that
models are not capable or incompatible of storing instance data. Instead, there are endeavors providing MOF repositories for run-time access to model data [Hoessler/Soden 2004].

**DC3: Ontologies are for representing web based information, models are not**

This refers to the association of ontologies with OWL and the Semantic Web. However, Semantic Web is not an aspect of ontologies per se. Semantic Web is only one application area of ontologies. There are other application areas, which are base on languages, e.g. FLogic [Kifer et al. 1995], that are not related to the web. Thus, ontologies support the realization of Semantic Web, but the Semantic Web is not a distinct feature. Further, there are endeavors from MDA to support Semantic Web as well (cf. [Frankel et al. 2004]).

**DC4: Ontologies are formal, models are not**

There are a lot of papers that distinguish models, represented by modeling languages such as UML, from ontologies, represented by OWL, by their level of formality (e.g. [Brockmans et al. 2004,Djuric et al. 2005]). In this context, formality does not mean machine-readable. Instead, formality means the explicit and precise description of formal semantics using axioms in a way that some reasoning logic can conclude about inconstancies and infer about new knowledge. This might be true considering UML and OWL only. However, this holds not true for models and ontologies in general. First of all, there are ontology representations that are less formal. For instance, ontology languages based on frames do not provide any axioms. Second, there are modeling languages that are more formal and provide precise semantics. For instance, the Object Constraint Language (OCL) [Object Management Group (OMG) 2006a] as an enhancement for UML provide formal semantics [Richters/Gogolla 2000]. This makes UML models as formal as OWL ontologies [Atkinson/Kiko 2006].

**DC5: Ontologies do support reasoning, models do not**

The statement that reasoning is not possible for models is derived from the previous statement, i.e. models are less formal than ontologies. And indeed, models such as UML were not initially devised for reasoning. However, since models can be as formal as ontologies, there is nothing that stops applying reasoning on models as well. For instance [Calì et al. 2002] provides a formal framework for reasoning on UML class diagrams.

**DC6: Models use the close world assumption; ontologies use the open world assumption**

This statement refers to different presumptions for reasoning about models and ontologies. Close world assumption is based on the presumption that everything what is not known to be true is false. In contrast, in open world assumption everything what is not known to be true is undefined (cf. [Mazzocchi 2005]). Indeed, the statement above is correct considering UML models and OWL ontologies only. However, this is not correct in general. For instance, there are ontologies using Prolog language that are explicitly based on closed-world assumption. Fur-
ther, there are models such as MDA Computational Independent Model that are not used for realization and, thus, could be based on open-world perspective [Atkinson et al. 2006].

None of these statements about the distinction of models and ontologies is right or wrong per se. Most of the statements rely on particular languages, such as OWL and UML, but are not true in general. In most cases, these statements are cited by author, in order to provide reasons for labeling a particular information representation as a model or and ontology. However, most of the decision relies on predilection and the authors’ background, i.e. if an author is more related to software engineering or more to the artificial intelligence (AI) or Semantic Web community. Anyway, these statements and their explanations do not help for the decision, which one to choose for service representation. However, one can infer the following facts from the six bullets mentioned above and further research papers:

- Ontologies and models are similar in their motivation of information representation.
- Distinctions between ontologies and models rely on a particular representation language.
- Ontologies and models can be used in conjunction, e.g. in WSMO [Roman et al. 2006b], or in MOF based meta-model for OWL [Brockmans/Haase 2008].
- Models and ontologies can be transformed into each other without losing any information (e.g. ER2OWL [Fahad 2008], DB2OWL [Cullot et al. 2007] and UML2OWL transformation [Slezak et al. 2009]). Thus, certain models can be denoted as ontologies and vice versa.

These facts testify that models and ontologies can be treated equally for information representation and thus for service representation as well. This means that a service representation, which is based on a modeling (e.g. on MOF), can be still formal, machine-readable and conform to ontologies (as required in RR3 and RR4). Although, these facts do not answer the question, which of both is more appropriated for service representation.

Next existing service representation approaches are examined. This will help for deducing, which of both (modeling or ontologies), is used more often for service representation. Further, the next chapter will illuminate basic service concepts for BS and ES representation (as required in RR1, RR2 and RR6).

3.3 Existing service representation approaches

There are several existing approaches for representing services. However, in general, there is no common understanding and no widely accepted services representation approach. In order to provide a proper conceptual model (as required in RR1), which is applicable for BS and ES (as required in RR2), existing service representation approaches are examined next (as required in RR6). In order to address RR2 approaches, which are focused on business and on technical aspects
are examined. Further, the examination of existing approaches will be focused on functional aspects only (as required in RR7). Existing BS representations are examined first. Service representations with a technical focus are examined afterwards. Moreover, there is one service representation approach that considers both perspectives equally, which will be examined lastly. At the end of this subchapter there is a conclusion about a conceptual service model.

### 3.3.1 Service representations business perspective

Service representations from business perspective refer to services in real world, such as hair cutting, plumbing or transportation. There are existing approaches for business services representation, which are examined in the following.

#### 3.3.1.1 Molecular Modeling approach

According to the Molecular Modeling approach “… services and products are intimately and symbiotically linked” [Shostack 1993] and form a larger entity. Shostack draws an analogy to atoms, which are connected as molecular configurations, and proposes molecular modeling, which addresses the coexistence of products and services in a model. Examples of basic molecular models are illustrated in Figure 3.5.

![Figure 3.5 Molecular modeling examples [Shostack 1993]](image)

Main elements of molecular modeling are service element, product element, service evidence, and bonds. Bonds connect all elements and state that there is a relationship between two elements. The service element represents a particular service. In turn, the product element represents a particular product. Service evidences are physical objects that cannot be categorized as real products. They are close connected to a service and play an important role for accomplishing the service. Shostack divides service evidences into peripheral evidence and essential evidences. Peripheral evidences are objects that are possessed by the customer, but do not have a value independent from the service. An admission ticket for an amusement park, for instance, is a typical peripheral evidence. Peripheral evidences are not considered in Figure 3.5. Instead, they are described by an external table next to the diagram. In contrast, essential evidences have dominant impact on the purchase of a service. Unlike peripheral evidences, essential evidences are not possessed by the customer. Because of their dominance, essential evidences
are modeled within the molecular model as quasi-products, i.e. with a dotted line instead of a solid one. As shown in Figure 3.5, a vehicle for a car rental service is a typical example for an essential evidence. An example that shows the coexistence of peripheral and essential evidences is flight service: The aircraft of a flight service is an essential evidence, whereas the flight ticket is a peripheral evidence. Furthermore, the fast food restaurant example shows the coexistence of service and products. Food is the product and preparation of the food is the service (cf. Figure 3.5). Moreover, there are different amounts of service and product elements in each molecular model diagram. The dominance of either the service elements or the product elements affects whether one considers this the model more likely as a service or as a product

[Shostack 1993].

As one of the earliest service modeling approaches Molecular Modeling represents a model for the coexistence of services and products. Evidences show resources that are engaged within a service. Elements are connected via bonds. However, the expressiveness of a bond seemed to be uncertain and inconsistent (e.g. considering Figure 3.5 the meaning of the bond between Ride and Game in the amusement park example is not clear). Furthermore, internals of a service are not addressed, which makes Molecular Modeling very abstract. A formal specification is missing.

Next, Service Blueprinting is considered, which addresses the modeling of a service in more detail.

### 3.3.1.2 Service Blueprinting

Service Blueprinting is related to Molecular Modeling and offers a way to model service elements. The objective is to provide a possibility for modeling all relevant aspects that appear during creation and management of a service. Service Blueprinting introduces processes for that purpose and refers to existing process description approaches, such as PERT [Project Management Institute 2008]. Service Blueprinting provides a processes model, including process steps, time flow, time scheduling, branches, merges, and possible fail points including the predicted effect of the scheduling, as well as the line-of-visibility to the customer. Further, there are tables that contain service functions, and tables that keep track about time tolerances and input materials [Shostack 1993]. Figure 3.6 shows a shoe cleaning example modeled using Service Blueprinting.

---

20 This makes it hard to label a fast food restaurant as a service or a product respectively [Shostack 1993].
Figure 3.6 Service Blueprinting Example

Advantages of Service Blueprinting are the visibility of service processes, the possibility to view deviation tolerances, fail points, and the benefit for the consumer. Similar to Molecular Modeling Service Blueprinting is one of the first approaches in the service modeling and pointed the way in this area. Accordingly Service Blueprinting is referenced a lot [Böttcher 2008]. Although Service Blueprinting provides a comprehensive service model, there is a lack of formalization, i.e. most of the explanations are examples, a modeling according to an abstract and concrete syntax is missing. Neither a meta-model nor a modeling-language is provided.

3.3.1.3 Structured Analysis and Design Technique (SADT)

[Congram/Epelman 1995] criticize Service Blueprinting and provide an own approach for the modeling of service processes – the Structured Analysis and Design Technique (SADT). SADT is focused on activities, which are constituted as basic elements of a service. Each activity has input, output, mechanisms and controls (cf. Figure 3.7). Input and output represent “what is being transformed into what” [Congram/Epelman 1995]. Mechanisms state “who” or “what performs the activity”. Controls state “what guides or limits the activity” [Congram/Epelman 1995]. Further, SADT allows a step-by-step refinement of activities, which in turn allows structuring processes hierarchically [Congram/Epelman 1995].
In contrast to Service Blueprinting, SADT provides a graphical modeling-language including a grammar. The concepts “line of visibility” as well as the modeling of fail-points and evidences/resources explicitly are missing in comparison to Service Blueprinting. However, the mentioned concepts input, output, mechanisms and controls are important concepts for service modeling [Böttcher 2008].

### 3.3.1.4 Service Concept

The idea behind Service Concept [Kaner/Karni 2007] is to describe constituents of a service. Service Concepts is based on a meta-model, which contains classes, objects, attributes, and values. Classes are functional homogeneity categories of objects of a system. They are subdivided in major classes, main classes, and minor classes. An object is constituted as a functional homogeneity component of a system. Objects contain attributes and values. An attribute is a symbolic or numeric descriptor of objects and has a symbolic or numeric value. Thus, objects are central. They are categorized by a three-level class hierarchy. Further, attributes and values describe each object in detail. Classes, attributes and values lead to a five level hierarchy. All concepts had been derived from an analysis of 19 research papers, which had been published between 1981 and 2006 [Karni/Kaner 2007].

Considering the five level hierarchy, on first level there are 9 major classes. Major classes are customers, goals, inputs, outputs, processes, human enabler, physical enabler, informatics enabler, and environment. (1) Customers are those who consume services, who are affected by services and who benefit from them. (2) Goals represent the meaning as well as the purpose of the service. (3) Inputs are entities (physical, human, or informational entities) that are processed by the service. (4) Outputs are entities (also physical, human, or informational entities) that are the results after having processed the service. (5) Processes are sequences of activities to be fulfilled for transforming inputs into outputs. (6) Human enablers are those who are operating and providing the service. (7) Physical enablers are aiding the operation of the service. (8) Informatics enablers are information and knowledge
resources that are needed for accomplishing the service. And (9) environment comprises all external factors that may influence the operation of a service. Further, on second and third level there are 75 main classes and 351 minor classes. On fourth level and fifth there are attributes a service contains of as well as possible values. Main classes, minor classes, attributes and possible values are not mentioned here. Instead, an example is given that illustrates the categorization and description of an object. For instance, an intangible object can be categorized into the major class: input, the main class: utilization factors; minor class: usage context, and can comprise the attribute: context in which product is used, as well as the values: work, school, home, and recreation [Kaner/Karni 2007]. Service Concept presents a comprehensive set of objects that are categorized and described. However, formal restrictions between objects and attributes are missing, there are no detailed semantics about objects, and process related aspects, such activity flow, fail over and time constraints. Anyway, the work of Kaner and Karni [Kaner/Karni 2007] provides a bunch of useful concepts that can be used as a foundation for service representation [Böttcher 2008].

3.3.1.5 Service Ontology / OBELIX

Akkermans et al. share the opinion of business service and electronic service as suggested in this thesis. However, [Akkermans et al. 2004] use the terms “real-world service” and “Web Services”. “Real-world services - that is, non-software-based services - differ significantly from Web Services, usually defined as software functionality accessible and configurable over the Web.” [Akkermans et al. 2004]. Within the project OBELIX (Ontology-Based Electronic Integration of Complex Products and Value Chains) Akkermans et al. built that is generic, component-based, and focused on real-world services. The service ontology distinguished three interrelated top-level viewpoints: (1) The service value, which represents a service description from customer’s perspective, (2) the service offering, which represents a service description from supplier’s perspective, and (3) the service process, which represents how the service offering is put into operation. Each viewpoint is described by an ontology, which is examined in the following.

The service value ontology (cf. Figure 3.8) contains the concepts customer, demands and sacrifices. A customer requires one or many demands and is willing to give one or many sacrifices. The demand is specified by quality of the service, cf. service quality. A sacrifice can be a price or it can be represented by direct-, indirect- or psychological relationship costs. All mentioned concepts and their relations are shown in Figure 3.8.
The service offering ontology (cf. Figure 3.9) represents the supplier side of a service. The central concept of service offering ontology is the service element. Service element represents what a supplier offers to its customers – “a business performance of a typically intangible nature” [Akkermans et al. 2004]. A Service element can be decomposed into smaller service elements and composed to larger entities. Larger service entities are called service bundles. A service element can have different roles represented by the function concept. Functions can be enhancing supplementary services, supporting supplementary service, bundle services, substitute a certain service or simply providing core functionality in terms of the main businesses. Further, the service element uses and results in resources, which represent main inputs or outcomes of a service. Last not least, service elements and resources contain properties, which are not further discussed in OBELIX. All mentioned concepts are depicted in the Figure 3.9.

Since OBELIX service ontology focuses on service value and service offering, the service process viewpoint is not further discussed in [Akkermans et al. 2004]. Instead, Akkermans et al. refer to existing approaches (such as ebXML [Gibb/Damodaran 2002], Web Service Flow Language [Leymann 2001], Business Process Execution Language [Andrews et al. 2003] and OWL-S [Martin et al. 2004]) which can be adopted for service process description.
Resources, which are key for service offering (and service process), are specified in more detail. Resources present main inputs and outcomes of a service. OBELIX service ontology typifies resources into physical goods (e.g. cargo), human resources (e.g. drivers), monetary resources (e.g. bill), information resources (e.g. shipment document), capabilities (e.g. special drivers license), experiences (e.g. in tour planning) and state changes (e.g. transportation from A to B).

Furthermore, since the service element is key, OBELIX service ontology provides a graphical for service element modeling. The basic notion of the service element is depicted in Figure 3.10. This notion is support by the e3value toolset [Gordijn/Akkermans 2001].

As illustrated in Figure 3.10 each service element has an input interface and an outcome interface. Each interface has one or many ports. Each port refers to one resource that is used as an input for the service or produces as an outcome of the service. Service elements can be coupled by connecting the outcome of one service with the input of another one.

The OBELIX service ontology provides a comprehensive work for service modeling. It comprises relevant service concepts based on found literature research. Further, the service ontology provides different viewpoints and perspectives for service modeling, which is important for emphasizing customer and operational concerns. Furthermore, it provides notion, visualization and tool support (i.e. e3value [Gordijn/Akkermans 2001,Gordijn et al. 2011]). Although the OBELIX service ontology does not address the process viewpoint directly, it provides a significant contribution in the area of service representation [Böttcher 2008].

Considering service representations from business perspective five approaches had been considered so far. However, the thesis does not claim to provide a full exploration of all service representation approaches. At least the most referenced ones are considered. Service representations for electronic service, i.e. from technical perspective, are examined next.
3.3.2 Service representations technical perspective

Electronic services represent the software infrastructure that supports the fulfillment of business services. It is assumed that each software product can be transformed into several electronic services. Since, the dominant realization of electronic service are Web Service, most of service representations rely on Web Service representations as well. However, there are models, which rescind from Web Service realization and provide a more abstract and general view. Starting with the most common service representation for Web Service - Web Service Description Language (WSDL) is considered first.

3.3.2.1 Web Service Description Language (WSDL)

The Web Service Description Language (WSDL) [Christensen et al. 2001] is an XML-based language that provides a model for describing Web services. As such WSDL is platform independent, i.e. it does not rely on a specific type of server nor an operating system, it is independent from the programming language, e.g. C++ [Stroustrup 1997], Java [Gosling 2000] or .Net C# [Albahari et al. 2002], and it is independent from its transportation protocol, i.e. it can be transferred using TCP [Stevens/Wright 1995], HTTP [Fielding et al. 1996] or any other protocol. Basically, WSDL describe the operation including input and output parameters of the Web service as well as the endpoint for accessing the Web service. In particular, the elements of the WSDL 2.0 specification [Chinnici et al. 2007] are divided into two sections - an abstract and a concrete section. The abstract section contains the definitions of types and interfaces, i.e. operations with input and output parameter. The concrete section contains binding information and service endpoint specifications. This allows reuse of abstract definitions. Elements of WSDL 2.0 are depicted in Figure 3.11.

![Figure 3.11 WSDL 2.0 specification](image)

The overall element is the description element, which contains types, interface, binding and a service element. Types describe the data types used within the description. Usually XML Schema [World Wide Web Consortium (W3C) 2009] is
used for this purpose. The interface element contains the operations of a Web service. An interface has one or many operations. Each operation contains input and output parameters as well as the message exchange pattern (cf. [World Wide Web Consortium (W3C) 2007b]). Input and output refer to the types defined in the types section. The binding element describes a concrete message format. As such, it refers to interface and operations. Further, the binding element defines the transportation protocol, e.g. HTTP [Fielding et al. 1996], or SOAP [World Wide Web Consortium (W3C) 2007a]. Last not least, the service element contains one or many endpoints. Endpoints define the concrete connection point to the Web service. As such it contains an address, i.e. usually a simple HTTP URL [Internet Engineering Task Force (IETF) 1994b], and refers to a particular binding.

WSDL provides definitions on syntactical level. A software program that wants to access the Web service gets information about the endpoints, messaging formats, protocols and communication patterns from WSDL. So, WSDL provides information that fosters machine-to-machine communication, i.e. software programs know how to interact with the Web service. However, there is no semantic information, i.e. WSDL does not provide any description about service capabilities. For instance, one cannot infer from a WSDL containing two numbers as an input and one number as an output whether the Web service provides a summation, subtraction, a currency translation, or any other functionality. Semantic information for Web service supports Web service discovery, Web service reuse, and may facilitate integration and composition of Web services [World Wide Web Consortium (W3C) 2005]. An approach that provides semantic information to WSDL is WSDL-S [World Wide Web Consortium (W3C) 2005]. WSDL-S is examined next.

### 3.3.2.2 WSDL-S

Since WSDL lacks in semantic expressivity, [World Wide Web Consortium (W3C) 2005] proposes Web service semantics. Web service semantics seeks in improving the reuse of Web service. The Web Service semantics document [World Wide Web Consortium (W3C) 2005] proposes WSDL-S as a mechanism that associates semantic annotations to Web Service Description Language (WSDL). WSDL-S is guided by the following principles:

- **WSDL-S should be build on existing Web service standards, i.e. WSDL [World Wide Web Consortium (W3C) 2007b].**
- **WSDL-S should not be tied to a particular semantic representation language, since there are a number of potential languages available, such as OWL [World Wide Web Consortium (W3C) 2004a], WSML [Keller et al. 2004], and UML [Object Management Group (OMG) 2011a].**
- **WSDL-S should support existing standard data types, such as XML Schema data types [World Wide Web Consortium (W3C) 2001].**
Following these principles WSDL-S proposes adding semantics to WSDL. WSDL-S takes advantage of the extensibility of WSDL. Annotations of the elements types, instances, operations, inputs, and outputs are used in order to refer to concepts of a semantic model. Beside the mentioned elements, WSDL-S adds two new ones, called precondition and effect. Preconditions define requirements that have to be true before an operation of a Web service can be successfully invoked. In contrast, effects represent statements that have to be true after the operation of a Web service completes execution. Both, preconditions and effects, are defined by a set of semantic expressions, which refer to a semantic model. The semantic model is used for preconditions and effects, and for connecting other element annotations as well. As stated in the principles above, the semantic model does not rely on a specific language, i.e. any arbitrary semantic language, such as OWL [World Wide Web Consortium (W3C) 2004a], WSML [Keller et al. 2004], or UML [Object Management Group (OMG) 2011a], may be used.

The coexistence of WSDL, WSDL extensions for WSDL-S and the semantic model is illustrated in Figure 3.12.

![Figure 3.12 Web Service semantics with WSDL-S (cf. [World Wide Web Consortium (W3C) 2005])](image)

To conclude, WSDL-S is a simple but pragmatic approach. Adding annotations to WSDL provides basic semantic descriptions to Web service. However, WSDL-S is focused on Web services and does not provide a semantic framework for electronic service descriptions. There are more sophisticated approaches, such as OWL-S [Martin et al. 2004] and WSMO [Roman et al. 2006b], which are considered next.
3.3.2.3 OWL-S

Based on the idea of Semantic Web [Berners-Lee et al. 2001] OWL-S [Martin et al. 2004] proposes the application of semantics for Web services by using OWL [Lacy 2005]. Derived from DAML-S the first version of OWL-S had been published in 2001 [Burstein et al. 2001]. This makes OWL-S the earliest approach for Semantic Web services. OWL-S is based on an upper-ontology, which serves as a conceptual model and defines basic elements. The root element of this ontology is the service element, which contains three sub-elements: ServiceProfile, Service-Model, and ServiceGrounding. The relation between these elements is illustrated in Figure 3.13.

![Figure 3.13 Main elements of OWL-S (cf. [Martin et al. 2004])]()

The ServiceProfile describes the intended purpose of the service, i.e. what the service provided by the service provider offers to the service requester. There are three basic ServiceProfile information types: information about the organization that provides the service, a functionality description of the service, and information about specific characteristics of the service (e.g. non-functional aspects). First, service provider information comprises contact information of the service provider. Second, the functionality of a service is described by a description of the service transformation process. Basically, the transformation description contains the state before and the state after service execution. In detail, the ServiceProfile contains the attributes input, output, precondition and effect (IOPE). Input represents required inputs that are needed to perform the service. Output represents the service outcome. Further, since services may require external conditions and may have the effect that these conditions change after execution, OWL-S introduces preconditions and effects. As an example, [Martin et al. 2004] describes a selling service, which “may require as a precondition a valid credit card and as input the credit card number and expiration date. As output it generates a receipt, and as the effect the card is charged.” [Martin et al. 2004]. Precondition and effects are described by logical axioms. However, the format of these axioms is not prescribed by OWL-S. Instead, OWL-S refers to existing languages, such as KIF [Genesereth 1998] or SWRL [Horrocks et al. 2004]. Third, a ServiceProfile contains specific characteristics of a service. As such, [Martin et al. 2004] mentions the categorization of a service using the UNSPSC classification system.
Further, there are service quality parameters, such as reliability, response time, security, etc. Since a service requester may want to check the quality of the service before execution, non-functional properties contain important information.

The ServiceModel informs a service requester how to use the service. In order to describe how the requester interoperates with the service, the service is described as a process. Similar to ServiceProfile, the functionality of a process is described by input, output, precondition, and effect (IOPE). Further, a process contains a participant parameter, which refers to participants that are involved within the process. OWL-S distinguishes between atomic, simple and composite processes. Atomic processes can be invoked and executed in a single step (from the requesters’ point of view). Simple processes cannot be invoked, but are viewed as executed in a single step as well. Simple processes are used as elements of abstraction that allows multiple views on the same process. Composite processes correspond to actions that require multi-steps. Thus, they can be decomposed into other subprocesses. In order to define a control flow inside a composite process OWL-S provides a set of constructs, such as sequence, split, if-then-else and more (cf. [Martin et al. 2004]).

ServiceGrounding works as a connector between an abstract description and an executable service. It provides a mapping from an abstract to a concrete service specification and details of how to access the service. As such, ServiceGrounding proposes, for instance, a mapping from an OWL-S atomic process to a WSDL operation, including the mapping of OWL-S input and output parameters to WSDL input and output message parts. However, unless a predefined mapping from OWL-S to WSDL, OWL-S does not dictate the mapping. Any other mapping for grounding OWL-S services is possible.

OWL-S provides a comprehensive framework for Semantic Web service description. Further, it introduces functionality description by using IOPE. Beside OWL-S, there is the Web Service Modeling Ontology (WSMO), which is examined next.

3.3.2.4 WSMO

WSMO [Roman et al. 2006b] has its origins in the Web Service Modeling Framework (WSMF) [Fensel 2002]. WSMF has been taken as a starting point and, after refinement and extension, turned into the formal ontology and language of WSMO. WSMO aims at being compliant to Web technologies and service-oriented computing. For this purpose, WSMO defines eight design principles:

- **Web compliance**: WSMO is compliant to Web technologies by adopting concepts like URI [Internet Engineering Task Force (IETF) 1994a], Namespaces, and by supporting XML and other W3C recommendations.
• **Ontology-based**: Ontologies had been identified as the enabling technology for the Semantic Web. Thus, ontologies are used as the main data model in WSMO, i.e. all resources and all data interchange are based on ontologies.

• **Strict Decoupling**: WSMO defines resources in isolation, i.e. each resource is independent from its possible usage and from other resources.

• **Centrality of mediation**: By connecting elements, mediation can be understood as a complementary concept to strict decoupling. Mediation solves interoperability issues between data, underlying ontologies, processes and protocol. Interoperability issues naturally arise in open and heterogenic environments. Solving these issues is essential for successful Web service deployment.

• **Ontological role separation**: Usually, users (i.e. clients) and Web services exists in two different environments. Thus, the user’s objectives could not directly be mapped to the service capabilities (e.g. a user may wish to book a rail trip according to his/her preferences for landscape, whereas the Web services will just cover the availability of trains). WSMO is aware of these differences by separating desires of clients and the capability of available services.

• **Description versus Implementation**: WSMO differentiates between the description and the implementation of Semantic Web service. Implementation deals with the execution of Semantic Web service and related technologies. Its counterpart (the description), should provide a sound but concise semantic description of Web services. Ideally, the description is based on a framework including appropriate formalisms. WSMO provides the description model based on ontologies, and aims at being complaint to existing and emerging technologies.

• **Execution semantics**: WSMO provides technical realizations by some references implementations like WSMX [Haller et al. 2005] in order to verify the specification and the description model.

• **Service versus Web service**: WSMO differentiates between a service and a Web services. WSMO authors refer to [Baida et al. 2004] and [Preist 2004] and state that a Web service is a computational entity and a service is the value provided by the invocation of the Web service.

WSMO is guided by these eight principles. Before explaining the main constructs of WSMO it should be noted that WSMO is based on the meta-object facility (MOF) specification [Object Management Group (OMG) 2005]. The four layers of MOF had been discussed already (cf. section 3.2.1.3). Referring to Figure 3.2 the four layers are adopted as follows.

• **M3**: The meta-meta-model layer addresses the description of the structure and the semantics of the meta-metadata. WSMO uses basic MOF constructs, such as classes, attributes, etc., on M2.

• **M2**: The meta-model layer contains the description that defines the structure and semantics of the metadata. WSMO constructs and elements, i.e. ontologies, goals, web services and mediators, as well as their attributes belong to this layer.

• **M1**: On model layer there is a model that has to be compliant to the M2 meta-model. An actual WSMO description based on meta-model constructs correspond to this layer.
M0: The information layer contains description of data. In WSMO this refers to a concrete Web service and the data that is exchanged.

Main WSMO elements are: ontologies, goals, Web services and mediators. Ontologies provide the terminology used by all other elements. More precisely, ontologies can be imported in goals and Web service. Mediators use ontologies in order to solve heterogeneity. Web services describe functional and non-functional aspects of Web services. Goals describe the objectives, i.e. problems that should be solved by a Web service. Goals are used for discovering a Web service that solves a certain problem. Mediators solve interoperability problems, i.e. mediators are used in order to solve heterogeneity between two ontologies, between two Web services, between two Goals and between a Goal and a Web service (cf. Mediator types in [Roman et al. 2006b]). Since, this chapter is concerned with service representation, Web service description based on WSMO is examined in detail. Ontologies are not examined any further, since ontologies had been discussed already (cf. 3.2.2). Mediators are not examined any further either, since they are not concerned with service representation. And Goals, which are the counterpart of Web services, are not further discussed, since they are very similar to Web services descriptions.

The WSMO conceptual model, i.e. the meta-model, is located on M2 and uses MOF constructs from M3 for element definition. Basic constructs are the class construct (and the sub-class construct used for generalization), the attribute constructs (used for the definition of class attributes), as well as the type construct (used for specification of attribute types) and the attribute multiplicity specifications. Hence, using MOF-style description the WSMO Web service element is shown in Listing 3.2.

```plaintext
class webService sub-class wsmoElement
    importsOntology type ontology
    usesMediator type [ooMediator, wwMediator]
    hasNonFunctionalProperties type nonFunctionalProperty
    hasCapability type capability multiplicity = single-valued
    hasInterface type interface
```

Listing 3.2 WSMO Web service element

As shown in the listing above, the Web service element is derived from wsmoElement, which contains basic annotation, such as creator, owner, date, description, language, version etc. The Web service element refers to one or many ontologies, to one or many mediators, contains non-functional properties, has a

---

21 In WSMO each attribute has its multiplicity set to multi-valued by default. Single-valued multiplicity is stated explicitly.

22 Basic MOF elements from M3 are emphasized bold.

23 Examples for non-functional properties are accuracy, financial, network-related QoS, performance, reliability, robustness, scalability, security, transactional, etc. However, WSMO does neither provide details about non-functional properties nor provide information how to describe these properties.
capability element, and may have several interface definitions. Being focused on functional aspects, the capability and the interface element are examined next.

According to WSMO, the functionality of a Web service is defined by its capability. The capability element is shown in Listing 3.3.

```java
class capability sub-class wsmoElement
  importsOntology type ontology
  usesMediator type [ooMediator, wgMediator]
  hasNonFunctionalProperties type nonFunctionalProperty
  hasSharedVariables type sharedVariables
  hasPrecondition type axiom
  hasAssumption type axiom
  hasPostcondition type axiom
  hasEffect type axiom

Listing 3.3 WSMO capability element
```

Similar to the webService element, each capability element is derived from wsmoElement, refers to an ontology, to a mediator and contains non-functional properties as well. However, a capability contains preconditions, assumptions, postconditions, and effects. Similar to OWL-S, WSMO describes the functionality of a Web service by state transformation. Unlike to OWL-S, WSMO distinguishes between the state of the information space of the Web service and the state of its environment (called the state of the world). Hence, precondition specifies the information space of the Web service before its execution and postcondition describes the information space of the Web service after the execution of the Web service. Further, assumptions describe the state of the world before the execution of the Web service, and effects describe the state of the world after the execution of the Web service. Preconditions, assumptions, postconditions, and effects are described by using axioms in terms logical expression [Roman et al. 2006a].

The interface element describes how the capability of a Web service can be fulfilled, i.e. how the functionality of the Web service can be achieved and used. WSMO proposes a twofold view on the operation of a Web service by providing a description of choreography and orchestration. Choreography provides information that enables communication with the Web service from the client perspective. Orchestration describes the internals of a Web service, i.e. how the service achieves its capability. Further considerations about choreography and orchestration are available [Roman et al. 2005]. WSMO interface element is shown below.

```java
class interface sub-class wsmoElement
  importsOntology type ontology
  usesMediator type ooMediator
  hasNonFunctionalProperties type nonFunctionalProperty
  hasChoreography type choreography
  hasOrchestration type orchestration

Listing 3.4 WSMO interface element
```
All in all, there are more service representation approaches that address technical concerns such as UDDI [Clement et al. 2004] for instance. However, the four approaches mentioned above are the most prominent once today.

### 3.3.3 Service representation business and technical perspective

Beside the service representation approaches mentioned above, which have a clear business or technical focus, there are approaches that claim considering both perspectives. One of these approaches is the Unified Service Description Language.

#### 3.3.3.1 Unified Service Description Language (USDL)

The Unified Service Description Language (USDL) proposes a unified language for describing services. The USDL was developed by SAP Research. Similar to LSEM, SAP Research shares the idea of business service and electronic services. They state that there are electronic services, such as Web Services, which are electronically consumed by Web users, and business service, which are consumed by mainstream industries like logistics, banking, public sector and manufacturing. Further, even if business services are delivered physically, trading these services can be facilitated by IT and, thus, it is necessary to represent business service using a formal language. Furthermore, since electronic and business service are interrelated, they have to be represented by an unified language covering both. Hence, USDL seeks in a holistic service description for business services and electronic services [Barros et al. 2011d].

USDL is based on MOF and uses UML constructs, such as packages, classes, attributes, etc. USDL started with dividing information about services into business, operational and technical information [Cardoso et al. 2010]. However, this distinction had been proven to be too coarse-grained for adequate structuring. So, USDL had been split into nine packages (called modules) containing one class each. Modules are depicted in Figure 3.14 and explained below.
The foundation module contains concepts that are common among other modules, such as naming and identification etc.

The service level module contains concepts that are concerned with guarantees regarding the quality of the services, which are requested and claimed from different participants.

The participant module contains concepts, which are related to actors that participate in the consumption, delivery and provisioning of a service, such as consumer, provider, etc.

The pricing module contains concepts about the pricing structure of a service.

The legal module contains concepts concerning general terms and conditions as well as license and copyright aspects.

The service module contains central service concepts, such as services, service bundles, and relations between.

The interaction module contains concepts for outlining interactions between a consumer and a service.

The functional module contains concepts that describe the functionality offered by a service, e.g. functions, parameters and faults.

The technical module contains concepts for describing how to get access to a service, e.g. interface and access protocols.

Being focused on functional aspects of services, the service module and the functional module are examined in more detail.

The service module [Barros et al. 2011b] is the center of USDL. The service module contains basic concepts about services, refers to concepts from the foundation module (e.g. description) and is used in other modules, such as pricing, legal, service level, functional, technical etc. Central concepts in the service module are: service, composite service, service bundle, service variant, and abstract service. The service concept constitutes a particular service that provides a distinct capability. Assembled or aggregated services that provide more complex and comprehensive capabilities are constituted as composite services. According to the class
diagram, all composite services are services as well. A service bundle provides aggregated capabilities as well. However, a service bundle is different from a composite service, since service bundle aggregations are not composed in a functional way. Instead, service bundles are grouped for the purpose of selling. According to [Barros et al. 2011b] “It is important to point out that the parts of a service bundle share no functional dependencies among each other. This means that, unlike with composite services, rendering the same set of individual services will achieve the same result.” Beside these there are service variants that represent combinations of service options offered as a pre-packaged version of that service. Further, abstract services are used to represent classes of services. As such the AbstractService concept works as a template, which is used to group services with same description properties. Beside the concepts mentioned above there are more concepts, such as dependency, dependency type, composition type, part, service nature, etc. Details are provided in [Barros et al. 2011b].

The functional module is dedicated to the description of what a consumer achieves when consuming the services, i.e. the service capabilities. USDL describes service capabilities through service functions. Functions are the building blocks of rendering a capability [Barros et al. 2011c]. Functions produce outcomes (e.g. created something, transform something, deliver something, or destroy something), and functions are performed by an actor, who usually operates on one or more resources. Usually, resources are consumed, produced or affect by a function. Furthermore, actors use resources as tools to perform an action. And in some cases, it is necessary to describe conditions that have to be fulfilled before an action can be started and some effect that is set after an action is completed. A function is represented by the Function concept in the USDL functional module class diagram. The Function concept contains one or more names, may have a textual description, may refers a set of sub-function, may contain a set of input parameters (that are required for performing the function), may contain a set of output parameter (that are produced by the function), may have a set of preconditions (which have to be satisfied before the function can be performed), may have a set of post-conditions (which hold after the action is performed), contains a set of affectedResources and utilizedResources (i.e. resources that transformed or utilizes as part of performing the function), and more. Detailed information about all attributes of the Function concept and further concepts, such as Fault, Parameter and FunctionalOption are presented in [Barros et al. 2011c].

All in all USDL provides a comprehensive meta-model based on MOF that contains various aspects of services. USDL had involved dozens of researchers, whose expertise from computer science, business economics, marketing, legal, etc., had formed the language. However, modeling services using USDL requires knowledge in several disciplines. Further, the number of modules, classes, and relations makes USDL complex and requires a steep learning curve. Considering complexity [SAP AG 2011a] states “…generic service description language – like USDL – acting as a ‘one size fits all’ for domains as diverse and complex as banking/financials, healthcare, manufacturing and supply chains, is difficult to use and therefore not sufficient. […] not all aspects of USDL apply to all domains. Rather,
USDL needs to be configured for the particular needs of applications where some concepts are removed or adapted while new and unforeseen ones are introduced.”

3.3.4 Conclusion from existing service representation approaches

The conclusion about existing service representation has two directions. First, the question of using modeling or ontologies notation for service representation is still unanswered. By taking a deeper look on the notation used within existing approaches this question will be answered in the next section, i.e. section 3.3.4.1. Second, there is the question about common concepts used in BS representation and ES representation (cf. RR1 and RR2). Thus, with a special focus on functional aspects, basic service concepts used within existing approaches are examined in section 3.3.4.2. Based on this examination a conceptual model for service representation is concluded in chapter 3.4.

3.3.4.1 Notation for service representation

This section will take a look on the service representation notations used within existing approaches. The notations are summarized and listed in Table 3.1.

<table>
<thead>
<tr>
<th>Service representation approach</th>
<th>Service representation notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Molecular Modeling [Shostack 1993]</td>
<td>graphical notation, no formal language</td>
</tr>
<tr>
<td>Service Blueprinting [Shostack 1993]</td>
<td>graphical notation, no formal language</td>
</tr>
<tr>
<td>SADT [Congram/Epelman 1995]</td>
<td>graphical modeling-language</td>
</tr>
<tr>
<td>Service concept [Kaner/Karni 2007]</td>
<td>tables and descriptions, no formal language</td>
</tr>
<tr>
<td>OBELIX [Akkermans et al. 2004]</td>
<td>based on ontology languages</td>
</tr>
<tr>
<td>WSMO [Roman et al. 2006b]</td>
<td>based on MOF [Object Management Group (OMG) 2005] and ontology languages</td>
</tr>
<tr>
<td>USDL [Cardoso et al. 2010]</td>
<td>based on MOF [Object Management Group (OMG) 2005] and EMOF</td>
</tr>
</tbody>
</table>

Table 3.1 Languages of existing service representation approaches

Table 3.1 reveals non-formal language, graphical languages, representations based on XML, on ontology languages, and on MOF. Non-formal and graphical languages contradict requirement RR3. They are not formal, not machine readable and cannot be used for further processing. XML language are machine readable,
but they are not conform to ontologies and do not contain sound semantics, which contradicts the requirement RR4 and RR5. Ontologies and MOF provide sound semantics as required in RR5 and can be used in conjunction, as shown in WSMO [Roman et al. 2006a]. Ontologies are conform to RR4 per se. Further, since models can be transformed into ontologies as testified in chapter 3.2.3, MOF is conform to ontologies as well (as required in RR4). However, in order to ensure that BS and ES rely on the same conceptual model (as required in RR1), MOF provides a meta-data architecture that guarantees that resulting models are conform to a defined conceptual model, i.e. the meta-model M2. In more detail, one can define a meta-model based on MOF language and create models from that meta-model, i.e. instances of the meta-model, that follow the restriction, constraints, and rules of meta-model. MOF is used in WSMO [Roman et al. 2006b] and USDL [Barros et al. 2011d]. This does not mean that ontology languages are inappropriate for conceptual modeling - not at all. OWL-S [Martin et al. 2004] proposes an upper-ontology used as a conceptual model. However, the meta-data architecture proposed by MOF is intended to be created for meta-modeling, is more mature and provides tools and techniques for that purpose as well. Thus MOF is favored for conceptual modeling and will be used for further considerations (cf. section 3.4.1). Next, basic service concepts are examined with the scope on functional aspects (as required by RR7).

3.3.4.2 Basic functional service concepts

Several service concepts had been mentioned with the introduction of service representation approaches. These concepts are examined here with a special focus on functional aspects (as required in RR7) in order to conclude about proper service concepts from existing approaches (as required in RR6). The service functionality is always concerned with actions carried out by the service and resources, which are involved. Furthermore, transition happens during service execution, i.e. resource have different states before and after a service had been carried out. Additionally, one can conclude about service functionalities from the service name, a textual service description, by a service classification, and from involved stakeholders. Thus, the following functional aspects have to be considered by examining existing approaches: general characteristics such as identification, description and classification of services and other service concepts), actions, resources (including their state transition), and stakeholders, which are involved. These functional aspects and their occurrence in existing service representation approaches are listed in Table 3.2 and Table 3.3. Functional aspects are listed in rows. Existing service representation approaches are separated by columns. Each cell contains a description of the occurrence of a particular functional aspect within a particular service representation approach. If a functional aspect does not occur within a service representation approach the cell is filled with a '-' character. Further, most of the cells contain an explanation how the functional aspects occur. For instance, resources occur in Service Blueprinting [Shostack 1993] "as material". However, some of the cells contain the words implicit and barely. Barely
means that these concepts are mentioned within a particular service representation approach, but occur in a very simple and reduced form. Implicit means that the functional aspect does not occur in a particular service representation approach directly, but can is substituted by another concept, e.g. "implicit by using types" for resources in WSDL-S [World Wide Web Consortium (W3C) 2005].

As shown in Table 3.2 and Table 3.3, Molecular Modeling [Shostack 1993], Service Blueprinting [Shostack 1993] and WSDL [Chinnici et al. 2007] are the most basic approaches for functional descriptions. The most comprehensive ones, i.e. the approaches that contain the most comprehensive functional service description aspects are Service Concept [Kaner/Karni 2007], OBELIX [Akkermans et al. 2004], OWL-S [Martin et al. 2004], WSMO [Roman et al. 2006b] and USDL [Barros et al. 2011d]. Considering general information, service concepts are usually identified by their name. A textual description is provided by 3 of the 10 approaches only (i.e. OWL-S [Martin et al. 2004], WSMO [Roman et al. 2006b] and USDL [Barros et al. 2011d]). And classifications, such as service, resource and action hierarchies, are provided by most of the approaches. Further, actions play an important role, even when they are termed differently, such as operations, activities, processes, steps and more. Resources are important as well. Even though, resources are mentioned directly by the service representation approaches from business perspective only. Approaches from technical perspective, such as WSDL-S [World Wide Web Consortium (W3C) 2005], OWL-S [Martin et al. 2004] and WSMO [Roman et al. 2006b], refer to types and ontology concepts only and, thus, address resources implicit only. The transition, i.e. the state change of resources during an action, is address by mentioning inputs and outputs and, even more sophisticated, by addressing precondition and effects (i.e. IOPE). Stakeholders of a service are addressed comprehensively by USDL [Barros et al. 2011d] only.

All in all, the table below shows the occurrence of almost all functional aspects in each of the service representation approaches. Even though, the occurrence is more or less sophisticated. This examination will support the creation of a proper conceptual model that address BS and ES at same time. This is proposed in the following chapter 3.4.
Molecular Modeling [Shostack 1993]  
Service Blueprinting [Shostack 1993]  
SADT [Congram/Epelman 1995]  
Service concept [Kaner/Karni 2007]  
OBELIX [Akkermans et al. 2004]

General information  
Identification  
Free text description  
Classification  
Actions  
Resources  
Stakeholders  
Transitions

<table>
<thead>
<tr>
<th>Table 3.2 Comparison of basic functional aspects and their occurrence in existing service representation approaches (1/2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>General information</td>
</tr>
<tr>
<td>Identification</td>
</tr>
<tr>
<td>Free text description</td>
</tr>
<tr>
<td>Classification</td>
</tr>
<tr>
<td>Actions</td>
</tr>
<tr>
<td>Resources</td>
</tr>
<tr>
<td>Stakeholders</td>
</tr>
<tr>
<td>Transitions</td>
</tr>
</tbody>
</table>

| Table 3.3 Comparison of basic functional aspects and their occurrence in existing service representation approaches (2/2) |
3.4  Conceptual model for service representation

After the examination of existing approaches the conceptual model is presented in this chapter. Before doing so, the first subchapter (chapter 3.4.1) shows how MOF is used for that purpose. Next, the language and concrete syntax for conceptual model representation is introduce in subchapter 3.4.2. Third, the conceptual model for BS and ES is presented in subchapter 3.4.3. Last not least, in subchapter 3.4.4 there is a verification that the conceptual model and resulting BS and ES representations are conform to the requirements stated at the beginning of chapter 3.

3.4.1  Conceptual model and service representations based on MOF

As figured out in section 3.3.4.1, MOF is applied for conceptual modeling and service representation. MOF had been introduced in 3.2.1.3 already. The metadata architecture comprises four layers. According to MOF the meta-meta-model layer (M3) is used for defining structures and semantics of meta-metadata. MOF itself defines constructs, such as class, sub-class, type etc., on this layer. These constructs are used at the meta-model layer (M2) for defining the conceptual model called Service Comparison Meta-Model (SCMM). Analog to WSMO [Roman et al. 2006b] basic MOF syntax is used. The definitions on M2 are used for the representation of a particular business services or a particular electronic services on model layer (M1). The representations on M1 are termed business service models (BSM) or electronic service models (ESM) respectively. Matching logic will conclude about similarities between BSM and ESM later on (cf. chapter 5). The information layer (M0) contains data and instances of a concrete BS instance or ES instance respectively. M0 is considered for the sake of completeness only. The information layer will not be considered any further. The thesis is focused on matching BS and ES representations at M1. The relation between SCMM, BSM and ESM according to MOF is illustrated in Figure 3.15.

![Figure 3.15 Relation between SCMM, BSM and ESM according to MOF](image-url)
Before considering the conceptual service model SCMM in detail, one needs a syntax for expression. The syntax used here is based on MOF. Details about the syntax are provided next.

### 3.4.2 Syntax for SCMM representation

In order to describe the SCMM a concrete syntax is needed. As mentioned before the SCMM relies on MOF. MOF provides a meta-meta-model including basic constructs and a syntax as well. Thus, MOF syntax is used for SCMM expression. This syntax is introduced here.

The most important construct of MOF is the class construct and the class generalization construct sub-class. A class is identified by its name. Sub-Class refers to the name of another class the particular class is a generalization of. Each class contains zero or many attributes, which are identified by their name. Further, each attribute has a type. A type can be a data type, an enumeration type, or a class type. A data type refers to a simple data type, such as XML Schema Datatypes [World Wide Web Consortium (W3C) 2001]. An enumeration type refers to an enumeration, which is a data type whose values are elements of a finite set. A class type refers to a class. Using a class type means that a particular attribute is typed as that class. Each attribute has a multiplicity-value, which defines how many instances of the attribute may occur. The multiplicity-value is defined by a lower-bound and an upper-bound value. Lower-bound specifies the minimum occurrence, which is represented by an integer value greater or equal zero (0). The upper-bound value specifies the maximum occurrence occurrences, which is an integer value greater or equal one (1). If the attribute may occur infinite times the upper-bound is set to unbound represented by a star character (*). Last not least, there are special types of classes, which are denoted as abstract. An abstract class, servers as a template and cannot be instantiated. Thus an abstract class is used solely for the purpose of inheritance\(^24\). These are the many constructs used for the definition of the SCMM. Although there are more constructs defined in MOF, this subset is sufficient for SCMM definition. Further details about MOF constructs can be taken from [Object Management Group (OMG) 2005].

Using the MOF constructs mentioned above the concrete syntax for SCMM definition is illustrated in Listing 3.5. All MOF constructs are emphasized bold. Optional constructs are defined in square bracket. Names and identifiers are in camel case [Binkley et al. 2009].

\begin{verbatim}
[abstract] class ClassName [sub-class SuperClassName]
  AttributeName type TypeName
  multiplicities=LowerBound,UpperBound
\end{verbatim}

\textit{Listing 3.5 SCMM syntax specification according to MOF}

\(^{24}\) The MOF uses abstract classes in the same way as many object oriented programming languages, such as Java [Gosling 2000], do.
As mentioned, an attribute type can be a data type, an enumeration type or a class type. Enumeration types are defined in Listing 3.6.

<table>
<thead>
<tr>
<th>enumeration</th>
<th>EnumerationName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Value1, Value2, ..., ValueN]</td>
</tr>
</tbody>
</table>

*Listing 3.6 SCMM enumeration syntax specification according to MOF*

Based on this syntax the SCMM is specified in the next section.

### 3.4.3 SCMM specification

The statement “There is no one correct way to model […] - there are always viable alternatives.” [Noy/McGuinness 2001], cited at the beginning of this chapter, holds true for conceptual modeling as well. There are various existing approaches, which had been examined already in section 3.3. However, it is hard to favor for one of them. Some of the approaches lack in formality and semantics (e.g. Molecular Modeling [Shostack 1993], Service Blueprinting [Shostack 1993], SADT [Congram/Epelman 1995], Service Concept [Kaner/Karni 2007], and WSDL [Chinnici et al. 2007]) as required in RR3 and RR5 (cf. 3.3.4.1). Some of the approaches are too complex (e.g. OWL-S [Martin et al. 2004], WSMO [Roman et al. 2006b], USDL [Cardoso et al. 2010]), which contradicts RR8. And others are applicable for business concerns (e.g. OBELIX [Akkermans et al. 2004]) or electronic concerns (e.g. WSDL-S [World Wide Web Consortium (W3C) 2005]) only, which contradicts RR2. However, a proper SCMM for BS and ES representation is needed.

This chapter presents a SCMM, which is derived from existing approaches and is based on the functional service concepts that had been figured out in 3.3.4.2. Further, the meta-model will be according to the requirements in 3.1. Anyway, it is important to note that the conceptual model provided here is just one of many “… viable alternatives” [Noy/McGuinness 2001]. So, instead of providing the one and only SCMM, this SCMM is one possible candidate that works for this application here. Other possibilities that are according to the requirements are feasible as well.

The specification of the SCMM is presented next.

As figured out in 3.3.4.1, general information is needed for each service concept. General information will be represented by the ServiceConcept class in SCMM. This class is similar to the wsmoElement used in WSMO [Roman et al. 2006b] or the foundation module used in USDL [Barros et al. 2011a]. The ServiceConcept class is inherited by all other SCMM classes in order to provide a general information to each class. This ensures a sound description (as required by RR5) for each BSM and ESM. The ServiceConcept contains the attributes hasName, hasLabel, and hasDescription. Name refers to the name of a particular concept. Description contains free text description about a concept. Further, since a concept may have different names, e.g. a particular resource can be termed as a Truck or a Lorry, but refer to the same concepts, the label attribute is needed. The label attribute con-
tains all alternative names, i.e. all synonyms, for a particular concept. The label attribute is multi-valued, i.e. zero or many labels are possible. The ServiceConcept including all attributes is shown in Listing 3.7.

**Listing 3.7 SCMM ServiceConcept class**

```xml
<xs:simpleType name="ServiceConcept">
  <xs:attribute name="hasName" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:attribute name="hasLabel" type="xs:string" minOccurs="1" maxOccurs="1"/>
  <xs:attribute name="hasDescription" type="xs:string" minOccurs="1" maxOccurs="1"/>
</xs:simpleType>
```

The Service class represents a particular service. The Service class is a sub-class of ServiceConcept. Hence, a service inherits all attributes from ServiceConcept. Further attributes are not provided in the Service class. The Service class is declared as abstract. Abstract means that the Service class cannot be instantiated, i.e. the later model will not contain a concept, which is typed as a service. Instead, the Service class works as super-class for the AbstractService class and the ConcreteService class. Each service has to be an AbstractService or ConcreteService (not both at the same time). The Service class is shown in Listing 3.8.

**Listing 3.8 SCMM Service class**

```xml
<xs:simpleType name="Service">
  <xs:attribute name="hasAbstractService" type="AbstractService" minOccurs="0" maxOccurs="unbounded"/>
  <xs:attribute name="hasConcreteService" type="ConcreteService" minOccurs="0" maxOccurs="unbounded"/>
</xs:simpleType>
```

The AbstractService class is shown in Listing 3.9. According to [Preist 2004] an abstract service represents “the capacity to perform something of value”. In contrast, a concrete service represents a particular provisioning of some value. Abstract and concrete services are interrelated, since concrete service is the realization of an abstract service. Further, each abstract service has a set of realizations in terms of concrete services. In case of the SCMM, the AbstractService class is represented by a name, a set of labels, and a description (inherited from the ServiceConcept class). However, an abstract service does not contain any actions, resources or stakeholder directly. Abstract services contain as set of concrete services as suggested by [Preist 2004]. In addition to [Preist 2004], an abstract service may contain one or many abstract services (cf. hasAbstractService in AbstractService class), which facilitates that AbstractServices can be structured as a hierarchy or taxonomy. For example, an AbstractService called “TransportMangement” might contain three sub abstract services “OrderManagement”, “PhysicalTransport”, and “FinanceManagement”. All of these abstract services exist for the purpose of structuring services. An abstract service will never perform an action or contain a resource directly. Concrete services are used for this purpose. These are examined next.

**Listing 3.9 SCMM AbstractService class**

```xml
<xs:simpleType name="AbstractService">
  <xs:attribute name="hasAbstractService" type="AbstractService" minOccurs="0" maxOccurs="unbounded"/>
  <xs:attribute name="hasConcreteService" type="ConcreteService" minOccurs="0" maxOccurs="unbounded"/>
</xs:simpleType>
```
A ConcreteService (cf. Listing 3.10), i.e. a “particular provisioning of value” [Preist 2004], is defined as an “actual or possible performance of a set of tasks that represents a coherent functionality […]” [Preist 2004]. Further, [Preist 2004] specifies the constituents of a concrete service, which are one or more concrete tasks, a set of message, as well as a service provider, and a requestor. Thus, a ConcreteService in SCMM contains actions, which are tasks, and stakeholders that comprise service provider as well as service requestor. Messages, as suggested by [Preist 2004], will be added in terms of resources later on.

```plaintext
class ConcreteService sub-class Service
  hasAction type Action
  hasStakeholder type Stakeholder
```

**Listing 3.10 SCMM ConcreteService class**

Action and Stakeholder classes are defined below. Furthermore, since the ConcreteService class inherits all attributes from a service, a concreted service contains a name, a set of labels and a description text.

An action (Listing 3.11) describes an activity, operation, task, function etc., i.e. there are various synonyms as figured out in 3.3.4.1. Each action inherits all attributes from the ServiceConcept class and, thus, is defined by a name, a set of labels, and a textual description. During the execution of an action there are several resources involved. These are represented by the hasInvolvedResources attributes.

```plaintext
class Action sub-class ServiceConcept
  hasInvolvedResource type InvolvedResource
```

**Listing 3.11 SCMM Action class**

The InvolvedResource class (cf. Listing 3.12) represents a resource that is involved in an action. This class has a reference to exactly one resource (cf. hasResource). Further, the InvolvedResource class contains an involvement type, and the state change of the resource.

```plaintext
class InvolvedResource
  hasInvolvementType type ResourceInvolvementType
  hasStateBegin type ResourceState multiplicity=1,1
  hasStateEnd type ResourceState multiplicity=1,1
  hasResource type Resource multiplicity=1,1
```

**Listing 3.12 SCMM InvolvedResource class**

The involvement type is specified by the enumeration ResourceInvolvementType (cf. Listing 3.13). Each resource may have zero or many involvement types. However, each involvement type occurs only once. Possible values are Input, Output, Created, Changed, and Execute. Input means that the resource serves as an input for the action. Output means that the resource is the outcome of an action. Created means that the resource will be created during the action. Changed means that the
resource is changed during execution of the action. Execute means that a resource executes the action. Execute can occur if the resource is a machine or human. The attribute hasInvolvementType is multi-valued, which means that an involved resource may have multiple involvements. For instance, a document resource that serves as an input, might be changed and given out (i.e. output) within one single action.

```
enumeration ResourceInvolvementType
   Input, Output, Created, Changed, Executed
```

Listing 3.13 SCMM ResourceInvolvementType enumeration

Each resource has different states during execution of an action. The state of a resource changes from the beginning to the end. Thus, the state is represented by two attributes – hasStateBegin and hasStateEnd. The former describes the state of the resource before execution. The latter describes the state of the resource after execution of the action. The resource state is defined by a ResourceState class. Possible states for a particular resource are defined within the Resource class. Thus, there is a restriction that defines that states can be referred as hasStateBegin and hasStateEnd only, if they are defined within the resource that is referred in the InvolvedResource class. States that are not defined for that resources are invalid. The InvolvedResource class including the reference to the resource and its attributes about state change and the involvement type of the resource addresses the transition aspect as figured out in 3.3.4.1. This relates to IOPE (i.e. input, output, precondition, and effect) used in other approaches, such as OWL-S [Martin et al. 2004] and WSMO [Roman et al. 2006b]. However, instead of using logical axioms as proposed by OWL-S [Martin et al. 2004] and WSMO [Roman et al. 2006b], which contradicts the simplification requirement (RR8), the SCMM provides a much simpler transition representation here. USDL [Barros et al. 2011d] proposes transition representation in a similar way.

Resources enable the fulfillment of actions. As such the Resource class is defined as follows (cf. Listing 3.14).

```
abstract class Resource
   sub-class ServiceConcept
      hasResourceState type ResourceState multiplicity=0,*
      hasOntologyConcept type OntologyConcept multiplicity=1,1
```

Listing 3.14 SCMM Resource class

As mentioned earlier, a resource has several states, which are represented by the ResourceState class. Each ResearchState inherits all attributes from ServiceConcept (cf. Listing 3.15). The attribute hasResourceState of the Resource class defines possible states a particular resource may have. Thus, the multiplicity of this attribute is defined as zero to many. Resource states for a particular action can be set within the InvolvedResource class as mentioned before.

```
class ResourceState
   sub-class ServiceConcept
```

Listing 3.15 SCMM ResourceState class
The Resource class is declared as abstract. This means that a later model won’t contain a pure Resource type. However, the Resource class is inherited by three other classes called HumanResource, MaterialResource, and ImmaterialResource, which can be instantiated (cf. Listing 3.16).

```java
class HumanResource sub-class Resource
class MaterialResource sub-class Resource
class ImmaterialResource sub-class Resource
```

Listing 3.16 SCMM classes HumanResource, MaterialResource and ImmaterialResource

Material resources are tangible resources such as goods, material or machines. Immaterial resources, are intangible and immaterial. They are also known as information resources, such as documents. Human resources are humans that execute or interact within an action. Considering a truck, a driver and a delivery sheet, for example, the truck is a material resource, the driver is a human resource and the delivery sheet is an immaterial resource. The distinction between different resources is according to OBELIX [Akkermans et al. 2004], Service concept [Kaner/Karni 2007] and USDL [Barros et al. 2011d] (cf. 3.3.4.1). In course of this thesis the distinction is important for similarities comparison. In more detail, two material resources are more similar than a material resource and an immaterial resource (cf. chapter 5).

Going back to Listing 3.14, a resource has an attribute called hasOntologyConcepts. Each Resource class refers to exactly one OntologyConcept. An OntologyConcept represents a concept according to an ontology language such as OWL [Lacy 2005]. Referring to ontology concepts within the model is analog to other existing approaches, such as WSDL-S [World Wide Web Consortium (W3C) 2005] and WSMO [Roman et al. 2006b]. The OntologyConcept class is shown in Listing 3.17.

```java
class OntologyConcept sub-class ServiceConcept
    hasSubOntologyConcept type OntologyConcept multiplicity=0, *
    hasObjectProperty type ObjectProperty multiplicity=0, *
    hasDataProperty type DataProperty multiplicity=0, *
```

Listing 3.17 SCMM OntologyConcept class

Similar to OWL [Lacy 2005] each Ontology may contain zero or many sub-concepts, i.e. hasSubOntologyConcepts, that are used for a taxonomic relation of concept. Further a OntologyConcept may have zero or many object properties and zero or many data properties, which are analog to the OWL [Lacy 2005] specification. The ObjectProperty class and DataProperty class are shown in Listing 3.18 and Listing 3.19 respectively.

```java
class ObjectProperty sub-class ServiceConcept
    hasLowerBound type xs:integer multiplicity=1, 1
    hasUpperBound type xs:integer multiplicity=1, 1
```

Listing 3.18 SCMM ObjectProperty class
hasType type OntologyConcept multiplicity=1,1

Listing 3.18 SCMM ObjectProperty class

class DataProperty sub-class ServiceConcept
    hasLowerBound type xs:integer multiplicity=1,1
    hasUpperBound type xs:integer multiplicity=1,1
    hasType type DataType multiplicity=1,1

Listing 3.19 SCMM DataProperty class

Both classes contain a hasUpperBound and a hasLowerBound attribute from type integer. Both are used for setting property cardinalities. Further, both classes have a hasType attribute. However, the hasType attribute of the ObjectProperty class refers to another OntologyConcept. Instead, the hasType attribute of the DataProperty class refers to a DataType enumeration. As shown in Listing 3.20, the DataType enumeration is analog to the most common XML Schema data types [World Wide Web Consortium (W3C) 2001], i.e. Boolean, DateTime, Decimal, Integer, and String.

enumeration DataType
    Boolean, DateTime, Decimal, Integer, String

Listing 3.20 SCMM DataType enumeration

Last not least, each ConcreteService (cf. Listing 3.10) contains an attribute called hasStakeholder, which refers to a Stakeholder class (cf. Listing 3.21).

class Stakeholder sub-class ServiceConcept
    hasStakeholderInvolvementType type StakeholderInvolvementType multiplicity=0,*

Listing 3.21 SCMM Stakeholder class

A stakeholder is a natural or legal person, which can affect or is affected by a service. The Stakeholder class is used for specifying these persons. This is analog to participants and stakeholders mentioned in other approaches, such as OBELIX [Akkermans et al. 2004] and USDL [Barros et al. 2011d] (cf. 3.3.4.1.). The Stakeholder class inherits all attributes from ServiceConcept. Further, a Stakeholder has one or many hasStakeholderInvolvementType attributes, which refer to the StakeholderInvolvementType enumeration (cf. Listing 3.22).

enumeration StakeholderInvolvementType
    Provider, Consumer, Initiator, Controller, Responsible

Listing 3.22 SCMM StakeholderInvolvementType enumeration

A stakeholder can be provider, consumer, initiator, controller, or a stakeholder can be responsible for a particular concrete service. Initiator means that the stakeholder triggers the action. Provider means that a particular stakeholder provides this service. Consumer means that a stakeholder is the consumer of a concrete service.
Controller means that the stakeholder controls, monitors and supervises actions of a service. Responsible means that the stakeholder is responsible and accountable for the service’ actions. StakeholderInvolvementType is used for modeling how particular stakeholders are involved within a concrete service. As shown in Listing 3.21, the involvement type is multi-valued. Thus, for instance, a stakeholder can be a controller of a service and responsible for that service at same time.

As mentioned at the beginning of this section, this SCMM is one possible candidate. The SCMM provide here neither claims to be the one and only meta-mode for services nor it claims to be complete. For instance, considering enumerations, there are various additional values for StakeholderInvolvementTypes and ResourceInvolvementTypes possible. Further, there are various other terms existing for actions, resources, stakeholder etc., which are feasible as well. Next, requirements, which had been set up at the beginning of this chapter, will be verified in order to testify that the SCMM is conform to them.

3.4.4 Requirements verification

Requirements for service representation that had been defined in 3.1. These requirements must be checked in order to verify that the SCMM proposed here is a proper and suitable for further consideration.

RR1: Service representations for BS and ES have to rely on a conceptual model.

This requirement is ensured by MOF. Since the SCMM relies on MOF-layer M2, one can be sure that resulting instantiations, i.e. BS representations (BSM) and ES representations (ESM) on MOF-layer M1, are conform to that meta-model (cf. 3.4.1).

RR2: A conceptual model has to be applicable for BS and ES

This is ensured by examining existing service representation approaches from business perspective and from technical perspective as well (cf. section 3.3). Further, common service concepts are derived from these approaches (cf. section 3.3.4.2) and taken as a basis for the SCMM specification. Thus, the SCMM contains service concepts which are applicable for BS and ES at same time.

RR3: Service representations have to be formal and machine-readable

Formality had been considered by the examination of service representations in general (cf. section 3.2) and by examining notations of existing approaches (cf. section 3.3.4.1). Based on these findings, an SCMM is proposed that is based on MOF syntax (cf. section 3.4.2), which makes SCMM formal and machine-readable. There are various software tools that support creating, reading and managing MOF models, such as EMF [Eclipse Foundation 2011b]. Further, since MOF models can be transformed into ontologies (cf. section 3.2.3) there are other formal representations possible too.
RR4: Service representations have to be conform with ontologies

As figured out in section 3.2.3, MOF models can be transformed into ontologies without losing information. Thus, SCMM, which is based on MOF, is conform to ontologies per se. Further, the SCMM contains elements from the ontology language OWL - in particular the classes OntologyConcept, ObjectProperty and DataProperty – which makes SCMM conform to ontologies additionally.

RR5: Service representations should contain sound semantics

Sound semantics had been defined as names, identifiers, textual descriptions, synonyms, hierarchies (i.e. taxonomies), and relations between concepts. All of these elements are considered in the SCMM. For instance, the ServiceConcept class, which is inherited by all other classes, contains hasName (i.e. name), hasLabel (i.e. synonyms) and hasDescription (i.e. a textual description), and thus provides sound semantics to all concepts.

RR7: Service representations for BS and ES have to consider functional aspects of services.

Common functional aspects had been considered in section 3.3.4.2. Since the SCMM specification is based on these aspects (cf. section 3.4.3), the functional scope of the SCMM is testified.

RR8: Service representation should be simple, not too complex and not too complicated.

The SCMM provides one conceptual model without any modules or sub-models. There are 15 classes, having not more than 4 attributes each, and 3 enumerations. All service concepts are distinct from each other and well-arranged. Essential service concepts are included, but not too many. Thus, the SCMM is considered as simple and not complex. Further a simple way for expression transitions, i.e. without any logical expression, is used.

All eight requirements are fulfilled and verified. This makes the SCMM proposed here a proper conceptual model for further considerations. It will be used for service extraction in the next chapter and for the application of service matching later on.

3.5 Summary

This chapter was dedicated to the representation of services. It started with the definition of eight distinct requirements that ensure proper service representation in course of this thesis. Afterwards, existing approaches had been examined in detail. First, a general examination about information representation, i.e. models vs. ontologies, had been fulfilled. As a result it has been testified that models and ontologies can be treated equally for information representation and thus for service representation as well. Second, ten existing service representation approaches
from business and technical area had been examined in detail. Basic service concepts had been derived from that leading to a service concept metal-model (SCMM), which is used for BS and ES representation. At the end, all eight requirements had been verified in order to ensure conformity of the SCMM.
4 Service modeling

After the definition of the SCMM in the previous chapter, the question arises: How to model information for business service model (BSM) and electronic service model (ESM)? This chapter is dedicated to this question. For the extraction of information for BSM and ESM according to the SCMM one has to examine existing business cases, i.e. business service, and existing software products, i.e. electronic service, in detail. Thus, methods, techniques and tools are need for extracting knowledge and modeling. Extracting knowledge into a machine-readable form relates to knowledge engineering [Studer et al. 1998], software engineering [Sommerville 2010] and ontology engineering [Gómez-Pérez et al. 2004]. However, this thesis will not examine all engineering related topics. Instead, ontology engineering is selected. Ontology engineering is appropriated, since service representation based on SCMM is focused on ontologies and sound terminological semantics.

This chapter starts with a general introduction to methodologies, methods, techniques, etc. in section 4.1. Next, existing approaches of ontology engineering methodologies are examined and evaluated in section 4.2. Further, a method for BSM and ESM creation is presented in section 4.3. The validity of that method is proven by a case study in section 4.4. Afterwards, a tool that supports service modeling is presented in section 4.5. Chapter 4 ends with a summary about service modeling in section 4.6.

4.1 Methodologies, methods, techniques, processes, activities, and tasks

The general usage of the term methodology is ambiguous. The term methodology is used synonymously with terms like method or technique. In order to prevent misunderstanding these terms have to be defined clearly [de Hoog 1997]. Furthermore, methodology, method, and technique have to be related to terms like processes, activities, and tasks. This section provides definitions to all terms in relation to ontology engineering.

The term methodology is used in software engineering and knowledge engineering. The term was adopted for ontology engineering by [Gómez-Pérez et al. 2004]. In software engineering methodology is defined as „A comprehensive, integrated series of techniques or methods creating a general systems theory of how a class of thought-intensive work ought to be performed.” [IEEE 1995a] Hence, each methodology contains a set of methods and techniques. A technique is defined as a “Technical and managerial procedures used to achieve a given objective” [IEEE 1995a] and a method is „A formal, well-documented approach for accomplishing a task, activity, or process step governed by decision rules to provide a description of the form or representation of the outputs.” [IEEE 1994] Thus, there is a relation
between methodologies, methods, and techniques. However, the relation is not stated clearly by these definitions yet.

De Hoog examines the relation between methodologies and methods in detail. He wrote “… methodologies are in the same domain, [but] they are not the same” [de Hoog 1997]. De Hoog states that methodologies contain knowledge about methods. In more detail, a methodology defines, which method (what) is performed, at which time (when) and which person (who) is involved. [Gómez-Pérez et al. 2004] define the relation between methods and techniques. A method is a general procedure and, in contrast, a technique is a specific application of a method [Gómez-Pérez et al. 2004]. For example, a method describes that a circle can be drawn by line, where all points of the line have the same distance to a single point called center. However, for drawing a circle there are several techniques possible, e.g. drawing the line using a glass, a pencil tied to a string, a circle template, a compass, etc. Further, the IEEE method definition mentioned above contains tasks, activities, and processes. A process is „An organized set of activities performed for a given purpose … “ [IEEE 1995c], an activity is a “A defined body of work to be performed, including its required input and output information.” [IEEE 1997] and a task refers to “The smallest unit of work subject to management accountability. A task is a well-defined work assignment for one or more project members. Related tasks are usually grouped to form activities.” [IEEE 1995b]. These definitions lead to Figure 4.1, showing the relations between these terms.

![Figure 4.1 Relation between methodology, method, technique, process, activity and task](image)

As illustrated in Figure 4.1, each methodology contains one or many methods and one or many techniques. A technique specifies the general procedure of a method. Each method has one or many processes, which in turn have one or many activities. An activity can be divided into sub-activities. Each activity may have zero to
many tasks. An activity may have zero tasks, since not all methodology define tasks for their activities. The following examination of ontology engineering methodologies is according to these definitions.

4.2 Ontology engineering methodologies

Ontology engineering refers to the set of activities that concern the development process of ontologies, the methodologies and methods for building ontologies, the ontology lifecycle, and the languages and tools that support these activities [Gómez-Pérez et al. 2004]. This section examines existing ontology engineering methodologies. Four methodologies are considered here, i.e. the methodology of Uschold and King [Uschold/King 1995], the methodology of Grüninger and Fox [Grüninger/Fox 1995], METHONTOLOGY [Fernández-López et al. 1997] and On-To-Knowledge [Sure/Studer 2002]. However, there is a special focus on METHONTOLOGY, since it is the most comprehensive methodology [Gómez-Pérez et al. 2004]. Subsequently, an appropriated method for modeling services, i.e. BSM and ESM, based on SCMM, will be derived from this examination in the following section. The examination of ontology engineering methodologies is focused on main method constituents, i.e. the activities (cf. 4.2.1), sequence, and lifecycle (cf. 4.2.2), and on conceptualization and formalization (cf. 4.2.3).

4.2.1 Activities

There are certain activities that have to be carried out in order to build an ontology. [Gómez-Pérez et al. 2004] suggest 18 activities. All activities are considered in METHONTOLOGY [Fernández-López et al. 1997]. Other methodologies do contain a portion of these activities as well. The activities are categorized in management related activities, development oriented activities, and support activities (cf. Figure 4.2).

Management activities are scheduling, control and quality assurance. The output of the scheduling activity is a project plan. The project plan comprises concrete tasks, their arrangements, as well as time and resources needed for completion. The control activity guarantees that the scheduled tasks are completed according to the project plan. Last not least, quality assurance assures that the quality of each output artifact is satisfactory.

Ontology development activities are subdivided in a pre-development, development and post-development. Pre-development starts with an environment study that reveals applications and platforms the ontology will be used in. Further, the pre-development contains a feasibility study that answers question like: Is it possible and suitable to build an ontology? Development contains the activities specification, conceptualization, formalization and implementation. The specification activity defines goals, reasons, potential end-users, and the intended use of the
ontology. Conceptualization structures domain knowledge in terms of concepts and relations. Formalization transforms the conceptual model into a formal model. Implementation transforms the formal model into a computable model by using ontology languages. During post-development the ontology is used by applications or other ontologies. Post-development is dedicated to maintenance in terms of updates and corrections.

Support activities comprise a series of activities that are performed at same time to the development-oriented activities. Support activities are knowledge acquisition, integration, merging, alignment, evaluation, documentation, and configuration management. As the name suggests, knowledge acquisition addresses the acquisition of knowledge from domain experts. This is done manually or through some kind of a (semi-)automatic process. Integration, merging, and alignment are interrelated. Integration is needed when (re)using an existing ontology during development. Merging means that concepts are transferred from one existing ontology into the ontology under development. In contrast, alignment means that concepts between the existing ontology and the ontology under development are just linked. The evaluation activities make a judgment about potential technical issues or inconsistencies of the ontology and the associated documentation and the software environment as well. Documentation activity records meta-data of the ontology development process. Last not least, the configuration management activity records all versions of the ontology in order to control changes.

This collection of activities gives a comprehensive overview about things that have to be carried out within an ontology development project. However, activi-

---

An examples for semi-automatic knowledge acquisition is given by [Kietz et al. 2000].
ties cannot be done in isolation. Usually, activities depend on each other and, thus, a sequence of activities has to be defined.

### 4.2.2 Sequence and lifecycle

Basically there are three different sequence types: sequential, incremental, and prototyping. Sequential means that the sequence flow is linear. A linear flow corresponds to the waterfall model in software engineering [Specker 2004]. Activities are carried out one after another. Going back is not designated in the waterfall model. In contrast, incremental [McCracken/Jackson 1982] means that activities are carried out several times. Regarding to ontology development, the ontology will be developed in versions. Once the whole development process is completed, it starts again. During each iteration cycle a new ontology version is build systematically. Prototyping [Kendall/Kendall 1995] means that the ontology grows as required. Adding or removing concepts from the ontology is possible to any time of the development process. The process does not have to be completed before adding/removing a concept. Since ontology development is not a straight forward procedure and changes happen naturally, incremental and prototyping are more suitable for ontology development than the waterfall model. METHONTOLOGY, for instance, proposes evolving prototypes, which is a combination of incremental and prototype development. As shown in Figure 4.3 the development phase of METHONTOLOGY contains back loops between specification, conceptualization, formalization, implementation, and maintenance.

Beside the sequence of development activities, METHONTOLOGY defines an ontology lifecycle, including further activities and stages through which the ontology has to move during lifetime (cf. Figure 4.3).

Each ontology development project starts with the scheduling activity. After that the specification activity starts. At same time the management activities (i.e. control and quality assurance) and the support activities (i.e. knowledge acquisition, integration, evaluation, documentation and configuration management) start as well. All management and support activities are performed in parallel to the development activities (i.e. specification, conceptualization, formalization, implementation and maintenance). However, the intensity of these activities is different. In more detail, knowledge acquisition starts with the ontology specification activity with increasing intensity until the conceptualization activity. After conceptualization the intensity decreases until the maintenance activity is finished. Evaluation follows the same pattern as knowledge acquisition. Instead, the integration activity starts with its highest intensity at the specification phase and decreases until the maintenance activity has been finished. All other support activities (i.e. documentation and configuration management) and all management activities (i.e. control and quality assurance) have the same intensity during whole the development process.
4.2.3 Conceptualization and formalization

Conceptualization and formalization means eliciting relevant domain knowledge and transform it into a formal language. Conceptualization and formalization are crucial in ontology development and, thus, get a special attention in ontology engineering. METHONTOLOGY illustrates conceptualization and formalization as a transformation process shown in Figure 4.4.

The transformation of knowledge of a certain domain into a conceptual model that describes this knowledge is represented by $T_1$ and termed as conceptualization. The transformation $T_2$ converts the conceptual model into a formalized one. $T_2$ is called formalization. Last not least, $T_3$ represents the transformation from a formal model into a computable model. $T_1$ is termed implementation. $T_1$ and $T_3$ are depicted by continues lines. The dotted line of $T_2$ implies that some information is lost in this step, i.e. the conceptual model is more expressive then the formal model.
METHONTOLOGY provides details about the conceptualization activity $T_1$ and defines 11 tasks for structuring knowledge (cf. Figure 4.5). All tasks are explained in the following.

**Task 1: Build glossary of terms.** First of all, the ontology engineer should build a glossary including all relevant terms of the domain. The terms should be typed as concepts, instances, attributes, or relations between concepts. Further, a natural language description of each term as well as its synonyms and acronyms are collected. METHONTOLOGY suggests using a table for collecting this information. It is important to note that at initial stage of conceptualization inconsistencies appear by nature. This is not a problem as long as the ontology engineer detects them and refines the ontology later on. For instance, it may happen that some terms appear in separation, but refer to the same concept. In this case one should define these terms as synonyms.

**Task 2: Build concept taxonomies.** When the glossary contains a substantial number of terms the ontology engineer should bring concepts into hierarchy, i.e. define concept taxonomy. [Uschold/Gruninger 1996] mentioned three strategies for that purpose: top-down, bottom-up and middle-out. The bottom-up strategy starts with specific concepts and defines abstract concepts later on. The abstract concepts are specified from the specific ones by generalization. However, [Uschold/Gruninger 1996] confess that the bottom-up strategy may lead to a high degree of generalization. As a result a common understanding of the ontology is on risk. The top-down strategy defines abstract concepts first and specific concepts afterwards. The ontology engineer decides about the level of detail. However, there is the risk of starting with arbitrary (maybe not useful) abstract concepts, who lead to an ontology that may not represent the domain properly. The middle-out strategy provides a suitable trade-off between bottom-up and top-down strategy. Middle-out starts with fundamental concepts of the domain of interest. Afterwards, more specific and more general concepts are added. The ontology grows meaningful at discretion of domain experts. According to [Uschold/Gruninger 1996] middle-out is more suitable than bottom-up and top-down, because starting from fundamental concepts and adding specialization and generalization as required results in a model that represents the domain better and gains higher acceptance by the domain experts. After having finished the concept taxonomy the ontology engineer should examine possible taxonomy errors, such as common instances in partition or loops in hierarchy.

**Task 3: Build ad hoc binary relations.** After finishing the taxonomy, METHONTOLOGY proposes building ad hoc binary relation diagrams. The objective of this task is to connect concepts within a one taxonomy and between different taxonomies. As an example one could build a relation between the concept “Passenger” and the concept “Flight” which is called “books” (i.e. “Passenger --books--> Flight”).

**Task 4: Build concept dictionary.** Once the glossary of terms, the concept taxonomy and ad hoc binary relation diagrams had been build the ontology engineer has to specify relations, properties and instances of concepts in a concept dictionary.
Basically a concept dictionary is a table including concept name, instance names, class attributes, instance attributes and relations as columns. Each row contains one concept and a list of instances, attributes and relations. Details about relations and attributes, i.e. class attributes, instance attributes and constants, are described in tasks five to seven. Instances are specified at the end (i.e. task eleven).

As indicated in Figure 4.5, there is no special order for carrying out the following four tasks, i.e. Task 5 to 8.

**Task 5: Define ad hoc binary relations.** This task had already been touched by binary relations diagrams and the concept dictionary, i.e. Task 1 and 3. However, here relations are defined in more detail. METHONTOLOGY suggests a table for collecting this information. This table has six columns, i.e. relation name, source concept, source cardinality (max), target concept, mathematical properties, and inverse relation, on top. The rows contain each relation once (identified by its name) and each cell filled with an appropriated value. For instance, the relation “sameFlightAs” has the source concept “Flight” and the target concept “Flight”. The source cardinality is “N”, since each flight may have several similar flights. The relations is “Symmetrical”, i.e. flight A is “sameFlightAs” B means that B is “sameFlightAs” A, and “Transitive”, which means that if flight A is “sameFlightAs” flight B and flight B is “sameFlightAs” flight C than A and C are the same flights. “Symmetrical” and “Transitive” are indicated by mathematical properties. An example for an inverse relation for the relation “arrivalPlace” is “isArrivalPlaceOf”, i.e. a “Package” concept has a relation “arrivalPlace” to the concept “Location” and “Location” “isArrivalPlace” of the “Package” at the same time.

**Task 6: Define instance attributes.** The goal of this task is to describe instance attributes, which are already mentioned in the concept dictionary, in detail. Instance attributes are those attributes that differ (or may differ) for each instance of a concept. For example, each location has a certain name. Thus “name” is an instance attribute of the concept “Location”. Further, instance attributes are inherited by sub-classes always. Similar to the content dictionary, METHONTOLOGY suggests using a table for this purpose. Each row of the table contains an instance attribute (represented by its name), the associated concept, the value type (e.g. string, integer, float, Date, etc.), a measurement unit (e.g. the measurement unit for the attribute “budget” is “Currency quantity”), the precision (e.g. precision of budget is usually 0.01), the range of the values (e.g. 0 to 100000), default values if exist and the cardinality (e.g. (0,1), (0,N), (1,1) or (1,N)). Within the table, each instance attribute may occur several times, since an instance attribute might be used in several concepts (e.g. “name” is an instance attribute for “Location” and a “Person” as well).

**Task 7: Define class attributes.** Unlike to instance attributes, which describe values, class attributes describe concepts. For instance, the concept “Five-Star-Hotel” has an attribute which is called “numberOfStars”. The value of “numberOfStars” is 5. Since, “Five-Star-Hotel” is a concept (not an instance) the attribute “numbe-
“rOfStars” is a class attribute\textsuperscript{26}. Similar to the tasks before, METHONTOLOGY suggest using a table. Each row of the table contains detailed description of a certain class attribute. The columns are attribute name, defined at concept, value type, measurement unit, precision, cardinality, and value. Similar to instance attribute, each class attribute may occur several times in the table. This is because one attribute might be used by several concepts (e.g. “Five-Star-Hotel” “numb-rOfStars” “5” and “Four-Star-Hotel” “numberOfStars” “4”, etc.).

Task 8: Define constants. Constants have already been mentioned in the glossary of terms. Here the ontology engineer defines details about these constants. This is done by a table as well. The table contains a row for each constant including the name of the constant, the value type, the value, and the measurement unit. An example of a constant is “maximumTransportVolumeOfaPlane” with the value type “integer”, the value “800” and the measurement unit “m$^3$”.

The next two tasks are concerned with the definition of axioms and rules. Axioms are defined as logical expressions, which have to be always true. Axioms are used as ontology constraints. Rules, instead, are used to infer about ontology knowledge in general. As indicated in the figure above, METHONTOLOGY proposes to define axioms and rules in parallel.

Task 9: Define formal axioms. The definition of a formal axiom means that the ontology engineer must identify necessary axioms and have to describe them precisely. METHONTOLOGY suggests specifying the following information for each axiom: axiom name, natural language description, logical expression, used concepts, used attributes, used ad hoc binary relations, and used variables. The crucial part of the description is the logical expression. Here, METHONTOLOGY proposes first order logic as an expression language.

Task 10: Define rules. Similar to task nine, the ontology engineer must identify necessary rules first and have to describe them precisely afterwards. Similar to the previous task, METHONTOLOGY suggests the following information for rules: rule name, natural language description, logical expression, as well as concepts, attributes, ad hoc binary relations, used variables used in the expression. For the rule expression, METHONTOLOGY proposes the template “if <condition> then <consequent>”. “<condition>” contains of a one or many conditions, which can be connected using “<and>” and “<or>” statements. “<consequent>” is a single atom, which represents the consequence if the condition is true.

Task 11: Define instances. After tasks nine and ten, the conceptual model of the ontology has been created. Last not least, the ontology engineer might define relevant instances. These instances should appear in the concept dictionary in the instances column. For each instance METHONTOLOGY suggests the definition of instance name, concept name, instance attribute, and values of attributes.

\textsuperscript{26} As shown by this example, the term „concept attribute“ would be more appropriate than „class attribute“. 

- 90 -
4.3 Method for BSM and ESM modeling

Providing a complete methodology for engineering services is beyond the scope of this thesis. However, it is necessary to examine activities and tasks that have to be carried out in order to create BSM and ESM. Thus, a method is presented here, but partly only. First of all, activities that are necessary for building a BSM/ESM are summarized and arranged in sequence (cf. section 4.3.1). Second, tasks for conceptualization are presented in detail (cf. section 4.3.2). Activities and tasks are derived from existing ontology engineering methodologies (cf. section 4.2).

4.3.1 Activities and sequence

Following [Gómez-Pérez et al. 2004] most of the 18 reference activities are suitable for BSM/ESM development as well (cf. Figure 4.3). Thus, it is recommended starting a service modeling project with a scheduling phase in which a project plan is created. Other management activities such control and quality assurance are meaningful during the whole development process, too. Support activities such as documentation of the project, configuration management of each BSM/ESM version, evaluation of the BSM/ESM and integration with existing BSM/ESM are necessary as well. An important support activity is knowledge acquisition. Certainly, the fulfillment of knowledge acquisition differs between BSM and ESM a lot, i.e. the knowledge for BS is gained from interviews with stakeholders and experts from business domain and, in contrast, knowledge for modeling ES can be derived from existing software products (e.g. documentation, manuals, graphical
user interface, etc.). However, knowledge acquisition is crucial for proper development. The development of a BSM and ESM should start with a specification phase that defines the application domain, requirements, the goal, some design guidelines, and accessible sources for knowledge extraction (e.g. books, journals, magazines, interviews, existing BSM/ESM etc.). Next, the conceptualization phase should be fulfilled in order to transform domain knowledge into a conceptual model. Tasks for conceptualization will be addressed in the next section (cf. 4.3.2). After conceptualization, one has to transform the conceptual model into a formal model in order to provide BSM/ESM in machine-readable form. The formal model has to be conform to the SCMM defined in chapter 3. Both post-development activities (i.e. usage and maintenance) do make sense for BSM/ESM as well. Last not least, the development sequence evolving prototypes can be adapted from METHONTOLOGY, too. All in all, generic activities from ontology engineering are suitable for BSM/ESM development as well. However, the fulfillment for BSM/ESM modeling is different.

### 4.3.2 Conceptualization

Since conceptualization is crucial and has a special attention in ontology engineering, conceptualization within BSM/ESM modeling is examined in detail as well. Basically, conceptualization is adapted from ontology engineering with regard to the SCMM concepts. There are nine tasks for conceptualization within BSM/ESM modeling. Each task is explained below. Tasks and their sequence are shown in Figure 4.6.

*Task 1: Define services.* First of all, one should list names of all services (i.e. business services in case of BSM or electronic services in case of ESM). Next, each service must be typed as an abstract or a concrete service. It is a concrete service if one can define concrete stakeholders, actions, and resources involved (cf. section 3.4.3). If one cannot define stakeholders, actions and resources the service is typed as an abstract service. Further, a natural language description and synonymous names are defined for each service. Similar to ontology engineering inconsistencies appear by nature (cf. section 4.2.3). This is not a problem as long as the service modeler detects and refines them. For instance, if two services are different, but share the same set of synonymous words (i.e. labels), the service modeler has to refine this issue.

*Task 2: Define service taxonomy.* After having defined all services a service modeler should bring services in taxonomic order, i.e. define sub- and super services. This is done by using top-down, bottom-up or middle-out strategy as suggested by [Uschold/Gruninger 1996]. However, a combination of these strategies has the advantage of a result that is not too specific or too general. During this task additional services may appear. Thus, a service modeler might repeat task 1 (as indicated by the back loop arrow from task 2 to task 1 in Figure 4.6).
Tasks 3 to 5 follow no special order, i.e. they can be carried out parallel or in sequence. However, before starting task 6, a service modeler has to complete all three tasks.

**Task 3: Define stakeholders.** Each concrete service has one or many stakeholders that are involved within the service. Stakeholders can be service providers, service consumers, a person who is responsible or someone who controls the service. The SCMM provides a set of predefined stakeholder involvement types (cf. section 3.4.3) for that purpose. Further, the definition comprises stakeholder name, alternative names and a natural language description. A stakeholder is always assigned to a particular concrete service. Each concrete service has zero or many stakeholders.

**Task 4: Define actions.** Actions, which are carried out during realization of a concrete service, are defined here. Similar to the definition of stakeholders a service modeler defines a name, alternative names, and a description for all actions of a particular concrete service. All actions have to be assigned to a concrete service. Each concrete service has one or many actions.

**Task 5: Define resources.** Resources are involved during the fulfillment of a service. Thus, service modelers should define these resources by specifying name, alternative names, a textual description and the resource type (i.e. material resource, human resource, or immaterial resource, cf. section 3.4.3). Further, the resource involvement type is specified by the service modeler. SCMM provides four predefined involvement types for that purpose: input, output, created, changed and executed (cf. section 3.4.3). Furthermore, the transformation of resources must be specified. Resource transformation is defined by specifying the state at the beginning and at the end of the fulfillment of the action. Resource states are connected with the resource as defined in the SCMM.

**Task 6: Define ontology concepts.** Each resource is further detailed by an ontology concept (cf. section 3.4.3). An ontology concept represents a concept of an ontology. Thus, tasks from conceptualization mentioned in section 4.2.3 are appropriated here. However, since there are no instances, rules and axioms involved, the tasks 6 to 9 show a subset of 4.2.3 only. First of all, ontology concepts are defined including their name, synonyms (alternative names) and a natural language description.

**Task 7: Define ontology concept taxonomy.** A service modeler defines sub- and super concepts. Once again a combination of top-down, bottom-up or middle-out strategy [Uschold/Gruninger 1996] leads to a better result. Further, since new concepts may appear, during taxonomy development there is a back loop from task 7 to task 6.

**Task 8: Define object properties.** Each OntologyConcept may have zero or many object properties. An object property has a name, alternative names, and a natural language description. Further, a service modeler has to define the type of the object property as well as cardinalities. The later can be defined by setting lower bound and upper bound, which represent minimum and maximum occurrence of
the property. Since, new ontology concepts may occur during object property definition there is a back loop from Task 8 to Task 6.

Task 9: Define data properties. Last not least, a service modeler defines data properties. In contrast to object properties, data properties refer to primitive data types. SCMM defines the data types Boolean, DateTime, Decimal, Integer, and String. Beside the data type, a data property has a name, one or more alternative names, a natural language description as well as a cardinality definition.

Figure 4.6 Conceptualization tasks for BSM/ESM modeling

The conceptualization tasks described above follow a top-down approach, starting with services definitions going to the definition of service concepts such as stakeholders, actions, resources, and specifying details about these concepts. However, each task has a back loop to one of the previous tasks, indicated by the dotted arrows in Figure 4.6. Thus, it is possible to jump to one of the tasks, whenever something has to be added. This is according to evolving prototypes as proposed by METHONTOLOGY (cf. section 4.2.3).

4.4 Case study conceptualization

For prove of application of the proposed conceptualization approach a case study is provided here. The objective of this study is to describe a SAP® TM order management service, i.e. an ESM. It is assumed that all management and support activities are carried out in parallel. Further, it is assumed that all pre-development activities had already been accomplished and all following activities (i.e. especially post-development activities) will be carried out afterwards (cf. section 4.3.1). For knowledge acquisition the book “Transportation Management with SAP®
“TM” [Lauterbach et al. 2009] and the software itself, i.e. the graphical user interface of SAP® TM, had been identified as a proper source already.

**Task 1: Define services.** First of all, the order management service is defined by a name, a list of alternative names and a natural language description. Alternative names and description can be found in [Lauterbach et al. 2009].

**Task 2: Define service taxonomy.** Next, sub- and super services of order management are defined. Using the middle out strategy and following the navigation of the SAP® TM software product there is a sub service called order receipt. Further there is a super service called transport management. Since, transport management and order management are too generic in order to assign distinct activities and resources both are categorized as abstract services. Order receipt is categorized as a concrete service, since there are actions that belong to that service directly, such as order entry, order completion, and order confirmation.

Task 1 and task 2 results in Table 4.1, where labels refers to a list of all names of a service, type refers to the categorization into abstract service (A) or concrete service and super ID refers to the ID of the super service.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transport-Management</td>
<td>transportation management</td>
<td>Transport management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Order-Management</td>
<td>order management</td>
<td>Order management and order receipt are generally the beginning of an operational process in transportation management.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Order-Receipt</td>
<td>order receipt, order inquiry, receive order</td>
<td>Order receipt comprises a series of key operations that can be performed in the acceptance of a transportation request.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

*Table 4.1 Service definitions and taxonomy*

**Task 3: Define stakeholder.** Each concrete service has one or many stakeholder. In case of order receipt stakeholders are the transportation service provider and the ordering party. A transportation service provider is the provider of the service, but he is responsible for the service and he controls the service as well. Instead, an ordering party is consumer of the service and may initiate the service. Stakeholder definition results in Table 4.2. Type refers the involvement type of the stakeholder, i.e. provider (P), consumer (C), initiator (I), controller (Ctrl), and initiator (I). The service ID is a reference to the service a stakeholder belongs to.

---

27 The IDs are used for connecting concepts. This is similar to primary key and foreign key definitions used in relational databases [Date/Warden 1985].
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TrantransportationServicePro-</td>
<td>transportation service</td>
<td>The order manager manages the process of order from order entry to order</td>
<td>P, R, Ctrl</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>vider</td>
<td>provider, order manager</td>
<td>confirmation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OrdeeringParty</td>
<td>client, ordering party</td>
<td>The client sends an order request and gets an order confirmation.</td>
<td>C, I</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.2 Stakeholder definition

Task 4: Define actions. Actions that will be carried out during order receipt are order entry, order completion, and order confirmation. The definition of names and the description of all actions results in Table 4.3. The service ID refers to the service an action is involved in.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OrderEntry</td>
<td>order entry</td>
<td>A shipment request is entered either by directly creating it or by copying a</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>reference business object. If the data on the time of input cannot be</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>fully entered, intermediate storage of the new business object is possible,</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>followed by later processing.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OrderCompletion</td>
<td>order processing, order</td>
<td>If the shipment request has not been completely and consistently created</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>completion</td>
<td>during the first entry, it can be completed in a second or further processing step.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OrderConfirmation</td>
<td>order confirmation</td>
<td>If the shipment request has been completed so far and planned or priced if</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>necessary, so that a confirmation can be sent to the ordering party, the</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>order confirmation step is performed.</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3 Action definition

Task 5: Define resources. The definition of resources and their involvement within particular actions is done by listing all resources first and assign each resource to an action afterwards. Further, the involvement type as well as the resource state transition is specified. The resource definition leads to Table 4.4. Each definition has a name, a resource type definition, i.e. immaterial resource (I), material resource (M) or human resource (H), is assigned to an action using the Action ID, has an involvement type, which is predefined by the SCMM, and defines the state transition of the resource by specifying start and end state. Possible resource states have to be defined previously. In case of order receipt there is one resource called shipment request, which is involved within the three actions order entry, order completion, and order confirmation. Shipment request is an immaterial resource. It is created during order entry and changed within the actions order completion and order confirmation. The state transformation is defined in Table 4.4 as well.
Task 6: Define ontology concepts. Each resource is further detailed by the ontology concept. Ontology concepts are defined including their name, synonyms (alternative names) and a natural language description. Initially there is the ontology concept of shipment request only, as shown in the first row of Table 4.5. However, during fulfillment of task 7 and task 8 new ontology concepts are added, e.g. business partner, address, etc.

### Table 4.4 Resource definition

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Completed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>Completed</td>
<td>Confirmed</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shipment-Request</td>
<td>shipment request, transport order, freight request</td>
<td>A shipment request is an agreement between a transportation service provider and a ordering party regarding the transportation of goods or transportation equipment from an issuing partner or location to a receiving partner or location according to the agreed conditions.</td>
</tr>
<tr>
<td>2</td>
<td>Business-Partner</td>
<td>business partner</td>
<td>The subnotes for the business partners, aside from the mandatory specifications Shipper, Consignee and Customer, allow further parties involved in the transportation operation to be defined. These may be, for example, predetermined service providers (e.g. if requested by the customer), agents, or invoice recipients.</td>
</tr>
<tr>
<td>10</td>
<td>Shipper</td>
<td>shipper, transportation service provider, transporting party, carrier, forwarder</td>
<td>The party that ships the good. Shipper is also known as a carrier or freight forwarder.</td>
</tr>
<tr>
<td>11</td>
<td>Consignee</td>
<td>consignee, addressee</td>
<td>The party or business partner that is the consignee or addressee of the good, item or freight.</td>
</tr>
<tr>
<td>12</td>
<td>Customer</td>
<td>customer, consigner, addresser</td>
<td>The party or business partner that is the customer, addressee or consigner of the good, item or freight.</td>
</tr>
<tr>
<td>20</td>
<td>Address</td>
<td>address</td>
<td>The address of a business partner.</td>
</tr>
</tbody>
</table>

### Table 4.5 OntologyConcept definition

- 97 -
Task 7: Define ontology concept taxonomy. Initially, there is only one ontology concept, i.e. shipment request, which has no taxonomic relations. However, during fulfillment of task 8 and task 6 several times new ontology concepts are added. For instance, there is the concept business partner, which has three sub concepts called shipper, consignee and customer (cf. Table 4.5).

Task 8: Define object properties. After having finished task 6 and task 7, properties are defined for each ontology concept. An object property is defined by the name (Name), alternative names (Label), and a natural language description (Description). Further, each property belongs to a concept (i.e. a Domain) and has a type (i.e. a Range). This is done by referring to other ontology concepts. Through the rang definition new ontology concepts may occur, which leads back to task 6. Last not least, cardinalities are defined by setting lower and upper bounds. In case of hasBusinessPartner, which is an object property of shipment request, there are 3 to infinity business partners. This is represented by the definition of "[3,-1]" in Table 4.6.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasBusiness-Partner</td>
<td>business partner</td>
<td>Comprises at least three business partners, such as shipper, consignee, and the customer.</td>
<td>[3,-1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>hasAddress</td>
<td>address</td>
<td>A business partner does always has an address.</td>
<td>[1,1]</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.6 ObjectProperty definition

Task 9: Define data properties. Table 4.7 shows five data properties, which are hasStreet, hasNumber, hasCity, hasZIP, and hasCountry. Each of these properties has a name definition, alternative names (i.e. labels), a description and a cardinality definition. This is similar to object properties. Further, each data property belongs to a ontology concept, which is defined by a referencing ID (i.e. Domain ID) of the ontology concept. In case of Table 4.7, all properties refer to the same concept, i.e. address. Last not least, the data type is defined. Here, all data properties have the data type String. However, other data types such as Boolean, DateTime, Decimal, and Integer are possible for other properties.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasStreet</td>
<td>street</td>
<td>The address street</td>
<td>[1,1]</td>
<td>20</td>
<td>String</td>
</tr>
<tr>
<td>2</td>
<td>hasNumber</td>
<td>street number, house number</td>
<td>The number of the house in the street.</td>
<td>20</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>20</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasZIP</td>
<td>zip, zip code, area code, zip</td>
<td>The zip code of the address.</td>
<td>[1,1]</td>
<td>20</td>
<td>String</td>
</tr>
</tbody>
</table>
This case study shows the application of the conceptualization approach (cf. section 4.3.2) for BSM and ESM modeling. Table 4.1 to Table 4.7 showed the results from knowledge acquisition. Knowledge had been acquired from the book “Transportation Management with SAP® TM” [Lauterbach et al. 2009] and from the graphical user interface of SAP® TM. However, due to lack of space, not all concepts could be presented here. Further, tables and ID references are not handy and may lead to confusion within a more complex scenario. The definition of many concepts would become unmanageable. Thus, a tool is needed that supports the collection of concepts. Further, a tool is needed that guarantees that all models (i.e. BSM or ESM) are conform to the meta-model (i.e. SCMM).

### 4.5 Tool support for service modeling

As suggested in the previous section a tool is needed that supports service modeling. An appropriated tool is provided by the Eclipse Modeling Framework (EMF). Next, EMF is introduced (cf. section 4.5.1). Further the application of EMF for meta-modeling (cf. section 4.5.2) and modeling (cf. section 4.5.3) is examined.

#### 4.5.1 Eclipse Modeling Framework (EMF)

Eclipse Modeling Framework (EMF) [Eclipse Foundation 2011b] is part of Eclipse open source community. Eclipse is focused on building an open development platform comprising frameworks, tools, and runtimes for building, deploying, and managing software. It is supervised and supported by the Eclipse Foundation, which is a non-profit, member supported corporation [Eclipse Foundation 2011a].

Eclipse Foundation hosts various development projects. However, there are four top-level projects: the Eclipse Project itself, the Modeling Project, the Tools Project and the Technology Project. The Eclipse Project supports the development of a framework for building integrated development environments (IDEs). The Modeling Project is the key project for providing model-based development within Eclipse. Core of the Modeling Project is EMF, which provides the basic framework. Other sub-projects build on top of EMF and provide capabilities such as graphical editor generation, model transformation, and more. The Tools Project is dedicated to the implementation of developing tools. Last not least, the Technology Project serves as a home for new or experimental works and provides researchers an opportunity to get involved in the evolution of Eclipse. Beside these projects there is a growing number of sub-projects [Steinberg et al. 2008].

### Table 4.7 DataProperties definition

<table>
<thead>
<tr>
<th>area code</th>
<th>hasCountry</th>
<th>country, country code</th>
<th>The country of the address</th>
<th>[1,1]</th>
<th>20</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
EMF has emerged to one of the key Eclipse technologies. Main capability of EMF is code generation based on model definitions. As such, EMF bridges the gap between a modeler and a programmer. Further, EMF makes reinforcements to the modeling theory that a great portion of coding can be automated by an appropriated tool. As such, EMF provides a runtime framework that allows the validation of models, model persistence, and model editors.

EMF conforms to MOF and can be used according to the meta-model architecture as introduced in section 3.2.1.3. However, EMF does not use MOF basic constructs for meta-modeling, but provides Ecore [Steinberg et al. 2008], which is a similar but less complex. Basically, Ecore defines four kernel elements.

![Ecore kernel elements](image)

As shown in Figure 4.7 there are EClass, EAttribute, EDataType and EReference.

- EClass models classes themselves. Each class is identified by a name and may have a number of attributes and references. In order to support inheritance, a class can refer to a number of other classes.
- EAttribute models attributes. Each attribute is identified by a name and does have a type.
- EDataType is used to represent simple types such as String, Integer, Boolean etc.
- EReference is used for modeling associations between classes. Each reference has a name and refers to a type, which is a class in turn. Further, there are cardinality definitions, i.e. lowerBound and upperBound definitions.

Beside these four basic elements there are more elements, which are explained in detail in [Steinberg et al. 2008]. Using these elements for service meta-modeling leads to an Ecore based SCMM, which is introduced in the following section (cf. 4.5.2). Further, EMF provides the capability of generating an editor from Ecore models, which leads to a BSM and ESM editor as introduced in section 4.5.3.
4.5.2 SCMM meta-modeling using Ecore

EMF includes a simple tree-based editor for modeling Ecore. Additionally, there is a graphical Ecore editor based on UML notation. Other, third-party tools are available as well (e.g. Topcased Ecore Editor [TOPCASED 2011], Omondo Eclipse ML [OMONDO 2011], etc.). SCMM meta-modeling using the tree-based Ecore editor leads to Figure 4.8.

![Figure 4.8 SCMM in EMF tree-based Ecore editor (partly)](image)

Figure 4.8 shows a root element, which has four EClasses below. First, there is the ServiceConcept class, which has three EAttributes. The attributes hasName, hasLabel, and hasDescription have the EDataType EString. Further, the attribute icon indicates the cardinality, i.e. hasName [1,1], hasLabel [0,1], and hasDescription [0,1]. The Service EClass is inherited from ServiceConcept, which is indicated by an arrow between Service and ServiceConcept. Service class and ServiceConcept class are defined abstract, which is not illustrated in Figure 4.8, but can be set by a properties window within the EMF Ecore editor. Next, there is the AbstractService EClass, which has two EReferences - hasAbstractService and hasConcreteService referring to the EClasses AbstractService and ConcreteService respectively. Figure 4.8 shows a subset of the SCMM only. The whole SCMM is depicted in Appendix A.

Beside the tree-based representation EMF uses XMI [Object Management Group (OMG) 2011b] as an internal language. XMI had been introduced in 3.2.1.2 already. The XMI representation of Figure 4.8 is shown in Listing 4.1. The XMI code of the SCMM is shown in Appendix B. Furthermore, there is an UML based representation of the SCMM shown in Appendix C.

```xml
...  
<eClassifiers type="EClass" name="ServiceConcept" abstract="true">  
  <eStructuralFeatures type="EAttribute" name="hasName" eType="EString" lowerBound="1"/>
  <eStructuralFeatures type="EAttribute" name="hasLabel" eType="EString"/>
  <eStructuralFeatures type="EAttribute" name="hasDescription" eType="EString"/>
...  
```
4.5.3 BSM and ESM modeling based on SCMM editor

After the definition of the SCMM using Ecore, EMF provides a mechanism for generating an editor out of the meta-model. The procedure is straightforward. Basically there are the following steps: Define an Ecore meta-model, specify a generator model28, generate java code, generate an EMF editor based on that java source code, run the editor and create a new SCMM modeling project. Details about editor generation are provided in [Vogel 2011] and [Steinberg et al. 2008].

As a result there is an editor for creating models based on the SCMM. Similar to the Ecore standard editor the SCMM editor is simple and tree based29. The application of the editor according to the case study mentioned in section 4.4 leads ESM representation, which is a more appropriated representation than provided by the tables shown previously. The tree view of the order management ESM is shown Figure 4.9. Due to lack of space the full editor representation is shown in Appendix D.

All in all, the SCMM editor based on EMF has some advantages. Most importantly, it ensures that all models created with the SCMM editor conform to the meta-model definition. Thus, each BSM or ESM created with the editor is valid according to the SCMM. Further, resulting models can be transformed into various other formats and can be represented in different notations. This allows a graphical representation, for instance, or a java source code representation. A java source code representation makes the model machine readable and programmable. This allows the application of programming logic, e.g. matching algorithms.

---

28 The generator model contains additional information for the code generation, e.g. the path information and control parameter.

29 However, EMF provides mechanisms for implementing other probably graphical editors as well.
4.6 Summary

This chapter was dedicated to the BSM/ESM modeling based on SCMM. Service modeling includes methods, techniques, and tools that support the modeling process. First of all, existing ontology engineering methods had been examined. Next, an appropriated method for BSM/ESM modeling had been derived from that with a special focus on conceptualization. Afterwards, the method had been applied in a case study. The objective of the case study was modeling an order management ESM by acquiring knowledge from SAP® TM. Conceptualization had been carried out successfully. However, the case study showed the necessity of tool support. Thus, EMF had been introduced as a tool that fits the requirements. The application of EMF including meta-modeling, editor generation and BSM/ESM modeling had been shown. Since EMF supports java source code generation, BSM/ESM are machine readable and can be used for further processing. This provides the foundation for applying matching algorithms on service models. Service matching is introduced in the following chapter.
5 Service matching

Business service and electronic services are described based on the SCMM so far, leading to several BSM and ESM. All BSM and ESM are based on the same conceptual model and the same syntactic rules. Hence, syntactic heterogeneity is avoided. Next step is measuring terminological/semantic heterogeneity in order to conclude about the matching level between a particular BSM and a particular ESM. Measuring semantic heterogeneity is carried out by applying algorithms that calculate semantic concept correspondence. This is examined in detail within this chapter. First of all, relevant terminology, such as comparison, similarity, alignment, etc., is defined. Next, existing matching algorithms are examined. Matching algorithms are divided into basic matching algorithms and matching strategies. The former measures terminological similarity on basic level. The latter is dedicated to combine basic matching algorithms and to aggregate matching results. At the end of this chapter an overall matching algorithm is presented that provides a solution for BSM and ESM matching. All techniques are defined formally and illustrated by examples from the logistics domain.

It should be noted that ontology matching algorithms [Euzenat/Shvaiko 2007], which are usually dedicated to measuring semantic heterogeneity between two ontologies, are adopted here. This is possible, since ontologies and service models are treated equally within this thesis (cf. chapter 3).

5.1 Terminology and definitions

Ontology Matching is a process with the objective to find correspondence between ontology entities (such as concepts, relations, attributes, etc.) of different ontologies [Euzenat/Shvaiko 2007]. Adopted to services, service matching is defined as a process with the objective to find correspondences between service representations, i.e. correspondence between service model entities of different service models. The service matching process is illustrated in Figure 5.1.

![Figure 5.1 Service matching process (cf. [Euzenat/Shvaiko 2007])](image)

The service matching process has service models ($sm_{BS}$ and $sm_{ES}$), parameters ($p$) and resources ($r$) as input, contains a matching algorithm, and produces an output.
that is called alignment (A). As such, the matching process can be formalized as a function (cf. Definition 5.1).

\[ A = f(sm_{BS}, sm_{ES}, p, r) \]

**Definition 5.1 Service matching function (cf. [Euzenat/Shvaiko 2007])**

Input and output variables are defined as follows.

- \( sm_{BS} \) and \( sm_{ES} \) \( \in SM_{SCMM} \), where \( SM_{SCMM} \) represents the set of all service models, i.e. all BSM and all ESM, which are based on the same SCMM. For service matching one particular BSM (\( sm_{BS} \)) and one particular ESM (\( sm_{ES} \)) is needed.
- \( p \) represents a set of necessary parameters such as weights and thresholds
- \( r \) are external resources, which may support the matching process. External resource can be top-level ontologies, domain ontologies, a dictionary, a thesaurus, etc.
- \( A \) is the alignment that represents the matching result

Alignments represent correspondences between ontology entities (cf. [Euzenat/Shvaiko 2007]). Adopted to service matching, alignments represent correspondences between service model entities. Thus, an alignment may have one to many correspondences. Further, since service models have several different entities one has to define matchable entities, i.e. entities that can be matched. A matchable entity \( e \) is an element of the set of all matchable entities \( E \). Since \( E \) depends on the service model, it is defined as a function \( E(sm) \). Matchable entities of a business service model are defined in Definition 5.2.

\[ \{e_{BS1}, e_{BS2}, \ldots, e_{BSn}\} = E_{BS}(sm_{BS}) \]

**Definition 5.2 Matchable business service model entities**

The matchable entities function for electronic service model depend on a particular business service model entity, since entities from the same service concept, e.g. AbstractService, ConcreteService, Stakeholder, Action, etc., can be matched only. Thus, the matchable electronic service model entity function has one additional parameter and is defined as shown in Definition 5.3.

\[ \{e_{ES1}, e_{ES2}, \ldots, e_{ESn}\} = E_{ES}(sm_{ES}, e_{BS}) \]

**Definition 5.3 Matchable electronic service model entities**

Alignments represent correspondences between service model entities. Thus an alignment contains a set of correspondences \( c \). In ontology matching, a correspondence is defined as an existing relation between two entities of two different ontologies \( o \) and \( o' \) (cf. [Euzenat/Shvaiko 2007]). Adopted to service matching, the correspondence is an existing relation between two service model entities of two service models \( sm_{BS} \) and \( sm_{ES} \). More formal the correspondence measure is a 5-tuple shown in Definition 5.4.

\[ c = <id, e_{BS}, e_{ES}, r, n > \]

**Definition 5.4 Service correspondence (cf. [Euzenat/Shvaiko 2007])**
The variables of the 5-tuple are explained below.

- id is a unique identifier of the correspondence
- \( e_{BS} \) is an element of \( E_{BS} \), and \( E_{BS} \) is a subset of \( sm_{BS} \), i.e. \( e_{BS} \in E_{BS}(sm_{BS}) \)
- \( e_{ES} \) is an element of \( E_{ES} \), and \( E_{ES} \) is a subset of \( sm_{es} \), i.e. \( e_{ES} \in E_{ES}(sm_{ES}, e_{BS}) \)
- \( r \) is a relation of \( \Theta \), i.e. \( r \in \Theta \)
- \( n \) is a confidence degree of \( \Xi \), i.e. \( n \in \Xi \)

The 5-tuple asserts that there is a relation \( r \) between the service model entities \( e_{BS} \) and \( e_{ES} \) with the confidence \( n \). \( r \) represents a relation from a set of relations \( \Theta \). The most common relations used by most of the matching algorithms is the equivalence relation (\( = \)). However, others such as more general (\( \geq \)), more specific (\( \leq \)), and disjointness (\( \perp \)) exist as well. Further, a confidence degree is assigned to a correspondence. The confidence degree \( n \) is a measure of trust in the fact that a particular relationship between two entities does exist. Usually the confidence degree \( n \in \Xi \) is expressed by an interval \([0, 1]\) where 0 means no confidence and 1 means full confidence. Equivalence relations (\( = \)) and confidence degree are discovered in combination usually. This is done by measuring the similarity between entities. Adopted from [Euzenat/Shvaiko 2007], similarity between service model entities is defined as follows (cf. Definition 5.5).

**Similarity** \( \sigma \) is a function from a pair of entities \( e \) and \( e' \) from two different service models \( sm \) and \( sm' \) to a real number from \( \mathbb{R} \) expressing the similarity between those entities such that:

\[
\forall e \in sm, \forall e' \in sm', \sigma(e, e') \geq 0 \text{ (positiveness)}
\]

\[
\forall e \in sm, \forall e' \in sm', \forall e'' \in sm'', \sigma(e, e) \geq \sigma(e', e'') \text{ (maximality)}
\]

\[
\forall e \in sm, \forall e' \in sm', \sigma(e, e') = \sigma(e', e) \text{ (symmetry)}
\]

**Definition 5.5 Similarity** (cf. [Euzenat/Shvaiko 2007])

As the counterpart of the similarity, i.e. a dissimilarity, is defined as follows.

**Dissimilarity** is a function \( \delta \) from a pair of entities \( e \) and \( e' \) from two different service models \( sm \) and \( sm' \) to a real number from \( \mathbb{R} \) expressing the dissimilarity between those entities such that:

\[
\forall e \in sm, \forall e' \in sm', \delta(e, e') \geq 0 \text{ (positiveness)}
\]

\[
\forall e \in sm, \delta(e, e) = 0 \text{ (minimality)}
\]

\[
\forall e \in sm, \forall e' \in sm', \delta(e, e') = \sigma(e', e) \text{ (symmetry)}
\]

**Definition 5.6 Dissimilarity** (cf. [Euzenat/Shvaiko 2007])

The definitions of dissimilarity can be further constraint by a distance measure:

**A distance is a dissimilarity function** \( \delta \) satisfying the definiteness and triangular inequality:

\[
\forall e \in sm, \forall e' \in sm', \delta(e, e') = 0 \text{ if and only if } e = e' \text{ (definiteness)}
\]
Definition 5.7 Distance (cf. [Euzenat/Shvaiko 2007])

If similarity of different kinds of entities should be compared, it is recommended to normalize the measures. This is done by reducing each value to the same scale in proportion to the space size. Thus a normalized (dis)similarity is defined as follows:

A (dis)similarity is normalized if all values are in the range of real number in the interval \([0,1]\). A normalized version of a similarity \(\sigma\) is denoted as \(\overline{\sigma}\), a normalized version of a dissimilarity \(\delta\) is denoted as \(\overline{\delta}\), respectively.

Definition 5.8 Normalized (dis)similarity (cf. [Euzenat/Shvaiko 2007])

A normalizes similarity \(\overline{\sigma}\) corresponds to a normalized dissimilarity \(\overline{\delta} = 1 - \overline{\sigma}\) and vice versa. Since the following (dis)similarity measures have to be comparable, all (dis)similarity measures will be normalized.

Matching, alignment, correspondence, similarity, dissimilarity, and distance constitute basic terms that will be used in the following. However, before starting the examination of concrete matching algorithms, one has to be clear about the matching level, i.e. on which level of detail matching algorithms will be applied. Thus, service matching levels are examined next.

5.2 Service matching levels

Service models are based on the SCMM, which includes distinct service concepts (e.g. AbstractService, ConcreteService, Action, Resource, etc.) and service concept properties (e.g. name, label, description, etc.). Concerning service matching the SCMM provides a hierarchy, which leads to four matching levels:

- **Overall service matching level** denotes the overall matching result between a BSM and an ESM. The overall matching result is expressed by the correspondence measure \(c_{SM}\), which includes the BSM, the ESM and a similarity value: \(c_{SM} = < id_{BS}, sm_{BS}, sm_{ES}, \sigma_{BS}\rangle\). For instance, \(c_{SM} = < id, BS1, ES1, 0.57 >\) means that the overall similarity between BS1 and ES1 is 0.57
- **Service concept matching** level denotes matching results between one BSM and one ESM aggregated on service concept level. A service concept matching result is expressed by the correspondence measure \(c_{SC}\), which is defined as \(c_{SC} = < id, sm_{BS}, sm_{ES}, sc, \sigma_{SC}\rangle, sc \in SC\) (where SC represents the set of all service concepts). For instance, \(c_{SC} = < id, BS1, ES1, AbstractService, 0.43 >\) means that the similarity between BS1 and ES1 on AbstractService level is 0.43 (where AbstractService \(\in\) {AbstractService, ConcreteService, Action, Resource, etc.}).
• **Service entity matching level** denotes a set of correspondences, one for each e\(_{BS}\) and e\(_{ES}\) combination, within a particular BS-ES combination, where e\(_{BS}\) and e\(_{ES}\) belong to the same service concept: \(c_{SE} = < id, e_{BS}, e_{ES}, sc >, e_{BS} \in E_{BS}(sm_{BS}), e_{ES} \in E_{ES}(sm_{ES}, sc(e_{BS})), \) where \(sc(e)\) returns the service concept for a particular entity. For instance, the AbstractService "Transport" from BSM and the AbstractService "TransportationManagement" from ESM with a similarity of 0.80 is expressed as \(c_{SE} = < id, \text{Transport}, \text{TransportationManagement}, 0.80 >\). Transport and TransportationManagement belong to the same service concept (i.e. AbstractService).

• **Service entity property matching level** denotes a set of correspondences, which are related to service concept property instances. There are many properties defined for each concepts, e.g. name, label, description etc. The instantiation of a service concept results in particular values for each property. For instance, the abstract business service transport has a name, which is "Transport", has labels, which are "transport" and "transportation", and has a description, which is "Transport or transportation is the movement of people or goods from one location to another." Thus, matching names of two abstract services (i.e. one from a BSM and one from a ESM) leads to a correspondence on service entity property level, for instance. This is defined as \(c_{SEP} = < id, e_{BS}, e_{ES}, sp_{SC}, a_{SEP} >, e_{BS} \in E_{BS}(sm_{BS}), e_{ES} \in E_{ES}(sm_{ES}, sc(e_{BS})), sp_{SC} \in SP_{SC} \) (where \(sp_{SC}\) defines the property of a particular service concept.)

All correspondence measures are combined as an alignment A, that contains one \(c_{SM}\), a set of \(c_{SM}\), i.e. \(c_{SM} \in \{c_{SC1}, c_{SC2}, c_{SC1}, \ldots, c_{SCn}\} = C_{SC}\), a set of \(c_{SE}\), i.e. \(\{c_{SE1}, c_{SE2}, c_{SE1}, \ldots, c_{SEPn}\} = C_{SE}\), and a \(c_{SEP}\), i.e. \(\{c_{SEP1}, c_{SEP2}, c_{SEP1}, \ldots, c_{SEPn}\} = C_{SEP}\). Hence, the alignment A is defined as follows.

\[
A = \{C_{SM}, C_{SC}, C_{SE}, C_{SEP}\}
\]

**Definition 5.9 Alignment including correspondences on different level**

After definitions matching algorithms are examined next. Basic matching algorithms are used for matching service entities and service entity properties, i.e. they will be used for matching on service entity level and service entity feature level.

### 5.3 Basic matching algorithms

Basic matching algorithms assess the similarity between service entities. Basic matching algorithms are adopted from basic ontology matching algorithms [Euzenat/Shvaiko 2007]. Basic means that these matching algorithms address one particular feature of an entity. These algorithms are classified into four categories: name-based techniques, structure-based techniques, and extensional techniques.
5.3.1 Name-based techniques

Name-based techniques consider identifiers of entities, such as names, labels, or description. Name-based techniques can be sub-divided into string-based techniques and language-based techniques. String-based techniques compare the structure of strings. Language-based techniques consider names as words in natural language. String-based and language-based techniques are examined here. However, before examining details some definitions about strings and words have to be considered.

5.3.1.1 Definitions

A string contains a sequence of characters over an alphabet, which defines the set of possible letters for the string. The following definitions apply.

$S$ represents the set of all possible strings of any length over an alphabet $L$. A single string $s$ is defined as $s \in S$. The length of a string $s$, i.e. the numbers of characters of $s$, is denoted as $|s|$. $s[i]$ for $i \in [1, |s|]$ stands for the character of $s$ in position $i$. Furthermore, $e$ denotes an empty string, i.e. $|e| = 0$, and $s + t$ with $s, t \in S$ denotes the concatenation of the string $s$ and $t$.

**Definition 5.10 String (cf. [Euzenat/Shvaiko 2007])**

Example: The string "transport" is made of the characters 't', 'r', 'a', 'n', 's', 'p', 'o', 'r' and 't' and has the length 9. "" is the empty string and "transport"[7], i.e. the character on the seventh position, is 'o'. Given two other strings " " and "order", the concatenation of the three strings results in "transport order", which has the length 15.

String $s$ is a substring of another string $t$, if there are two (possible empty) string $s'$ and $s''$ such that $s' + s + s'' = t$ (denoted as $s \in t$). If $s'$ and $s''$ are empty then $s$ and $t$ are equal. Equality is denoted as $s = t$. In this case $s \in t$ and $t \in s$ at same time. The number of occurrences of $s$ in $t$ is denoted as $s \# t$.

**Definition 5.11 Sub-string (cf. [Euzenat/Shvaiko 2007])**

Example: The string "transport order" has the string "sport" as a substring. Further it has two occurrences of the substring "or".

The main problem in comparing service entities on the basis of their names are the existence of synonyms, homonyms, different languages and syntactic variations.

- **Synonyms** are different words used to name the same thing. For instance, "transport" and "shipment" refer to the same thing in some context.
- **Homonyms** are words used to name different thing. For instance, the word "order" may refer to a "sequence" or to a "booking". Multiple meanings of a certain word is also known as polysemy.
- **Words in different languages** are different from their appearance, but may refer to the same thing as well. For example, the word "transport order" means
"Transportauftrag" in German. There are several obstacles in different languages, which cannot be solved by translating word by word. For instance, compound nouns are divided by a space character in English (e.g. "transport order"), but occurs as a single word in German (e.g. "Transportauftrag").

- **Syntactic variations** occur through abbreviations, the usage of different spellings, optional prefixes/suffixes, etc. In the area of logistics, for instance, abbreviations are used for the definition of Incoterms. The string "FCA" stands for the Incoterm "Free Carrier". So, both strings refer to the same concept differ in their spelling. Furthermore, "FCA" could also refer to "Full cost accounting", "Function cost analysis" or even to the "Franchise Council of Australia" as well.

Consequently, it is not possible to deduce with certainty that two entities are the same when both have the same name. And in turn, it is not possible to deduce with certainty that two entities are different if they have different names. However, the way in which things are named and labeled is very important in communication and names remain a good indication of the similarity of things [Euzenat/Shvaiko 2007].

There are many different techniques assessing the similarity of two entities and the strings that denote them. As mentioned, these techniques can be subdivided into two categories: methods for comparing terms considering the characters of the strings only (string-base techniques), and methods using some linguistic knowledge to interpret these strings (language-based methods). Methods from both categories will be examined next.

### 5.3.1.2 String-based techniques

String-based techniques take advantage of the structure of the string. Strings can be compared in many different ways. [Euzenat/Shvaiko 2007] sub-divides string-based methods into (i) string, substring or subsequence techniques, which are based on the comparison of actual letters, (ii) edit distance methods that measure the effort in transforming one string into the other one, (iii) statistical measures that basically weight words within strings in order to highlight more important words and to mask less important ones, and (iv) path comparison taking related strings into account. An important part of string-based techniques is string normalization, which is applied before actual string-based matching. Thus, syntactic normalization is examined first.

#### Syntactic normalization

Normalization can help to improve the result of a subsequent comparison. Normalization techniques are, for instance:

---

30 Incoterms are a series of pre-defined commercial terms, which are widely used in international commercial transactions. They are intended primarily to communicate the tasks, costs and risks associated with the transportation and delivery of goods. Usually Incoterms are abbreviated by three-letters. [ICC - International Chamber of Commerce - The world business organization]
• **Case normalization** converts each alphabetic character of the string into lower case. For example "Transport order" will be converted to "transport order".

• **Diacritics suppression** means that characters with diacritic signs are displaced. For example the French word for location "localité" becomes "localite", and the city "Montréal" becomes "Montreal".

• **Blank Normalization** means that all blank characters, such as carriage return, blank tabulation and the sequence of blank characters are transformed into a single blank character.

• **Link stripping** converts links characters between words, such as apostrophes, underscores, hyphen into a single character. For instance, "location-based handling" becomes "location based handling".

• **Digit suppression** removes all digits. For example, "truck1843" becomes just "truck".

• **Punctuation elimination** removes all punctuation signs. For instance, "F.Ca." becomes "FCa".

However, normalization operations have to be used carefully, since they might be language dependent (in particular diacritics suppression), they are order dependent (i.e. the application of two or more operations in different order leads to different results usually), they can result in losing meaningful information (for instance H2O leads to HO by applying digit suppression), and normalization reduces variations, but increases synonyms on the other hand as well (e.g. the French word "livré" becomes "livre", but livré means delivered and livre means book).

In course of this thesis, English language is considered only. Since diacritics do not exist in English, this kind of normalization can be neglected here. Furthermore, it is assumed that blank normalization, link stripping, digit suppression, case normalization and punctuation elimination is has been regarded during service modeling. Thus, normalization is important but not considered any further in course of this thesis.

### String, substring and subsequence techniques

One of the simplest techniques for measuring similarity is the string equality measurement. String equality is a similarity function $\sigma$ that returns 1 if two strings are identical and 0 if they are not.

$$\forall s, t \in S, \sigma(s, t) = \begin{cases} 1, & \text{if } s = t \\ 0, & \text{if } s \neq t \end{cases}$$

**Definition 5.12 String equality (cf. [Euzenat/Shvaiko 2007])**

As announced at the beginning, each techniques and algorithm is illustrated by an example from logistics. The terms illustrated in the table below are common in logistics business and in logistics software as well. Let’s assume that all terms arranged horizontally in the top row are used within a particular ES and all terms
arranged vertically in the left column are used with a particular BS. Applying string equality to this example, results in the following in Table 5.1.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>transport order</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>source address</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>destination address</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>pickup time</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>good</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.1 String equality example

String equality identifies if strings are the same or not. However, it does not explain how strings are different. A more elaborated way for explaining how strings are different is the Hamming distance [Hamming 1950]. The Hamming distance counts the number of positions in which two strings differ from each other. Normalized by the length of the longest string, the Hamming distance is defined as follows:

\[
\forall s, t \in S, \delta(s, t) = \frac{\sum_{i=1}^{\min(|s|,|t|)} s[i] \neq t[i] + \text{abs}(|s| - |t|)}{\max(|s|,|t|)}
\]

Definition 5.13 Hamming distance (cf. [Hamming 1950])

Applying the terms of Table 5.1 for hamming distance results in Table 5.2. All results are transformed into a similarity.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.00</td>
<td>0.11</td>
<td>0.00</td>
<td>0.08</td>
<td>0.00</td>
<td>0.22</td>
</tr>
<tr>
<td>transport order</td>
<td>0.60</td>
<td>0.06</td>
<td>0.06</td>
<td>0.00</td>
<td>0.13</td>
<td>0.00</td>
<td>0.13</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.13</td>
<td>0.00</td>
<td>0.07</td>
<td>0.07</td>
<td>0.00</td>
<td>0.07</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.05</td>
<td>0.11</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.09</td>
<td>0.13</td>
<td>0.00</td>
<td>0.00</td>
<td>0.54</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.11</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.18</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 5.2 Hamming distance dissimilarity (as a similarity) example
String equality leads to a substring test when taking substrings into account. Substring test is a similarity function (measured between two strings) that returns 1 if one string is a substring of the other one (or the other way around) and returns 0 if there is no substring match. More formal:

$$\forall s, t \in S, \sigma(s, t) = \begin{cases} 
1, & \text{if there exist } p, q \in S \text{ where } s = p + t + q \text{ or } t = p + s + q \\
0, & \text{otherwise}
\end{cases}$$

Definition 5.14 Substring test (cf. [Euzenat/Shvaiko 2007])

An example of substring test is given below.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>transport order</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>source address</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>destination address</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>good</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.3 Substring test example

However this measure can be refined by a substring similarity that measures the ratio of the common subpart between two strings. So, let $x$ be the longest common substring of $s$ and $t$, the substring similarity is defined as:

$$\forall s, t \in S, \sigma(s, t) = \frac{2|x|}{|s| + |t|}$$

Definition 5.15 Substring similarity (cf. [Euzenat/Shvaiko 2007])

An example of substring similarity is given below.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.08</td>
<td>0.12</td>
<td>0.22</td>
<td>0.09</td>
<td>0.15</td>
<td>0.28</td>
</tr>
<tr>
<td>transport order</td>
<td>0.75</td>
<td>0.13</td>
<td>0.17</td>
<td>0.17</td>
<td>0.14</td>
<td>0.11</td>
<td>0.20</td>
</tr>
<tr>
<td>source address</td>
<td>0.09</td>
<td>0.13</td>
<td>0.09</td>
<td>0.07</td>
<td>0.07</td>
<td>0.11</td>
<td>0.11</td>
</tr>
</tbody>
</table>
The n-gram similarity [Kondrak 2005] is another possibility for comparing strings. The similarity function compares the number of sequences of n-characters. The n-character sequences are called n-grams. For example the 3-gram of the string "transport" is "tra", "ran", "ans", "nsp", "spo", "por", and "ort". More formal, n-gram(s, n) is a set substrings of s of length n. With this the n-gram similarity is defined as the intersection of n-grams of s and t.

\[
\forall s, t \in S, \sigma (s, t) = | \text{ngram}(s, n) \cap \text{ngram}(t, n) |
\]

**Definition 5.16 n-gram similarity (cf. [Kondrak 2005])**

The normalized n-gram similarity, i.e. normalized as an interval between 0 and 1, is defined as:

\[
\forall s, t \in S, \sigma (s, t) = \frac{| \text{ngram}(s, n) \cap \text{ngram}(t, n) |}{\min (|s|, |t|) - n + 1}
\]

**Definition 5.17 Normalized n-gram similarity (cf. [Kondrak 2005])**

An example of n-gram similarity is not provided here. However, the results are similar to the substring similarity results.

**Edit distance**

Edit distance measures are designed for measuring similarities between two strings that may contain spelling mistakes. Basically, the edit distance is the minimal cost of operations to be applied to one string in order to obtain the other one. More formal edit distance is defined as follows:

*Given a set of string operations \( op \) and a cost function \( w_{op} \) subjected to the operations, such that for any pair of strings exists a sequence of operations that transform one string \( s \) into the other one string \( t \) (and vice versa), the edit distance is a dissimilarity \( \delta(s, t) \) calculating the cost of the less costly sequence of operations for the transformation of \( s \) into \( t \).*

\[
\forall s, t \in S, \delta (s, t) = \min_{op_1(...op_i(s))=t} \left( \sum w_{op_i} \right)
\]

**Definition 5.18 Edit distance similarity (cf. [Euzenat/Shvaiko 2007])**

Operations that are usually considered in string edit distance measurement are the insertion of a character \( \text{ins}(c, i) \), i.e. insert character \( c \) at position \( i \), the deletion of a character \( \text{del}(c, i) \), i.e. delete character \( c \) at position \( i \), and the replacement of a character \( \text{sub}(c, c', i) \), i.e. replace character \( c \) by character \( c' \) at position \( i \). Each operation has certain costs assigned. The sum of all costs of operations needed to
transform one string into another one is the cost of one operation set. However, the minimum of all operation set costs is the edit distance measured between two strings.

Representative implementations of the edit distance are, for instance, the Levenshtein distance [Levenshtein 1966], the Needleman-Wunsch distance [Needleman/Wunsch 1970], the Smith-Waterman measure [Smith/Waterman 1981], the edit distance measure by Gotoh [Gotoh 1981] and the Monge-Elkan distance function [Monge/Elkan 1997]. The Levenshtein distance uses the operation insertion, deletion and substitution and weights all of them equal (i.e. the cost of 1). The Needleman-Wunsch distance instead weights the cost for the operations insertion and deletion higher than for the operation substitution. Other measures compute costs for each operation by a distinct function depended by the character or the substring that is affected by the operation. In that way, Smith-Waterman, for instance, uses a cost matrix for each operation. As an example, the table below shows the Levenshtein distance measure normalized by the maximum distance and transformed to a similarity in order to be comparable to the measures mentioned above. An example of the normalized Levenshtein distance represented as a similarity is given in the table below.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>Truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.18</td>
<td>0.53</td>
<td>0.53</td>
<td>0.35</td>
<td>0.47</td>
<td>0.59</td>
</tr>
<tr>
<td>transport order</td>
<td>0.65</td>
<td>0.18</td>
<td>0.41</td>
<td>0.29</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>source address</td>
<td>0.29</td>
<td>0.35</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>destination address</td>
<td>0.12</td>
<td>0.18</td>
<td>0.12</td>
<td>0.12</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.41</td>
<td>0.24</td>
<td>0.59</td>
<td>0.47</td>
<td>0.71</td>
<td>0.53</td>
<td>0.41</td>
</tr>
<tr>
<td>good</td>
<td>0.53</td>
<td>0.06</td>
<td>0.59</td>
<td>0.53</td>
<td>0.35</td>
<td>0.76</td>
<td>0.70</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.53</td>
<td>0.11</td>
<td>0.47</td>
<td>0.35</td>
<td>0.47</td>
<td>0.35</td>
<td>0.65</td>
</tr>
</tbody>
</table>

*Table 5.5 Normalized Levenshtein distance (as a similarity) example*

Another measure, which is not based on the pure edit distance, but follows the philosophy transforming one string into another is provided by the Jaro-Winkler [Winkler 1999]. The Jaro-Winkler measure is based on the Jaro measure [Jaro 1976, Jaro 1989], which basically counts number and proximity of common characters between two strings. The Jaro measure is a non symmetric measure and defined in the following.

\[
\forall s, t \in S, \sigma(s, t) = \frac{1}{3} \left( \frac{m}{|s|} + \frac{m}{|t|} + \frac{m - t}{m} \right)
\]
Where $m$ is the number of matching characters and $t$ is the number of transpositions. However, two characters are considered for matching only if their position within the string is not farther than ...

$$\left(\frac{\max(|s|,|t|)}{2}\right) - 1$$

... from each other.

Definition 5.19 Jaro measure (cf. [Jaro 1976, Jaro 1989])

For instance the strings "carte" and "trace" have the same five characters. However, $m$ is three only ($m=3$), because ‘c’ and ‘t’ do not count, since there are three positions between them, which is too far from $1.5 (=5/2-1)$. Moreover, $t$ as the number of transpositions for characters is one ($t=1$), since ‘e’ is already on the right position and ‘a’ and ‘r’ have to be transposed. Thus, the overall Jaro measure is for "carte" and "trace" is $0.62$ ($\frac{1}{3}*(\frac{3}{5}+\frac{3}{5}+\frac{3-1}{3}) = 0.62$).

The Jaro-Winkler measure is an improved version of the Jaro measure, taking into account, the Jaro measure $\sigma_{jaro}(s,t)$ itself, the length of the longest common prefix $l$ (e.g. the longest common prefix between "Martha" and "Marhta" is three, $l=3$) and a constant scaling factor $p$, which is 0.1 by default and should not exceed 0.25. With this the Jaro-Winkler measure is defined as follows.

$$\forall s, t \in S, \sigma(s, t) = \sigma_{jaro}(s, t) + l \cdot p \cdot (1 - \sigma_{jaro}(s, t))$$

Definition 5.20 Jaro-Winkler measure (cf. [Winkler 1999])

The comparison matrix for the Jaro-Winkler measure is shown below.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.49</td>
<td>0.36</td>
<td>0.00</td>
<td>0.45</td>
<td>0.63</td>
<td>0.92</td>
</tr>
<tr>
<td>transport order</td>
<td>0.92</td>
<td>0.51</td>
<td>0.30</td>
<td>0.00</td>
<td>0.44</td>
<td>0.61</td>
<td></td>
</tr>
<tr>
<td>source address</td>
<td>0.49</td>
<td>0.59</td>
<td>0.36</td>
<td>0.57</td>
<td>0.88</td>
<td>0.45</td>
<td>0.49</td>
</tr>
<tr>
<td>destination address</td>
<td>0.55</td>
<td>0.55</td>
<td>0.51</td>
<td>0.46</td>
<td>0.00</td>
<td>0.37</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.47</td>
<td>0.47</td>
<td>0.69</td>
<td>0.42</td>
<td>0.88</td>
<td>0.45</td>
<td>0.51</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.46</td>
<td>0.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.2</td>
<td>0.44</td>
<td>0.54</td>
<td>0.47</td>
<td>0.61</td>
<td>0.45</td>
<td>0.89</td>
</tr>
</tbody>
</table>

Table 5.6 Jaro-Winkler measure example

**Token-based distance**

Token-based techniques consider a string as a (multi)set of words. A set of words - also called “bag of word” - means that a string may contain several words possibly several times. Token-based approaches have their origin in information re-
trieval and do usually work well on long strings with many words. In case of service entities it is helpful to take words into account that are attached to the entity. This can be done (1) by aggregating different sources of strings such as labels, identifiers, descriptions, comments, or documentation, and (2) by splitting strings into independent tokens. In the first case, the service-comparison-meta-model defines a name, labels, and a description for each entity. For splitting strings into tokens compound names and label may become independent (but related) tokens. For example the bag of words "transport order" is split into the tokens "transport" and "order".

There are many ways of formalizing multisets in order to measure the similarity or the distance. However, it is common to consider a multiset $s$ as a vector $\vec{s}$, which belongs to a metric space $V$. Within this vector $s$ each token is represented by a single dimension. Further, the dimension value is represented by the number of occurrences of that token [Salton/McGill 1986]. For example, considering the two string $s$ "pickup time" and $t$ "pickup period", the metric space $V$ contains the three dimension "pickup", "time" and "period", i.e. one dimension for each token. Considering $V$, the occurrence of each token is 1, 1, 0 for $s$ and 1, 0, 1 for $t$. So, vectors $\vec{s}$ and $\vec{t}$ are:

$$\vec{s} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{t} = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

After the transformation of strings into metric vectors, metric distance measurements can be applied in order to calculate the distance between vectors. Common vector-based distance measures are the Euclidean distance [Deza/Deza 2009], the Manhattan distance [Krause 1986] and the Minkowski distance [Deza/Deza 2009]. However, a measure that is often used in information retrieval is the cosine similarity [Tan et al. 2006b]. The cosine similarity calculates the cosine of the angle made by two vectors. A cosine near 1 means that the angle is very acute and, thus, that both vectors (i.e. both multisets) are close related. On the other hand, a cosine near 0 indicates a very obtuse angle and, thus, that both vectors (i.e. both multisets) are not close related. The cosine similarity is measured by the following formula:

*Given two string $s$ and $t$ represented by the two vectors $\vec{s}$ and $\vec{t}$ in a vector space $V$, the cosine similarity is a function $\sigma_v$ such that:*

$$\sigma_v(s, t) = \frac{\vec{s} \cdot \vec{t}}{|\vec{s}| \cdot |\vec{t}|} = \frac{\sum_{i=1}^{n} (s_i \times t_i)}{\sqrt{\sum_{i=1}^{n} (s_i)^2} \times \sum_{i=1}^{n} (t_i)^2}$$

*Definition 5.21 Cosine similarity (cf. [Tan et al. 2006b])*

For example, given the two strings "pickup time" and "pickup period" and the two corresponding vectors $\vec{s}$ and $\vec{t}$ mentioned above. The scalar multiplication of $\vec{s}$ and $\vec{t}$ is 1 (i.e. $\vec{s} \cdot \vec{t} = 1 \times 1 + 1 \times 0 + 0 \times 1 = 1$). Further, $|\vec{s}| \cdot |\vec{t}| = \sqrt{2} \times \sqrt{2} = 2$. 

- 117 -
Thus $\sigma_0(s, t)$ results in a similarity measure of 0.5 ($= 1/2$). The application of the cosine similarity to the logistics terms results the following table.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>transport order</td>
<td>0.71</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.71</td>
</tr>
</tbody>
</table>

*Table 5.7 Cosine similarity example*

Another, token-based technique is based on path comparison. Path comparison takes not only the labels, but the sequence of labels into account. As an example, the concrete service "TransportOrder" belongs to a service hierarchy with an abstract parent element "OrderManagement", which has in turn a parent abstract service "Transportation". This can be illustrated as a path "Transportation::OrderManagement::TransportOrder". Other services can be "TransportManagement::OrderManagement::OrderReceipt" and "CustomerRelationshipManagement::CustomerEntry". Path comparison is defined as follows.

Given two sequences of strings $(s_i)_{i=1}^n$ and $(s'_i)_{i=1}^m$, the path distance is defined as:

$$
\delta \left((s_i)_{i=1}^n, (s'_i)_{i=1}^m\right) = \lambda * \delta'(s_m, s'_m) + (1 - \lambda) * \delta \left((s_i)_{i=1}^{n-1}, (s'_i)_{i=1}^{m-1}\right)
$$

with $\delta'$ as a specific string or language-based distance and $\lambda \in [0 1]$ as a penalty weight value.

*Definition 5.22 Path comparison*

For example, taking the Substring dissimilarity as the distance $\delta'$ and 0.7 as the penalty weight value $\lambda$ the comparison of "Transportation::TransportOrder" and "TransportManagement::OrderManagement::OrderReceipt" leads to the result 0.47 ($=0.7*0.615+(1-0.7)*(0.7*0.0+(1-0.7)*(0.438))$) and the comparison of "Transportation::OrderManagement::TransportOrder" and "CustomerRelationshipManagement::CustomerEntry" results in 0.76 ($=0.7*0.852+(1-0.7)*(0.556)$). Thus, the first and the second path are closer than the first and the third path.

*Summary on string-based techniques*

String-based techniques are useful when service models have similar strings in order to denote same entities. For instance, string-based techniques will typically
find similarities between strings like "transport" and "sea transport". In contrast, such algorithms will fail finding similarities between "transport order" and "shipment request", i.e. synonyms with different structure will yield a low similarity. On the other hand, selecting pairs with high similarities, may lead to false positives, since two strings can be similar, e.g. 'transport and 'sport', but denote different concepts. Thus, service matching must use more elaborated techniques and more reliable sources of information as well, which will be examined in the following. Nevertheless, the way in which things are named and labeled is very important in communication and names remain a good indication of the similarity of things [Euzenat/Shvaiko 2007].

5.3.1.3 Language-based methods

String-based methods consider a string as a sequence of characters. Instead, in language-based methods strings become text, i.e. a sequence of words. In contrast to string-based methods, where bag of words are considered as well, which contain words without any order, words follow a distinct sequence in a text.

Language-based methods rely on Natural Language Processing (NLP) that helps in order to extract meaningful words from a text. Comparing these words and their relationships may help to assess the similarity of ontology entities. Language-based methods can be divided into (i) intrinsic methods, which refer to linguistic normalization, and (ii) extrinsic methods, which make use of external resources such as dictionaries. Intrinsic methods, i.e. normalization, are considered first.

Intrinsic methods

Linguistic normalization aims at transforming terms in a way that they can be easily recognized. [Maynard/Ananiadou 1999] distinguishes three different kinds of term variations: morphological, syntactic, and semantic. In morphological variations form and function of the word are based on the same root. Syntactic variations means that words are added, removed or permuted. Semantic variation means that terms are replaced by a slightly different meaning. Table 5.8 shows six variation examples.

<table>
<thead>
<tr>
<th>Type</th>
<th>Subtype</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Morphological</td>
<td>Inflection</td>
<td>transport orders</td>
</tr>
<tr>
<td></td>
<td>Derivation</td>
<td>transportation order</td>
</tr>
<tr>
<td>Syntactic</td>
<td>Insertion</td>
<td>requested transport order</td>
</tr>
<tr>
<td></td>
<td>Permutation</td>
<td>order of transport</td>
</tr>
<tr>
<td></td>
<td>Coordination</td>
<td>transport and warehousing order</td>
</tr>
<tr>
<td>Semantic</td>
<td></td>
<td>transport request</td>
</tr>
</tbody>
</table>

Table 5.8 Variants of the term "transport order" (adopted from [Maynard 2000])
In order to make terms comparable, there are various techniques for normalizing those variations. Following [Euzenat/Shvaiko 2007] there are four main normalization functions: (1) tokenization, (2) lemmatization, (3) term extraction and (4) stopword elimination.

- **Tokenization** had already been mentioned within the explanation of the token-based distance measure. Basically, tokenization aims at segmenting a string into a sequence of tokens by recognizing special string characteristics. Those characteristics can be cases, digits, space characters, punctuations, or any kind of special characters. For example, the strings "requested transport order" and "RequestedTransportOrder" become both <requested, transport, order>.

- **Lemmatisation** means retrieving the root from a word. This is done by analyzing morphological variations and extracting flexions and derivations. Lemmatization involves suppressing gender, tens, and inflections (e.g. "transport orders" become "transport order"). Another approximate lemmatization mechanism is stemming [Porter 1997]. Stemming removes suffixes from terms, e.g. "requested" becomes "request".

- **Term extraction** is about extracting relevant terms from a given corpus. Relevant terms are identified by their occurrence. However, not only similar strings are recognized, but morphological similar phrases as well. Morphological similar phrases are identified by predefined patterns, such as that "noun1 noun2" is the same as "noun2 of noun1". This leads to the fact that "transport order" and "order of transport" are the same. Among others, term extraction is considered in [Bourigault/Jacquemin 1999], [Maynard/Ananiadou 1999] and [Cerbah/Euzenat 2001].

- **Stopword elimination** means that tokens like articles, conjunctions and prepositions, etc. (e.g. words like "to", "the", "of", "on" or "a") are disregarded, because of being considered as empty words, which are not meaningful for matching. For example, the "an order of transport" becomes "order transport".

After applying these functions words are represented as sets of terms that can be compared using other matching techniques.

**Extrinsic methods**

Extrinsic linguistic methods use external resources for finding similarities between terms. There are several kinds of linguistic resources, such as lexicons, thesauri, terminologies and more, which are explained in the following.

- **Lexicons, or dictionaries**, contain a set of words and a natural language description for each of them. Of course each word may have several of those descriptions. For instance, looking up the word "transfer" in Wikipedia [Wikimedia Foundation Inc.], which is a lexicon according to the definition above, there are 24 possible descriptions for this term [Wikipedia 2011b]. However, lexicons are helpful resources for similarity measures (see gloss-based distance below)

- **Multi-lingual lexicons** are certainly lexicons, but provide translations and descriptions of terms in another language, e.g. "transport order" in English corres-
ponds to "Transportauftrag" in German. Those dictionaries can be helpful considering ontology labels in different language. Since, different languages are out of scope in this thesis, multi-lingual lexicons are not considered any further.

- **Semantic-syntactic lexicons** record not only names, but functions and rules of these words. For instance, the term "to transport" is a verb which has any type of a means of transport as a subject and a transportation item as an object. However, since those lexicons are difficult to create and since (to the best of the author knowledge) there are no such lexicons for logistics these type of external resource is not considered any further either.

- **A thesaurus** is a lexicon, which contains terms and a description, but additionally relational information, such as hypernyms, hyponyms, synonyms and antonyms. An example for a thesaurus is WordNet [Miller 1995].

- **A terminology** is a glossary of terms. It contains explanations to those terms and includes relations to other terms as well. Thus a terminology is similar to a thesaurus, but contains phrases instead of words. Terminologies are usually domain specific and less equivocal then lexicons.

The resources mentioned above might provide general or domain specific information. A domain specific resource tends to be more adapted, since the resource may contain specialized senses, proper names and proper abbreviations. For example, in the logistics domain the abbreviation "FCA" refers to the incoterm "Free Carrier". In other domains "FCA" is interpreted as "Full cost accounting", "Function cost analysis", or even "Franchise Council of Australia".

However, since there is no common external linguistic resource for the logistics domain known so far, the WordNet [Miller 1995] is considered in the following. WordNet is a lexical database for English language. WordNet provides words including their semantic relations. However, the basis of WordNet are synonyms, i.e. each word contains a set of synonyms called synset. Other semantic relations, such as hypernyms, meronyms, hyponyms, holonyms, and antonyms, are provided as well. Beside semantic relations, each synset contains a general description and one or more examples for the words usage. Over the years, WordNet has been developed and extended. The latest version v3.0 from 2006 contains 155,287 words organized in 117,659 synsets. The database is free for download and use. Moreover, WordNet is searchable via a Web interface [Princeton University]. Beside WordNet there are initiatives that provide lexical databases for other language, cf. EuroWordNet [ELDA - Evaluations and Language resourcees Distribution Agency]. WordNet distinguishes between nouns, verbs, adjectives, and adverbs. However, WordNet does not comprise prepositions (e.g. of, to, in, for, with, on, etc.) and determines (e.g. the, my, a, some, which, both, etc.). In the following, nouns are in focus, since nouns are mainly in use for ontology concept names. Regarding nouns WordNet provides hyponyms, hypernyms, meronyms, holonyms, and antonyms within each synset.

- **Hyponyms**: X is a hyponym of Y if every X is a (kind of) Y, e.g. a truck driver is a hyponym of a driver, since every truck driver is a driver as well.
• **Hypernym**: X is a hypernym of Y if every Y is a (kind of) X, e.g. a driver is a hypernym of a truck driver, since every truck driver is a driver as well.

• **Meronym**: X is a meronym of Y if X is a part or a member of Y, e.g. Germany is a meronym of Europe, since Germany is a part of Europe.

• **Holonym**: X is a holonym of Y if Y is a part or a member of X, e.g. Europe is a holonym of Germany, since Germany is a part of Europe.

• **Antonym**: X is an antonym of Y if X is the opposite Y or (in case of X and Y share the same parent) X is a distinct sibling of Y. For example, a non-driver is an antonym of a driver. Furthermore, since Germany and France are distinct siblings of Europe, Germany is an antonym of France and vice versa.

Synset examples for some logistics terms are shown in Appendix N.

[Euzenat/Shvaiko 2007] mentions three methods for using WordNet as a resource for matching ontology entities:

• Considering two terms as similar, because they belong to some common synset

• Taking hypernym and hyponym structure into account and measuring the distance between two synsets that corresponds to two ontology entities

• Taking the term descriptions provided by WordNet into account in order to conclude about the distance between two synsets that corresponds to two entities

A matcher has to translate the relations used in WordNet into logical relations.

[McGuinness/Harmelen 2004] provides the following rules for that purpose.

• If t is a hyponym or meronym of t’ the relation between t and t’ is denoted as $t \subseteq t'$. So, hyponyms and meronyms are treated equally for matching purposes. For example a truck is a hyponym of a means of transport. Thus, the relation is denoted as $truck \subseteq means\ of\ transport$. Further, a wheel is a meronym of a truck, and is denoted as $wheel \subseteq truck$.

• If t is a hypernym or holonym of t’ the relation between t and t’ is denoted as $t \supseteq t'$. So, hypernyms and holonyms are treated equally for matching purposes. For example employee is a hypernym of driver. Thus, the relation is denoted as $employee \supseteq driver$. Further, a Europe is a holonym of Germany, and is denoted as $Europe \supseteq Germany$.

• If t and t’ are connected as a synonyms or they belong to the same synset the relation is denoted as $t = t'$. For example, a transport is a synonym for shipping. Thus, this relation is denoted as $transport = shipping$.

• If t and t’ are distinct siblings or denoted explicitly as antonyms the relation is defined as $t \perp t'$. For example, France and Germany are distinct siblings of Europe. Thus France and Germany are antonyms and can be denoted as $France \perp Germany$. Further, a direct antonym of "driver" is a "nodriver" (according to WordNet), i.e. $driver \perp nodriver$.

Synsets are the basis of WordNet. For simplification purposes only synonym relations are considered in the following. However, other semantic relations (i.e. hy-

---

31 A parent can be a hyponym or a holonym.
pony, meronym, hypernym, holonym and antonyms) can be treated in a similar way. Synonym measures is provided by the synonym similarity.

Given two terms s and t and a synonym resource $\Sigma$, the synonym similarity is a similarity such that:

$$\forall s, t \in S, \sigma(s, t) = \begin{cases} 1 & \text{if } \Sigma(s) \cap \Sigma(t) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

**Definition 5.23 Synonym similarity (cf. [Euzena/Shvaiko 2007])**

Considering the logistics terms synonym similarity cannot be applied out of the box, since WordNet does not contain compound nouns, such as "transport order", "source address", etc. Thus, a modification of a synonym similarity is necessary. A modification is provided by the synonym substring similarity.

Given two terms s and t and synonym resource $\Sigma$, where $\Sigma(s)$ is the set of synonyms of s and $\Sigma(s,i)$ is a particular synonym of s, and where $\Sigma(t)$ is the set of synonyms of t and $\Sigma(t,j)$ is a particular synonym of t. Further, let $|\Sigma(s)|$ be the number of synonyms of s, let $|\Sigma(t)|$ be the number of synonyms of t and $x_{i,j}$ be the longest common substring of $\Sigma(s,i)$ and $\Sigma(t,j)$. Then, the synonym substring similarity is defined as the maximum of the substring similarity of all synonyms of s and t:

$$\forall s, t \in S, \sigma(s, t) = \max \left( \sum_{i=1}^{|\Sigma(s)|} \sum_{j=1}^{|\Sigma(t)|} \frac{2|x_{i,j}|}{|\Sigma(s,i)| + |\Sigma(t,j)|} \right)$$

**Definition 5.24 Synonym substring similarity (cf. [Euzena/Shvaiko 2007])**

Taking tokens into account another modification of synonym similarity is provided by the token based cosine synonym similarity.

Given two terms s and t and represented as multisets $\tilde{s}$ and $\tilde{t}$ of a metric space V and a synonym resource $\Sigma$. Each single word $\tilde{s}_i$ relates to a synset $\Sigma(\tilde{s}_i)$ and each single word $\tilde{t}_j$ relates to a synset $\Sigma(\tilde{t}_j)$ as well. The token based cosine synonym similarity is defined as:

$$\forall s, t \in S, \sigma(s, t) = \frac{\tilde{s} \cdot \tilde{t}}{|\tilde{s}| \ast |\tilde{t}|}$$

with $\tilde{s}_i = \tilde{t}_j$, if $\Sigma(\tilde{s}_i) \cap \Sigma(\tilde{t}_j) \neq 0$

**Definition 5.25 Token based cosine synonym similarity (cf. [Euzena/Shvaiko 2007])**

The application of the token based cosine synonym similarity leads to Table 5.9.
In comparison to string-based techniques the table above shows some improvements, but some drawbacks as well. As an improvement, the term "pickup time" and "pickup period" are perfectly matched (1.00), since time and period are connected via a synset (i.e. "time period, period of time, period (an amount of time)"). That means, by adding synonyms, i.e. increasing interpretation of a word, the chances of finding matching terms (called true positives) increases. On the other side this increases the amount of available labels (i.e. homonyms) for a matching entity and, thus, the chances for matching non-matching terms (called false positives). An example for a false positive is the matching between "pickup time" and "truck" with a similarity of 0.71. Reason for that is that pickup belongs to the synset "pickup, pickup truck" with the description "a light truck with an open body and low sides and a tailboard". Other false positive matches are, for instance "source address" and "shipment request" as well as "destination address" and "shipment request". The problem dealing with false positives is known as word sense disambiguation [Lesk 1986]. Word sense disambiguation tries to estimate the sense of a word and restricts the match to this sense. This is done by selecting the sense in relation to other associated words and, thus, considering the context of a word. Putting a word into a certain context can be done by considering the terms’ domain. As an extension of WordNet the Research Institute for Artificial Intelligence of the Romanian Academy provides an online WordNet browser [RACAI - Research Institute for Artificial Intelligence] that relates a synset to a distinct domain. 166 unique domains had been defined so far including, for instance, enterprise, transport, commerce, and the neutral domain factotum. Filtering the WordNet synset by these domains leads to a more accurate measure with less false positive matching results.

Another issue of the results in Table 5.9 is that some of the terms that should obviously match, such as "transport order" and "shipment request", show no match (similarity = 0.00), even though "transport" and "shipping" belong to the same synset. Furthermore, the WordNet description of "order" contains "... a commercial document used to request someone …". The later issue can be resolved by the

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>transport order</td>
<td>0.71</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.32</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.71</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 5.9 Token based cosine synonym similarity example
gloss overlap similarity which will be considered later on. However, the former issue can be resolved by using a lemmatization mechanism, such as suffix stemming [Porter 1997]. So, "shipment" and "shipping" becomes "ship" and match perfectly.

Measuring the token based cosine synonym similarity, including a domain filter (i.e. considering synset of the domain enterprise, commerce, and transport only) and applying a suffix stemming previously the following results appear.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.71</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>transport order</td>
<td>0.71</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.71</td>
<td>0.00</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.71</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 5.10 Example of token based cosine synonym similarity with domain filter and suffix stemming

As shown in Table 5.10 "transport order" and "shipment request" match better than previously, since "shipment" and "transport" are related as synonyms here. Further, some false positives disappeared, since the domain filter matches terms from transport, enterprise and commerce domain only. However, the false positive match between "pickup …" and "truck …" still occurs, since the synset of "pickup" belongs to the transport domain unfortunately. Here, further mechanisms have to be applied in order to estimate about the context in a better way.

Synonym similarity and its token based variation work fine for synsets. However, both measures do not consider the distance between synonyms, i.e. how close two synonymous objects are. Since synsets are connected, WordNet can be represented as a graph and, thus, graph based measures work fine for synonymous relations. Some measures take the hyponym and hypernym hierarchy into account. As a simple measure, edge-count counts the numbers of edges separating two synsets. Additionally, weight edge-count weights the position of synsets in the hierarchy of the graph. A measure which has specifically developed for WordNet is proposed by Wu and Palmer [Wu/Palmer 1994], which is considered in 5.3.2.2.

Another way of measuring the relationship between terms is accomplished by taking the terms description (gloss) into account. WordNet is a good resource for that purpose, since it contains a definition for each word, e.g. transportation: "the commercial enterprise of moving goods and materials". More formal, each entry of the thesaurus $s \in \Sigma$ relates to a set of words presented by $\lambda(s)$. This way, any
string-based measure can be used for comparing strings. Especially token-based techniques are meaningful for gloss-based similarity measurement, e.g. the gloss overlap similarity by [Lesk 1986, Banerjee/Pedersen 2003]. Gloss overlap is based on the Jaccard similarity [Jaccard 1901], which is used for measuring the similarity of sets and builds the foundation of the cosine similarity.

Given a partially ordered synonym resource \( \Sigma \) bearing terms \( s, t \in \Sigma \) and definitions of each term \( \lambda(s), \lambda(t) \), the gloss overlap between the two terms is defined by the Jaccard similarity between their glosses:

\[
\forall s, t \in \Sigma, \sigma(s, t) = \frac{|\lambda(s) \cap \lambda(t)|}{|\lambda(s) \cup \lambda(t)|}
\]

**Definition 5.26** Jaccard similarity (cf. [Jaccard 1901])

Calculating gloss overlap for the logistic terms, the following constraints are defined previously, for instance:

- Consider the term definitions from the domains "transport", "enterprise", "commerce", "time_period" only.
- Suppress quotations, suppress special characters (such as '.', ',', ';', ':', '?', ':', '(', ')', etc.), suppress empty words (such as "or", "and", "with", "to", "the", "and", "an", "a"), suppress adjectives (such as "top", "heavy", "suitable", "similar"), suppress all numbers (such as "one", "two", '1', '2', etc.), suppress verbs that are used inflationary (such as "is", "are", "be", "been", "has", "have", etc.), and suppress empty phrases (e.g. "usually including") such that there are mainly nouns and expressive verbs within the gloss corpus.
- Stem words, e.g. "transportation" becomes "transport", "shipping" and "shipment" are stemmed to "ship" both.
- Each word in the gloss corpus appear only once.
- The term itself is included into the gloss corpus.
- Since compound nouns are not supported by WordNet, compound nouns are split into single words in order to gain their definition separately. Next, the gloss corpus of the compound noun is gained by concatenation. Considering the constraints above, this means that the gloss corpus of "transport order" is the concatenation from gloss corpus of "transport" ("commercial", "enterprise", "goods", "material"), and "order" ("commercial", "document", "request", "supply", "payment", "specification", "quantities"), which results in the following gloss corpus "transport order" is "commercial", "enterprise", "goods", "material", "document", "request", "supply", "payment", "specification", "quantities".

Applying these rules to the logistic terms and calculating the gloss overlap results in the following table.
Gloss overlap shows good results for transport for "transport order" and "transport", for "pickup time" and "pickup period" and for "truck" and "truck cargo". This is because at least one word of these strings is included within the other one. Further, there are other significant results. For instance "transport order" and "shipment request" indicates a gloss overlap of 0.1, since order is defined as "... a commercial document used to request ..." in WordNet. Nevertheless, this is a relatively small number. Even though that a lot of words (such as stopwords and non-meaningful verbs) had been suppressed before gloss overlap measure, the result of 0.10 does not represent that there is an obvious match between "order" and "request". In order to cope with this fact it is meaningful to weight gloss overlap measures. Thus the gloss overlap is modified by a weight factor that multiplies the ratio of the max gloss corpus size and the min gloss corpus size to each gloss overlap measure value.

\[
\forall s, t \in S, \sigma(s, t) = \frac{\max(|\lambda(s)|,|\lambda(t)|)}{\min(|\lambda(s)|,|\lambda(t)|)} \times \frac{|\lambda(s) \cap \lambda(t)|}{|\lambda(s) \cup \lambda(t)|}
\]

Definition 5.27 Weighted gloss overlap (cf. [Lesk 1986, Banerjee/Pedersen 2003])

The weighted gloss overlap results in the following table.
As shown in the table above, the measure of "transport order" and "shipment request" increases from 0.10 to 0.27. However, there is an issue with false positive matches that becomes even worse after applying the weight gloss overlap mechanism. For instance, the term "good" has a high gloss overlap with "transport", "shipment request" and "customer". The reason for this is that "good" is often used in WordNet definitions (e.g. customer: "… someone who pays for goods...").

Apart from weighting values, there are other modifications of gloss overlap, such as [Euzenat/Shvaiko 2007] and [Giunchiglia/Yatskevich 2004]. However, the results of these modifications as well as gloss overlap itself depend on the context and linguistic resources. Thus these matcher cannot be applied in all cases [Giunchiglia et al. 2006] and not as a standalone solution.

Instead, it is meaningful to combine gloss overlap measure with other measures. For instance, one can combine the weighted gloss overlap with the token-based cosine synonym similarity mentioned above. A feasible way for combining measurement is mean measurement. Basically there are three different ways of calculating the mean: (1) arithmetic mean, (2) geometric mean, and (3) harmonic mean.

\[
\text{(1) } \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \\
\text{(2) } \bar{x} = \left( \prod_{i=1}^{n} x_i \right)^{1/n} \\
\text{(3) } \bar{x} = n \times \left( \sum_{i=1}^{n} \frac{1}{x_i} \right)^{-1}
\]

**Definition 5.28** Mean calculation, (1) arithmetic mean, (2) geometric mean, and (3) harmonic mean

The arithmetic mean is the sum of all values divided by the number of the values. The geometric mean is the average useful for positive numbers that are interpreted according to their product. Further, the geometric mean can be used to mask values with zero numbers, because the result is zero no matter how big are all other factors are when one factor of the product is zero. Last not least the harmonic mean is an average mean that is useful for numbers which are defined in relation to some units.

In case of the results of the token-based cosine synonym similarity (TBCS) and the weighted gloss overlap (WGO) similarity it is feasible to choose the geometric mean in order to mask zero values. This means that if one matching result of either the TBCS or the WGO similarity is zero the combination of both measures is zero, no matter how high the value of the other measurement is. For example, the matching result of "transport" and "good" is 0.00 with TBCS and 1.00 with WGO and, thus the geometric means results in 0.00. On the other hand "truck cargo" and "truck" results in 0.84. Results are shown in Table 5.13.

As shown in Table 5.13, some false positive matches from the WGO results had been masked. However, beside some true positives, such as "pickup time" and "pickup period", and "truck cargo" and "truck", some issues remain. For example, the match between "transport order" and "shipment request" does not have the

<table>
<thead>
<tr>
<th>Good</th>
<th>1.00</th>
<th>1.00</th>
<th>1.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truck cargo</td>
<td>0.12</td>
<td>0.41</td>
<td>0.21</td>
<td>0.00</td>
<td>0.05</td>
<td>0.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.12 Weighted gloss overlap

As shown in the table above the measure of "transport order" and "shipment request" increases from 0.10 to 0.27. However, there is an issue with false positive matches that becomes even worse after applying the weight gloss overlap mechanism. For instance, the term "good" has a high gloss overlap with "transport", "shipment request" and "customer". The reason for this is that "good" is often used in WordNet definitions (e.g. customer: "… someone who pays for goods...").

Apart from weighting values, there are other modifications of gloss overlap, such as [Euzenat/Shvaiko 2007] and [Giunchiglia/Yatskevich 2004]. However, the results of these modifications as well as gloss overlap itself depend on the context and linguistic resources. Thus these matcher cannot be applied in all cases [Giunchiglia et al. 2006] and not as a standalone solution.

Instead, it is meaningful to combine gloss overlap measure with other measures. For instance, one can combine the weighted gloss overlap with the token-based cosine synonym similarity mentioned above. A feasible way for combining measurement is mean measurement. Basically there are three different ways of calculating the mean: (1) arithmetic mean, (2) geometric mean, and (3) harmonic mean.

\[
\text{(1) } \bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \\
\text{(2) } \bar{x} = \left( \prod_{i=1}^{n} x_i \right)^{1/n} \\
\text{(3) } \bar{x} = n \times \left( \sum_{i=1}^{n} \frac{1}{x_i} \right)^{-1}
\]

**Definition 5.28** Mean calculation, (1) arithmetic mean, (2) geometric mean, and (3) harmonic mean

The arithmetic mean is the sum of all values divided by the number of the values. The geometric mean is the average useful for positive numbers that are interpreted according to their product. Further, the geometric mean can be used to mask values with zero numbers, because the result is zero no matter how big are all other factors are when one factor of the product is zero. Last not least the harmonic mean is an average mean that is useful for numbers which are defined in relation to some units.

In case of the results of the token-based cosine synonym similarity (TBCS) and the weighted gloss overlap (WGO) similarity it is feasible to choose the geometric mean in order to mask zero values. This means that if one matching result of either the TBCS or the WGO similarity is zero the combination of both measures is zero, no matter how high the value of the other measurement is. For example, the matching result of "transport" and "good" is 0.00 with TBCS and 1.00 with WGO and, thus the geometric means results in 0.00. On the other hand "truck cargo" and "truck" results in 0.84. Results are shown in Table 5.13.

As shown in Table 5.13, some false positive matches from the WGO results had been masked. However, beside some true positives, such as "pickup time" and "pickup period", and "truck cargo" and "truck", some issues remain. For example, the match between "transport order" and "shipment request" does not have the
highest value in the row and in the column either. Further, the false positive problem resulting from the homonym relation between "truck" and "pickup" remains.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.25</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>transport order</td>
<td>0.84</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.45</td>
<td>0.00</td>
<td>0.22</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.29</td>
<td>0.45</td>
<td>0.00</td>
<td>0.00</td>
<td>0.16</td>
<td>0.00</td>
<td>0.84</td>
</tr>
</tbody>
</table>

Table 5.13 Geometric mean values with the combination of TBCS and WGO

Summary on language-based techniques

Two methods of language-based techniques, i.e. intrinsic methods and extrinsic methods, have been considered here. However, main issue of language-base techniques are false positive matches, which cannot be prevented easily. A proper matching result depend on a good sense for intrinsic normalization and the quality of an external linguistic resource. In order to cope with the issues of language-based methods, one can take service entities structure into account. Structure base-techniques are examined next.

5.3.2 Structure-based techniques

The structure of the service entities are defined in the SCMM. Each entity is usually one or many times connected with another one. For instance, an ontology concept may have data properties bearing a name and pointing to a certain data type. So, instead of comparing names and identifiers only (as suggested in the chapters above) matching these relations is considered by structure-based techniques. Structure-based techniques can be subdivided into techniques that consider the internal structure (cf. 5.3.2.1) and techniques that consider relations between entities (cf. 5.3.2.2).

5.3.2.1 Internal structure techniques

Internal structure techniques consider the internal structure of service entities. Internal structure comprises entity properties, domains, and ranges of properties, the cardinality or multiplicity of properties, and properties characteristics, such as
transitivity and symmetry. However, variation of entities internal structure can be huge. Thus, for matching based on entities internal structure techniques commonly establish correspondence clusters, instead of discovering exact correspondence measures. Further, these techniques are usually combined with other techniques. As an example the following figure shows the properties of the entities "Good" and "ItemData".

![Entity properties example](image)

Figure 5.2 Internal structure example

Entity properties' characteristics can be subdivided into names, datatypes, cardinalities or multiplicities, and others. Name-based techniques have been considered previously (cf. 5.3.1). The similarity of datatypes, cardinalities and other characteristics is considered here.

**Datatype similarity**

Comparing a property of an ontology concept involves the comparison of the property type. The SCMM defines data properties. Data properties do have distinct set of data types, such as string, integer, decimal or date time. Datatypes correspond to the way how values are stored in a computer. These datatypes used in several ways such as database tables, xml schema and OWL/RDF as well. Datatypes are not fully disjoint, since there are rules by which one value of a certain data type can be converted into a value of another data type\(^\text{32}\). Based on the ability of converting a value of one datatype into a value of another datatype one can conclude about the similarity between these datatypes.

**Definition 5.29** Datatype similarity (cf. [Euzenat/Shvaiko 2007])

However, the exact value for the similarity between \(e\) and \(d\) can be estimated only. For estimating the datatype similarity one could consider a datatype hierarchy. An

\(^{32}\) Conversion is known as casting in programming language.
example is given by the XML schema datatype from [World Wide Web Consortium (W3C) 2001].

![Diagram of XML Schema datatype hierarchy](image)

*Figure 5.3 Part of the XML Schema datatype hierarchy [W3C 2001, World Wide Web Consortium (W3C) 2001]*

Considering the direction of the conversion, i.e. converting an integer into a string or converting a string into an integer respectively, the estimated value might differ. For example, converting an integer into string is possible easily, but not all strings can be converted into integers. Since, the similarity measure is always direction independent one could estimate the conversion value separately and calculate the arithmetic mean afterwards. For instance, conversion value for integer to string is 0.9 and the value for converting string to integer is 0.1, which results in a similarity value of 0.5. Following the table in [Euzenat/Shvaiko 2007], but considering the datatypes mentioned in the SCMM and the conversion direction, the following table occurs.

<table>
<thead>
<tr>
<th></th>
<th>Boolean</th>
<th>DateTime</th>
<th>Decimal</th>
<th>Integer</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>1.00</td>
<td>0.00</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>DateTime</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.90</td>
</tr>
<tr>
<td>Decimal</td>
<td>0.10</td>
<td>0.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.90</td>
</tr>
<tr>
<td>Integer</td>
<td>0.25</td>
<td>0.00</td>
<td>0.90</td>
<td>1.00</td>
<td>0.90</td>
</tr>
<tr>
<td>String</td>
<td>0.10</td>
<td>0.10</td>
<td>0.10</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

*Table 5.14 Conversion proximity between row and column datatype*

The table above shows the estimated conversion from each row datatype row into the datatype in the particular column. The conversion is estimated by the following rules:

- 1.00 if datatypes are the same
- 0.90 if datatypes can be converted in all cases
- 0.50 if datatypes can be converted in some cases, but in some cases not
• 0.25 if datatypes can be converted in less cases
• 0.10 if datatype can be converted in raw cases
• 0.00 if datatypes can’t be converted

Applying the arithmetic mean, in order to suppress the influence of direction, leads to the following datatype compatibility table.

<table>
<thead>
<tr>
<th></th>
<th>Boolean</th>
<th>DateTime</th>
<th>Decimal</th>
<th>Integer</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>1.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.58</td>
<td>0.50</td>
</tr>
<tr>
<td>DateTime</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
</tr>
<tr>
<td>Decimal</td>
<td>0.50</td>
<td>0.00</td>
<td>1.00</td>
<td>0.70</td>
<td>0.50</td>
</tr>
<tr>
<td>Integer</td>
<td>0.58</td>
<td>0.00</td>
<td>0.70</td>
<td>1.00</td>
<td>0.50</td>
</tr>
<tr>
<td>String</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.15 Datatype compatibility matrix

Since these similarity measures are based on estimations only, other values could be chosen as well. Considering Table 5.15 and applying datatype comparison to the datatype properties using the values of this table, the following matching values of datatype similarities appear (cf. Table 5.16)

<table>
<thead>
<tr>
<th>Item</th>
<th>hasItemNumber&lt;Integer&gt;</th>
<th>hasItemName&lt;String&gt;</th>
<th>hasItemDescription&lt;String&gt;</th>
<th>hasGrossWeights&lt;Integer&gt;</th>
<th>hasGrossSizeBreadth&lt;Integer&gt;</th>
<th>hasGrossSizeHeight&lt;Integer&gt;</th>
<th>hasGrossSizeLength&lt;Integer&gt;</th>
<th>hasVolume&lt;Decimal&gt;</th>
<th>hasKg&lt;Decimal&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasID&lt;String&gt;</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasPackagingType&lt;String&gt;</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasDescription&lt;String&gt;</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasPackagingAmount&lt;Integer&gt;</td>
<td>1.00</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasVolume&lt;Decimal&gt;</td>
<td>0.70</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>hasKg&lt;Decimal&gt;</td>
<td>0.70</td>
<td>0.50</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
<td>0.70</td>
</tr>
</tbody>
</table>

Table 5.16 Data property comparison based on datatypes

As shown in Table 5.16 there are some correct matches such as hasDescription and hasItemDescription. However, most of the datatype matches cannot conclude about the correct matching level of the data properties. Thus, the datatype match based on datatype comparison cannot be used in isolation.
Cardinality similarity

Properties can be constrained by cardinalities. Similar to the comparison of data-types, the compatibility of cardinalities can be measured based on a table. Among others, [Lee et al. 2002] provides a cardinality compatibility table for DTD, where ‘*’ stands for zero or many, ‘+’ stands for one ore many, ‘?’ means zero or one, and the ‘none’ cardinality is default and means exactly one.

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>+</th>
<th>?</th>
<th>none</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>1.00</td>
<td>0.90</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>+</td>
<td>0.90</td>
<td>1.00</td>
<td>0.70</td>
<td>0.70</td>
</tr>
<tr>
<td>?</td>
<td>0.70</td>
<td>0.70</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>none</td>
<td>0.70</td>
<td>0.70</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.17 Cardinality compatibility [Lee et al. 2002]

A more sophisticated way is calculating the cardinality compatibility. SCMM defines cardinalities by a LowerBound and an UpperBound, i.e. cardinalities are expressed by an interval of positive integers. Thus, in addition to a cardinality compatibility table, one may defined that two cardinalities are (more or less) compatible if the intersection of both intervals is not empty [Euzenat/Shvaiko 2007]. In order to calculate the compatibility the cardinality similarity is defined as follows (adopted from the multiplicity similarity from [Euzenat/Shvaiko 2007]):

Given two cardinality expressions \([e \ f]\) and \([e'\ f']\), the cardinality similarity \(\sigma\) is a similarity between non negative integer intervals \([0 1]\) such that:

\[
\sigma([e \ f],[e'\ f']) = \begin{cases} 
0, & \text{if } e' > f \text{ or } e > f' \\
\frac{(\min(f, f') - \max(e, e')) + 1}{(\max(f, f') - \min(e, e')) + 1}, & \text{otherwise}
\end{cases}
\]

Definition 5.30 Cardinality similarity (cf. [Euzenat/Shvaiko 2007])

For example, if one calculates the cardinality of \([0 10]\) with \([1 5]\), \([5 12]\), \([12 12]\) and \([0 \infty]\), the comparison result is: 0.45, 0.46, 0.00 and 11/∞. The latter is very low but not null because it remains compatible in the interval from 0 to 10. Applying the cardinality similarity measure to the properties of the example above leads to the results in the table below.
As shown in the table above the cardinality similarity can’t be applied in isolation. However in combination with other techniques, cardinality similarity measure provides a useful contribution. As suggested by [Euzenat/Valtchev 2004] one should consider datatypes (and their hierarchy) for cardinality similarity measure, since a set of 1 to 4 children is closer related to a set of 3 people than 3 flowers (assuming that children are people), for example. The following table shows the results by combining datatype comparison results (cf. Table 5.16) and cardinality similarity measure (cf. Table 5.18) using an arithmetic mean.

<table>
<thead>
<tr>
<th>Item</th>
<th>hasItemNumber[1..1]</th>
<th>hasItemName[0..1]</th>
<th>hasItemDescription[1..1]</th>
<th>hasGrossWeight[1..1]</th>
<th>hasGrossSizeBreadth[1..1]</th>
<th>hasGrossSizeHeight[1..1]</th>
<th>hasGrossSizeLength[1..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasID[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>hasPackagingType[0..*]</td>
<td>1/00</td>
<td>2/00</td>
<td>1/00</td>
<td>1/00</td>
<td>1/00</td>
<td>1/00</td>
<td>1/00</td>
</tr>
<tr>
<td>hasDescription[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>hasPackagingAmount[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>hasVolume[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>hasKg[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

*Table 5.18 Cardinality similarity measure*

<table>
<thead>
<tr>
<th>Item</th>
<th>hasItemNumber-Integer[1..1]</th>
<th>hasItemName-String[0..1]</th>
<th>hasItemDescription-String[1..1]</th>
<th>hasGrossWeight-Integer[1..1]</th>
<th>hasGrossSizeBreadth-Integer[1..1]</th>
<th>hasGrossSizeHeight-Integer[1..1]</th>
<th>hasGrossSizeLength-Integer[1..1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasID&lt;String&gt;[1..1]</td>
<td>0.75</td>
<td>0.75</td>
<td>1.00</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>hasPackagingType&lt;String&gt;[0..*]</td>
<td>0.25</td>
<td>0.50</td>
<td>0.50</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>hasDescription&lt;String&gt;[1..1]</td>
<td>0.75</td>
<td>0.75</td>
<td>1.00</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
<tr>
<td>hasPackagingAmount&lt;Integer&gt;[1..1]</td>
<td>1.00</td>
<td>0.50</td>
<td>0.75</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>hasVolume&lt;Decimal&gt;[1..1]</td>
<td>0.85</td>
<td>0.50</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>hasKg&lt;Decimal&gt;[1..1]</td>
<td>0.85</td>
<td>0.50</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
<td>0.85</td>
</tr>
</tbody>
</table>

*Table 5.19 Arithmetic means of datatype comparison and cardinality similarity*
The results in the table above show a better matching result of the internal structure. However, their application is not useful in isolation either. For instance, the meaning, i.e. names and descriptions, of the data properties haven’t been considered so far.

Others

There are other internal structure factors that have not been considered yet. Most of these techniques are related to database schema comparison. Unlike to service models based on the SCMM, database schemas provide keys in order to identify unique objects. For instance, a book has an international standard book number (isbn) that identifies a single book and a person might be identified by a combined key comprising name, address and birth date. However, since the SCMM does not include primary keys (or properties that are defined as unique identifiers) keys as a feature for comparison is skipped here. Other internal structure features such as specific collection types or constraints are not considered here either, since they are not part of the SCMM. Specific collection types can be sets, lists, or bags, which are used in object oriented databases [Kim 1990]. Specific constraints are, for instance, domain definition for datatypes, e.g. the age of a person ranges from 0 to 130. All specific internal structure characteristic, have to be considered, if they are defined in the SCMM. The SCMM defined here does not have such characteristics. Thus, all ‘other’ internal structure characteristics are neglected here.

Summary on internal structure techniques

Internal structure techniques consider the internal structure of service entities. However, the internal structure does not provide much information about the entity itself, i.e. many very different entities may have similar data properties with same cardinalities and even same datatypes. On the other hand matching based on internal structure characteristics can highlight compatible and incompatible properties. In turn, different models may use different (incompatible) types, which might be compatible though transformation. As suggested previously, internal structure techniques should not be used in isolation, but jointly with other techniques.

5.3.2.2 Relational structure techniques

Relational structure techniques consider the relations of service entities. The similarity between two entities from two service models is calculated based on the similarity of their related entities. With other words, two entities are more alike the more their related entities are alike. Since there are several kinds of relations, there are several kinds of the similarity measure based on entities relations, which can be exploited in several ways. [Euzenat et al. 2004] names 5 different relation types, which can be taken into account for comparison:

- r comparing entities in the direct relation
• $r^r$ comparing entities in the transitive reduction of the relation
• $r^+$ comparing entities in the transitive closure of the relation
• $r^1$ comparing entities coming through a relation
• $r^\uparrow$ comparing entities which are ultimately in $r^+$ (these are the maximum elements of the transitive closure, i.e. the leafs of the subgraph in a taxonomic structure)

A typical relational structure is shown in Figure 5.4.

![Figure 5.4 Fragment of a service model](image)

For instance, the relation type subConcept (subC) of VehicleResource (VR) is:

$$
\text{subC}(\text{VR}) = \text{subC}(\text{VR}) = \{\text{Truck, Ship, Aircraft}\}
\text{
subC}^+(\text{VR}) = \{\text{Truck, Ship, Aircraft, MinTruck, LightTruck, MediumTruck, HeavyTruck}\}
\text{
subC}^1(\text{VR}) = \{\text{Resource}\}
\text{
subC}^\uparrow(\text{VR}) = \{\text{Ship, Aircraft, MinTruck, LightTruck, MediumTruck, HeavyTruck}\}
$$

The subConcept relation is the only relation of the OntologyConcept service concept defined in SCMM. Beside this particular relation, there are other relations and other service concepts defined. In order to cope with several relations one should create a features table. A feature table had been defined for ontology entities by [Ehrig 2006] and [Euzenat/Shvaiko 2007] so far. An adapted version of the feature table for SCMM base service entities is provided in Table 5.20.

Features mentioned in the table indicate techniques for comparing service entities. Basically, two service entities are similar if their features are similar. Thus each feature should be compared one by one and aggregated later on. Some features are from type String and can be compared using name-based techniques. Other features are from type DataProperty and can be compared using internal structure techniques. However, features from other types indicate relations between entities. Furthermore, features from type Set( …) are more difficult to deal with, since these types refer to multisets of a particular type.
Table 5.20 Features for comparing service entities

In order to deal with relations ontology matching considered ontologies as a graph [Euzenat/Shvaiko 2007]. A graph $G$ is defined as a tuple $(V, E)$ where $V$ refers to a set of vertices and $E$ refers to a set of edges. By considering an ontology as a

---

This feature is not covered by a Service Concept Property. However, there are three distinct resource types defined, i.e. HumanResource, MaterialResource, ImmaterialResource.
graph each vertex refers to a single concept and each edge refers to a labeled relation between these concepts. Similarity measures between graphs relate to solving the maximum common subgraph isomorphism problem [Garey/Johnson 1979]. Adapted to service matching, a service model can be considered as a graph as well. Thus graph-based matching techniques can be applied. Techniques for matching graphs, i.e. matching the relational structures of service models, are subdivided into taxonomic relations, mereologic relations and all relations in literature. These techniques are examined next.

**Taxonomic structure**

Taxonomic structure means that a graph is made of sub and super concept relations. Sub and super concept relations appear in the service model for Abstract-Service, Concrete-Service, and within the OntologyConcept (i.e. by the relation hasSubOntologyConcept). Since taxonomic structure is the "... backbone of ontologies" [Euzenat/Shvaiko 2007], matching techniques based on sub and super concept relations have been studied by researchers frequently. The most common techniques are based on the counting of edges of the taxonomic graph. For instance, [Valtchev/Euzenat 1997] provide a method, named structural topological dissimilarity on hierarchies:

*Structural topological dissimilarity on hierarchies is defined as a dissimilarity over a taxonomy, such that ...*  
\[ \forall e, e' \in S, \delta (e, e') = \min (\text{plen}(e, e')) \]

... where \( \text{plen}(e, e') \), i.e. the path-length, refers to the number of intermediate edges between entities \( e \) and \( e' \).

**Definition 5.31 Structural topological dissimilarity (cf. [Valtchev/Euzenat 1997])**

The structural topological dissimilarity can be normalized by the maximal path of two concepts in the whole taxonomy, which results in the following formula.

\[ \forall e, e' \in H, \bar{\delta} (e, e') = \frac{\delta (e, e')}{\max (\text{plen}(x, x'))} \]

**Definition 5.32 Normalized structural topological dissimilarity [Valtchev/Euzenat 1997]**

Considering Figure 5.4, the normalized structural topological dissimilarity between Truck and RailwayCar is \( \bar{\delta} (\text{Truck}, \text{RailwayCar}) = 0.8 \), i.e. the shortest path between Truck and Railway is 4 and the maximum path of the whole taxonomy is 5 (e.g. from HeavyTruck to RailwayCar). All results of are shown in the following table.
Having a look on the results, the closest concepts are the concepts that have a direct parent or child relation. Moreover, one can conclude that a HeavyTruck is closer to an Aircraft (0.6) than to a Container (1.0). However, the results calculated for a single taxonomy. Comparing two taxonomies and measuring the structural topological dissimilarity between concepts of both is different. One has to find similar entities, first, in order to identify connections or “bridges” between both taxonomies. For example, taking the taxonomy from Figure 5.4 and another taxonomy called MeansOfTransport including one child node called Truck, which has two child nodes called NormalTruck and RoadTrain, one should identify that the concept Truck is the same in both taxonomies. Having this “bridge” one can calculate the normalized structural topological dissimilarity between MiniTruck and RoadTrain with 0.40 (= 2/5 considering that the longest path is still 5). Thus, considering two taxonomies, the quality of the structural topological dissimilarity results rely on a pre-step that identifies “bridges” between taxonomies.

A more elaborated measure of this kind is the Wu-Palmer similarity [Wu/Palmer 1994]. This measure takes into account that two entities near the root are very close to each other, but might differ a lot conceptually, while two entities which are far from the root and far away from each other might be conceptually similar. More formal, the Wu-Palmer similarity is defined as follows.

<table>
<thead>
<tr>
<th>Resource</th>
<th>VehicleResource</th>
<th>Truck</th>
<th>Ship</th>
<th>Aircraft</th>
<th>Mini-, Light-, Medium-, HeavyTruck</th>
<th>TransportationUnit</th>
<th>Container, SwapBody, Railcar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>0.00</td>
<td>0.20</td>
<td>0.40</td>
<td>0.40</td>
<td>0.60</td>
<td>0.20</td>
<td>0.40</td>
</tr>
<tr>
<td>VehicleResource</td>
<td>0.20</td>
<td>0.00</td>
<td>0.20</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
<td>0.60</td>
</tr>
<tr>
<td>Truck</td>
<td>0.40</td>
<td>0.20</td>
<td>0.00</td>
<td>0.40</td>
<td>0.40</td>
<td>0.20</td>
<td>0.80</td>
</tr>
<tr>
<td>Ship</td>
<td>0.40</td>
<td>0.20</td>
<td>0.40</td>
<td>0.00</td>
<td>0.40</td>
<td>0.60</td>
<td>0.40</td>
</tr>
<tr>
<td>Aircraft</td>
<td>0.40</td>
<td>0.20</td>
<td>0.40</td>
<td>0.40</td>
<td>0.00</td>
<td>0.60</td>
<td>0.40</td>
</tr>
<tr>
<td>Mini-, Light-, Medium-, HeavyTruck</td>
<td>0.60</td>
<td>0.40</td>
<td>0.20</td>
<td>0.60</td>
<td>0.60</td>
<td>0.40</td>
<td>0.80</td>
</tr>
<tr>
<td>TransportationUnit</td>
<td>0.20</td>
<td>0.40</td>
<td>0.60</td>
<td>0.60</td>
<td>0.60</td>
<td>0.80</td>
<td>0.00</td>
</tr>
<tr>
<td>Container, SwapBody, Railcar</td>
<td>0.40</td>
<td>0.60</td>
<td>0.80</td>
<td>0.80</td>
<td>0.80</td>
<td>1.00</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.21 Results of the normalized structural topological dissimilarity for the taxonomy in Figure 5.4

For saving space, the entities Mini-, Light-, Medium-, and HeavyTruck, and the entities Container, SwapBody, and RailwayCar are subsumed in one row/column each. There are two different values, i.e. 0 for the same entity (e.g. MinTruck and MinTruck) and 0.4 for two sibling entities (e.g. Min-Truck and MediumTruck).
The Wu-Palmer similarity is a similarity over a taxonomy, such that:

\[ \forall e, e' \in H, \sigma(e, e') = \frac{2 \cdot \text{plen}(\text{Iso}(e, e'), r)}{\text{plen}(e, \text{Iso}(e, e')) + \text{plen}(e', \text{Iso}(e, e')) + 2 \cdot \text{plen}(\text{Iso}(e, e'), r)} \]

... where \( r \) is the root of the taxonomy, \( \text{plen}(e, e') \), i.e. the path-length, is the number of intermediate edges between entities \( e \) and \( e' \), and \( \text{Iso}(e, e') \), i.e. the lowest super-ordinate, refers to the concept that subsumes \( e \) and \( e' \) semantically.

Definition 5.33 Wu-Palmer similarity (cf. [Wu/Palmer 1994])

For example, the Wu-Palmer similarity of Truck and RailwayCar in the taxonomy (cf. Figure 5.4) is \( \sigma(\text{Truck}, \text{RailwayCar}) = 0.33 = (2 \times 1)/(2 + 2 + 2 \times 1) \), since the lowest super-ordinate is Resource, the path-length of Resource to the root\(^35\) is 1, and the path-length from Truck to Resource and from RailwayCar to Resource is 2 both. Applying the Wu-Palmer similarity to the concepts of the taxonomy results in the following table.

<table>
<thead>
<tr>
<th></th>
<th>Resource</th>
<th>VehicleResource</th>
<th>Truck</th>
<th>Ship</th>
<th>Aircraft</th>
<th>Mini-, Light-, Medium-, HeavyTruck</th>
<th>TransportationUnit</th>
<th>Container, SwapBody, Railcar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>1.00</td>
<td>0.67</td>
<td>0.50</td>
<td>0.50</td>
<td>0.40</td>
<td>0.67</td>
<td>0.50</td>
<td></td>
</tr>
<tr>
<td>VehicleResource</td>
<td>0.67</td>
<td>1.00</td>
<td>0.80</td>
<td>0.80</td>
<td>0.67</td>
<td>0.50</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>Truck</td>
<td>0.40</td>
<td>0.80</td>
<td>1.00</td>
<td>0.67</td>
<td>0.67</td>
<td>0.86</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>Ship</td>
<td>0.50</td>
<td>0.80</td>
<td>0.67</td>
<td>1.00</td>
<td>0.67</td>
<td>0.67</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>Aircraft</td>
<td>0.50</td>
<td>0.80</td>
<td>0.67</td>
<td>0.67</td>
<td>1.00</td>
<td>0.60</td>
<td>0.40</td>
<td>0.33</td>
</tr>
<tr>
<td>Mini-, Light-, Medium-, HeavyTruck</td>
<td>0.40</td>
<td>0.67</td>
<td>0.86</td>
<td>0.57</td>
<td>0.60</td>
<td>1.00</td>
<td>0.33</td>
<td>0.28</td>
</tr>
<tr>
<td>TransportationUnit</td>
<td>0.67</td>
<td>0.50</td>
<td>0.40</td>
<td>0.40</td>
<td>0.40</td>
<td>0.33</td>
<td>1.00</td>
<td>0.80</td>
</tr>
<tr>
<td>Container, SwapBody, Railcar</td>
<td>0.50</td>
<td>0.40</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>0.28</td>
<td>0.80</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5.22 Results of the normalized Wu-Palmer similarity for the taxonomy in Figure 5.4\(^{36}\)

Another similarity measure is the upward cotopic similarity [Maedche/Zacharias 2002], which applies the Jaccard similarity to cotopics, i.e. super concepts. It is defined as follows:

---

\(^{35}\) Resource itself is not the root of the taxonomy shown in Figure 5.4. The root element is the super-concept of Resource.

\(^{36}\) For saving space, the entities Mini-, Light-, Medium-, and HeavyTruck, and the entities Container, SwapBody, and RailtrainCar are subsumed in one row/column each. There are two values, one value for the same entity (e.g. MinTruck and MinTruck) and one value considering two sibling entities (e.g. MinTruck and MediumTruck).
The upward cotopic similarity \( \sigma \) is a similarity over a hierarchy \( H \) such that:

\[
\forall e, e' \in H, \sigma(e, e') = \frac{|UC(e, H) \cap UC(e', H)|}{|UC(e, H) \cup UC(e', H)|}
\]

... where \( UC(e, H) \) is the set of superconcepts of \( e \).

**Definition 5.34** Upward cotopic similarity (cf. [Maedche/Zacharias 2002])

Applied to the example of Truck and RailwayCar in taxonomy (cf. Figure 5.4), the upward cotopic similarity \( \sigma(\text{Truck, RailwayCar}) = 0.33 = 2/6 \), because \( UC(\text{Truck}, H) = \{\text{root}, \text{Resource}, \text{VehicleResource}\} \) and \( UC(\text{RailwayCar}, H) = \{\text{root}, \text{Resouce}, \text{TransportationUnit}\} \). Furthermore, the following table shows the results of all comparing all entities.

<table>
<thead>
<tr>
<th>Resource</th>
<th>VehicleResource</th>
<th>Truck</th>
<th>Ship</th>
<th>Aircraft</th>
<th>Mini, Light, Medium, HeavyTruck</th>
<th>TransportationUnit</th>
<th>Container, SwapBody, RailwayCar</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource</td>
<td>1.00</td>
<td>0.33</td>
<td>0.25</td>
<td>0.25</td>
<td>0.20</td>
<td>0.33</td>
<td>0.25</td>
</tr>
<tr>
<td>VehicleResource</td>
<td>0.33</td>
<td>1.00</td>
<td>0.40</td>
<td>0.40</td>
<td>0.33</td>
<td>0.50</td>
<td>0.40</td>
</tr>
<tr>
<td>Truck</td>
<td>0.25</td>
<td>0.40</td>
<td>1.00</td>
<td>0.50</td>
<td>0.43</td>
<td>0.43</td>
<td>0.33</td>
</tr>
<tr>
<td>Ship</td>
<td>0.25</td>
<td>0.40</td>
<td>0.50</td>
<td>1.00</td>
<td>0.43</td>
<td>0.43</td>
<td>0.33</td>
</tr>
<tr>
<td>Aircraft</td>
<td>0.25</td>
<td>0.40</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
<td>0.43</td>
<td>0.33</td>
</tr>
<tr>
<td>Mini-, Light-, Medium-, HeavyTruck</td>
<td>0.20</td>
<td>0.33</td>
<td>0.43</td>
<td>0.43</td>
<td>0.33</td>
<td>1.00</td>
<td>0.33</td>
</tr>
<tr>
<td>TransportationUnit</td>
<td>0.33</td>
<td>0.50</td>
<td>0.40</td>
<td>0.40</td>
<td>0.33</td>
<td>1.00</td>
<td>0.40</td>
</tr>
<tr>
<td>Container, SwapBody, RailwayCar</td>
<td>0.25</td>
<td>0.40</td>
<td>0.33</td>
<td>0.33</td>
<td>0.28</td>
<td>0.40</td>
<td>0.33</td>
</tr>
</tbody>
</table>

**Table 5.23** Results of the upward cotopic similarity for the taxonomy in Figure 5.4\(^{37}\)

As mentioned previously, in context of service model matching taxonomy structure techniques cannot be applied as they are, since service model contain taxonomic parts, but one can neither assume that service model share the same taxonomic parts nor even the same taxonomy. However, one may identify connections (i.e. "bridges") between the taxonomic parts of service model and can apply taxonomy structure techniques afterwards. The identification of connections can be done manually or can rely on a string-based or language-based similarity measure.

\(^{37}\) For saving space, the entities Mini-, Light-, Medium-, and HeavyTruck, and the entities Container, SwapBody, and RailtrainCar are subsumed in one row/column each. There are two values, one value for the same entity (e.g. MinTruck and MinTruck) and one value considering two sibling entities (e.g. MinTruck and MediumTruck).
Applying taxonomic structure techniques for service matching, one can conclude that concepts are similar if their sub or super concepts are similar. For instance, if sub concepts are the same the concepts are similar and if two concepts share the same super concept then both are similar as well [Dieng/Hug 1998]. However, this cannot by applied without care. One has to be sure that super concepts aren’t in a homonym relation, have the same meaning, and share the same domain. Otherwise, sub and super concepts might be considered to be the same, but they are not. For example, Truck and Ship share the super concept VehicleResource. If Truck has another super concept RoadVehicle one may conclude that VehicleResource and RoadVehicle are the same. Or even worse, one may conclude that Ship is a sub concept of RoadVehicle.

Another challenge of taxonomic structure techniques is that similarity measures between super or sub concepts rely on their super or sub concepts, in turn. Hence, one has to define an approach that calculates concept similarities in iteration. Iteration definition must have a starting point and a point of termination. Most importantly, by the definition of termination, one has to take care that iterations do not contain any infinite loops.

Next techniques based on mereologic structure are examined briefly.

**Mereologic structure**

In contrast to the taxonomic structure, which is based on sub and super concepts relations, the mereologic structure corresponds to is-part-of relations. The taxonomic structure is well established in various ontology language, e.g. in OWL/RDF this is defined by the subClassOf property. Instead, the part-of relation is not defined in ontology languages. Thus properties that bear a mereologic semantic has to be detected before applying similarity measures. However, the detection of part-of relations might be hard, depend on the ontology and cannot be generalized. Anyway, one might detect mereologic properties manually or predefined properties that bear part-of relations upfront, i.e. before ontology creation. Typical mereologic properties are, for instance, ‘contains’, ‘hasXYZ’ or ‘is-part-of’. If mereologic properties are predefined or detected successfully, these can be used for computing similarities. Analog to similarity measures base on taxonomic relations, one can conclude that concepts are similar if sharing similar parts. Similar to OWL/RDF, the SCMM does not predefine a mereologic relation. So, mereologic relations are treated as any other relations, which is examined next.

**Other relational structures**

Beside taxonomic relations and mereologic relation one can consider matching concepts based on their relations in general, i.e. properties and attributes they are connected with. However, in contrast to taxonomic and mereologic relations, general relation may contain circuits. These circuits are challenges in ontology matching and, thus, in service matching as well. Circuits will be addressed in detail later on (cf. 5.4.3 Global similarity computation).
Neglecting the circulation issues, [Maedche/Staab 2002] suggests that concepts are similar if they are connected by the same property. For example (cf. Figure 5.5), the concept TransportOrder has two properties called hasSource and hasDestination. hasSource is a connection to the concepts Consigner and hasDestination to the concepts Consignee. On the other hand there is the ShipmentRequest concept that has the properties hasSource and hasDestination as well. However, both properties point to the concept Address. Considering TransportOrder and ShipmentRequest are the same, Consigner and Address are similar and the concepts Consignee and Address are similar as well, since both are connected via the same property (i.e. hasSource and hasDestination respectively).

![Figure 5.5 Similarity based on relational structure (example)](image)

Same relations might indicate concept similarities. Further details can be taken from [Maedche/Staab 2002].

**Summary on relational structure techniques**

Matching services based on their relational structure is very powerful. However, relations can be considered only, if they are grounded on other tangible properties. For example, the assertion about the similarity between Address and Consigner (and Consignee respectively) cannot be made if the properties (hasSource and hasDestination) and the concepts TransportOrder and ShipmentRequest aren’t considered to be the same (cf. Figure 5.5). Thus, relational structures are usually used in combination with string-based, language-based and internal structure techniques.

Before applying techniques for relational structures one has to consider matching relevant relation types. Very common is the use of taxonomic relations, since they are widely used in ontology matching. Mereologic structures might be important as well. However, in contrary to taxonomic relations, mereologic relations can be expressed by different names. Thus, mereologic relations have to be detected before applying mereologic techniques. Considering relational structures in general raises the problem of which entity influences another entity, i.e. there are usually mutual influences between related parts. Therefore, it is important to associate general relational structure techniques with other techniques. Furthermore, it is necessary to establish strategies that resolve circulations (cf. 5.4.3 Global similarity computation).
5.3.3 Extensional techniques

Beside name-based techniques and structure-based techniques there are extensional techniques. Extensional techniques refer to similarity measures based on individuals or instances. Instances can facilitate matching concepts, i.e. when two concepts share exactly the same instance data these concepts are obviously the same. For instance, if the concept Address and the concepts Consignee share the same set of instances, e.g. the same name, street, zip, city and country, one can conclude that these concepts are the same. However, in case of BS and ES matching there is no instance data involved, since it is hard to collect all possible ES instances for matching possible BS instances. For instance, let’s assume that Address is an ES OntologyConcept and Consignee is a BS OntologyConcept, then one has to collect all Address and Consignee data of the world in order do find similarities for sure. This is not a realistic. Thus, extensional techniques are out of scope here.

5.3.4 Summary on basic matching algorithms

Basic matching techniques, classified by name-based techniques, structure-based techniques, and extensional techniques, have been discussed so far. However, it is beyond the scope of this thesis presenting all existing techniques. Instead, the examination above considers the most common ones. As shown, none of these techniques is perfect. There is still much work in finding better matching solutions.

As observed, most of the techniques cannot be used in isolation. Instead, each of the techniques may take advantage of the results provided by another technique. Thus, an important part of service matching relies on selecting and combining techniques in a proper way. This topic is addressed by matching strategies, which will be examined in the following.

5.4 Matching strategies

Most of basic matching algorithms cannot be used in isolation, but each of them provides a useful contribution in combination. Thus, matching strategies address the composition of matching algorithms. Basically, the following topics are involved (cf. [Euzenat/Shvaiko 2007]):

- Composing and sequencing matching algorithms
- Aggregation of matching results
- Global similarity computation and handling circulations

38 Among other, one evidence for that is the annually International Workshop on Ontology Matching (cf. [Shvaiko et al. 2011]).
• Deriving missing correspondence measures by considering probabilistic methods.
• Involving the user, i.e. manual steps, in the matching process.
• Considering learning methods for continues matching result improvement.

In this chapter, composition of matching results and sequencing of matching algorithms are considered, first (cf. 5.4.1). Next, techniques for aggregation are addressed (cf. 5.4.2). Last not least, there is a section about computing global similarities and handling circularities (cf. 5.4.3). Manual steps, probabilistic methods, and learning methods are not considered in this thesis (cf. future work section 9.4).

5.4.1 Matching composition and sequencing

The goal of matching strategies is combining local similarity measure gained from basic methods. The combination, also known as composition, of matching algorithms results in a new algorithm [Euzenat/Shvaiko 2007]. Local similarity measures address one single matching process only (cf. Figure 5.1). For improving a single similarity measure the most natural way is applying a second matching process after the first one. This type of composition is called sequential composition (cf. Figure 5.6).

As shown in the figure above, there are two matching processes fulfilled in sequential order. Both matching processes have different parameters and resources, but have the same service models (sm_{BS} and sm_{ES}) as the input. The alignment A', which is the output of the first matching process, is used as an input of the second one. Beside the sequential composition with two matching processes, a composition of three or more matching processes is possible. For instance, one would like to use a matching algorithm based on strings first, before running a matching algorithm based on the internal structure of entities and applying an algorithm based on the relational structure at the end.

There are several basic similarity measure techniques that are based on matrix representation. Thus, a matrix has to be extracted from given service models and, if available, a given alignments first. Afterwards similarities can be computed for
each matrix cell. Lastly, these values have to be extracted in order to gain proper alignment results. Both extraction processes have to be considered, and, thus, are added into the matching process chain, which leads to the following illustration (see Figure 5.7).

![Diagram](image)

*Figure 5.7 The matching process including similarity matrix and alignment extraction*

As shown in the figure above, matrix extraction comprises building an initial matrix $M$ from two service models $sm_{BS}$ and $sm_{ES}$ (considering alignment $A$, if available). Next, similarity computation produces a similarity matrix $M'$ from the initial Matrix $M$. And finally, alignment $A'$ is extracted from the Matrix $M'$. Next, the sequential composition of the matching including similarity matrices is shown in Figure 5.8.

![Diagram](image)

*Figure 5.8. Sequential composition of matching processes though similarity matrix*

Another way of combining algorithms is by running them independently. This is called parallel composition as illustrated in Figure 5.9.
Running matching algorithms independently, i.e. in parallel, results in two or more alignment results (cf. Figure 5.9, $A'$ and $A''$) which have to be aggregated to one single results (cf. Figure 5.9, $A'''$). The aggregation can be very different: Aggregation can be based on choosing one of the results based on certain criteria (e.g. the correspondence with the highest confidence) or it can be based on merging the results through some predefined operation (e.g. mean calculation). In case of merging results through some operation it is convenient to use similarity matrices and apply mathematical operations. Thus, Figure 5.10 shows the parallel composition chain based on matrices.

Basically there are two kinds of parallel compositions: Heterogeneous and homogeneous parallel composition. Within the former, the input is fragmented into different kinds of data, such as strings, graphs, etc., similarity values are measured and the aggregation step aggregates values or selects the most promising one only. Within later, i.e. homogeneous parallel composition, instead, the whole input is given to several different matching systems and aggregating there results or just select the best one.

All composition techniques have to be implemented within particular matching algorithms. The selection of matching algorithms and composition techniques
depends on the use case, i.e. on the service model, existing input parameters and resources as well as the application area.

5.4.2 Similarity aggregation

Compound similarities are concerned with the aggregation of heterogeneous similarity measures. As explained in 5.3.2.2 and suggested by Table 5.20 concepts have often different labels, names, description, relations etc. When computing the similarity between each of these entities the similarity values of entity features have to be measured and aggregated in order to provide a similarity assessment between the two particular entities. For instance, two entities contain a name, several labels, a description, data properties, objects properties, super and sub concepts and other relations. Thus, in order to assess about the (compound) similarity between two entities, one has to measure each of these features separately and have to aggregate the results. This section examines aggregation techniques.

5.4.2.1 Simple aggregation techniques

A simple ways of aggregation is by selecting the maximum similarity measure from two independent similarity computation results. However, it is important to note that the similarity measures have to be normalized prior. More formal the aggregation between values of two similarity measures is defined as follows.

Let $M'$ and $M''$ be two matrices containing the normalized similarity results $\sigma'(e_i, e_j)$ and $\sigma''(e_i, e_j)$ from two independent similarity computations, the maximum similarity aggregation measure $\sigma$ is defined as follows:

$$\sigma\left(\sigma'(e_i, e_j), \sigma''(e_i, e_j)\right) = \max\left(\sigma'(e_i, e_j), \sigma''(e_i, e_j)\right)$$

**Definition 5.35 Maximum similarity aggregation**

This function can be applied as a min-function for dissimilarity (or distance) measures. The drawback of maximum and minimum functions is that the best values are selected only. This aggregation would not consider if there are huge differences between $M'$ and $M''$. However, the advantage of such an aggregation function lays in its simplicity. Another simple aggregation function is the weight product aggregation, which is defined as follows:

Let $M_1, M_2, \ldots, M_l$ be $x$ matrices containing the normalized similarity results $\sigma_1(e_i, e_j), \sigma_2(e_i, e_j), \ldots, \sigma_l(e_i, e_j)$ from $n$ independent similarity computations, and $w_i$ is the weight (interval $[0…1]$), the weight product aggregation function $\sigma$ is defined:

$$\sigma\left(\sigma_1(e_i, e_j), \sigma_2(e_i, e_j), \ldots, \sigma_l(e_i, e_j)\right) = \prod_{x=1}^{n} \sigma_x(e_i, e_j)^{w_i}$$

**Definition 5.36 Weight product aggregation**
For example, considering the normalized Levenshtein similarity from Table 5.5 and the token based cosine similarity (including domain filter and suffix stemming) from Table 5.10 the maximum aggregation function and the weight product aggregation results are shown in Table 5.24 and Table 5.25 respectively.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>zen</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.71</td>
<td>0.53</td>
<td>0.53</td>
<td>0.35</td>
<td>0.47</td>
<td>0.59</td>
</tr>
<tr>
<td>transport order</td>
<td>0.71</td>
<td>0.50</td>
<td>0.41</td>
<td>0.29</td>
<td>0.24</td>
<td>0.24</td>
<td>0.24</td>
</tr>
<tr>
<td>source address</td>
<td>0.29</td>
<td>0.35</td>
<td>0.29</td>
<td>0.29</td>
<td>0.24</td>
<td>0.24</td>
<td>0.29</td>
</tr>
<tr>
<td>destination address</td>
<td>0.12</td>
<td>0.18</td>
<td>0.12</td>
<td>0.12</td>
<td>0.06</td>
<td>0.06</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.41</td>
<td>0.24</td>
<td>0.59</td>
<td>0.47</td>
<td>1.00</td>
<td>0.53</td>
<td>0.71</td>
</tr>
<tr>
<td>good</td>
<td>0.53</td>
<td>0.06</td>
<td>0.59</td>
<td>0.53</td>
<td>0.35</td>
<td>0.76</td>
<td>0.70</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.71</td>
<td>0.50</td>
<td>0.47</td>
<td>0.35</td>
<td>0.50</td>
<td>0.35</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 5.24 Maximum aggregation

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>zen</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.36</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>transport order</td>
<td>0.68</td>
<td>0.30</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>source address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>destination address</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.84</td>
<td>0.00</td>
<td>0.54</td>
</tr>
<tr>
<td>good</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.61</td>
<td>0.23</td>
<td>0.00</td>
<td>0.00</td>
<td>0.48</td>
<td>0.00</td>
<td>0.68</td>
</tr>
</tbody>
</table>

Table 5.25 Weight product aggregation with $w_1 = w_2 = 0.5$

In case of weight product aggregation the weight $w_1 = w_2 = 0.5$ is choose as an example. However, the weight plays an important role, since one can decide about the importance of the similarity measures. In case of $w_1 = w_2 = 0.5$ both similarity measure results have the same importance (except of zero values, see below). This is equal to geometric mean measurement.

In both cases, i.e. maximum aggregation and weight product aggregation, the algorithms tend to supplement one value by another one. For example, in maximum aggregation all zero values from token base cosine similarity are supplemented by the values greater than zero from Levenshtein similarity in most of the cases. Instead, in case of weight product aggregation all zero from Levenshtein
similarity persist as a zero value, since the multiplication with zero remains zero. This can be an advantage and a drawback at the same time. On the one hand, one can mask values that have zero value in one similarity measure, on the other hand, if there is only one zero value (from x values) all other values do not count since the multiplication with zero remains zero (even if the weight of the zero value is very low, e.g. 0.00001). Thus, there are other methods that cope with the drawbacks of maximum and weight product aggregation, which are examined next.

5.4.2.2 Multidimensional aggregation techniques

Simple aggregation functions tend to imply dependencies between values of different dimensions, so that a value of one dimension can override a value of another dimension. Instead, multidimensional aggregation techniques balance the values between dimensions and are thus better suited for independent dimensions. The Minkowski distance [Deza/Deza 2009] is a representative of multidimensional aggregation. Minkowski distance is defined as follows:

\[
\delta \left( \delta_1(e_i, e_j), \delta_2(e_i, e_j), ..., \delta_n(e_i, e_j) \right) = \sqrt[p]{\sum_{i=1}^{n} \delta_i(e_i, e_j)^p}
\]

**Definition 5.37** Minkowski distance aggregation (cf. [Deza/Deza 2009])

Minkowski distance is typically used the p=1 or p=2. The former is called Manhattan distance [Krause 1986] the latter Euclidean distance [Deza/Deza 2009]. A special case is p=+∞, which is called Chebyshev distance [Cantrell 2000] and is neglected here. Manhattan and Euclidean distance are focused only. Since normalized distance values are assumed, all distance values can be calculated as similarity values as well. This is done by using the formula \( \delta(e_i, e_j) = 1 - \sigma(e_i, e_j) \). Considering the maximum aggregation result (from normalized Levenshtein similarity and the token based cosine similarity) from Table 5.24 and the gloss overlap similarity results from Table 5.11, the normalized Manhattan and Euclidean distance results (represented as similarity) are shown below. In case of calculating the Manhattan distance and the Euclidean distance, normalization is fulfilled by dividing all results with the maximum possible value. The maximum possible value is 2 for Manhattan distance and 1.41 for Euclidean distance. The results are shown in Table 5.26 and Table 5.27.
<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>Consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.40</td>
<td>0.32</td>
<td>0.27</td>
<td>0.18</td>
<td>0.24</td>
<td>0.30</td>
</tr>
<tr>
<td>transport order</td>
<td>0.55</td>
<td>0.30</td>
<td>0.23</td>
<td>0.15</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>source address</td>
<td>0.15</td>
<td>0.18</td>
<td>0.15</td>
<td>0.15</td>
<td>0.12</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>destination address</td>
<td>0.06</td>
<td>0.09</td>
<td>0.06</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.21</td>
<td>0.12</td>
<td>0.30</td>
<td>0.24</td>
<td>0.59</td>
<td>0.27</td>
<td>0.37</td>
</tr>
<tr>
<td>good</td>
<td>0.35</td>
<td>0.12</td>
<td>0.25</td>
<td>0.42</td>
<td>0.18</td>
<td>0.38</td>
<td>0.05</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.38</td>
<td>0.34</td>
<td>0.27</td>
<td>0.18</td>
<td>0.26</td>
<td>0.18</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.26 Normalized Manhattan distance as a similarity based on dimension mentioned above

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>Consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.32</td>
<td>0.29</td>
<td>0.22</td>
<td>0.15</td>
<td>0.20</td>
<td>0.23</td>
</tr>
<tr>
<td>transport order</td>
<td>0.51</td>
<td>0.27</td>
<td>0.21</td>
<td>0.13</td>
<td>0.11</td>
<td>0.14</td>
<td>0.11</td>
</tr>
<tr>
<td>source address</td>
<td>0.13</td>
<td>0.15</td>
<td>0.13</td>
<td>0.13</td>
<td>0.11</td>
<td>0.14</td>
<td>0.13</td>
</tr>
<tr>
<td>destination address</td>
<td>0.06</td>
<td>0.08</td>
<td>0.06</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.18</td>
<td>0.11</td>
<td>0.23</td>
<td>0.20</td>
<td>0.42</td>
<td>0.22</td>
<td>0.28</td>
</tr>
<tr>
<td>good</td>
<td>0.32</td>
<td>0.11</td>
<td>0.39</td>
<td>0.22</td>
<td>0.15</td>
<td>0.27</td>
<td>0.26</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.30</td>
<td>0.32</td>
<td>0.23</td>
<td>0.15</td>
<td>0.22</td>
<td>0.15</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Table 5.27 Normalized Euclidean distance as a similarity based on dimension mentioned above

The values between Manhattan distance and Euclidean distance are very close to each other. However, the values of Manhattan distance tend to be higher than the Euclidean ones, because of p.

Minkowski distances can be weighted in order to give more importance to one of the dimension. For instance, the similarity on labels including a token distance based on an external linguistic resource, can be more important as a gloss overlap based on comments. Thus, one may add weights to the aggregation function in order to emphasize the importance of certain dimensions and weighting less important ones correspondingly. The function will thus use a set of weight w₁, w₂, w₃, … wₙ, corresponding to the dimension x. Applied to Manhattan distance the weight sum is defined as follows:
Let \( M_1, M_2, \ldots, M_x \) be \( x \) matrices containing the normalized distance results \( \delta_1(e_i, e_j), \delta_2(e_i, e_j), \ldots, \delta_n(e_i, e_j) \) from \( n \) dimension, the Manhattan distance \( \delta \) including weight sum is defined as follows:

\[
\delta \left( \delta_1(e_i, e_j), \delta_2(e_i, e_j), \ldots, \delta_n(e_i, e_j) \right) = \sum_{x=1}^{n} w_x \times \delta_x(e_i, e_j)
\]

... with \( w_i \in [0...1] \) and \( \sum_{x=1}^{n} w_i = 1 \)

**Definition 5.38 Manhattan distance aggregation**

For example, it appears that measures on labels (i.e. labels including token distance based on external linguistic resource) are more accurate than those on comments (i.e. gloss overlap measurement). Thus weighting this dimension in a ratio \( w_{\text{label}} = 3/4 \) and \( w_{\text{comments}} = 1/4 \) using the weight sum aggregation results in the following table.

<table>
<thead>
<tr>
<th></th>
<th>transport</th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport</td>
<td>1.00</td>
<td>0.40</td>
<td>0.32</td>
<td>0.27</td>
<td>0.18</td>
<td>0.24</td>
<td>0.30</td>
</tr>
<tr>
<td>transport order</td>
<td>0.55</td>
<td>0.30</td>
<td>0.23</td>
<td>0.15</td>
<td>0.12</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>source address</td>
<td>0.15</td>
<td>0.18</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>destination address</td>
<td>0.06</td>
<td>0.09</td>
<td>0.06</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
<td>0.00</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.21</td>
<td>0.12</td>
<td>0.30</td>
<td>0.24</td>
<td>0.59</td>
<td>0.27</td>
<td>0.37</td>
</tr>
<tr>
<td>good</td>
<td>0.35</td>
<td>0.12</td>
<td>0.25</td>
<td>0.42</td>
<td>0.18</td>
<td>0.38</td>
<td>0.035</td>
</tr>
<tr>
<td>truck cargo</td>
<td>0.38</td>
<td>0.34</td>
<td>0.27</td>
<td>0.18</td>
<td>0.26</td>
<td>0.18</td>
<td>0.75</td>
</tr>
</tbody>
</table>

**Table 5.28 Normalized weight sum based on manhattan distance (as a similarity) with \( w_{\text{label}} = 3/4 \) and \( w_{\text{comments}} = 1/4 \)**

Beside the mentioned aggregation techniques there are others, such as fuzzy aggregation [Gal et al. 2005] and ordered weighted average [Yager 1988]. However, details about them can be taken from literature. Next global similarity computation techniques are examined in order to address the issues of circulation, which was mentioned previously.

### 5.4.3 Global similarity computation

The compound similarity computation is still of local manner, since the aggregation similarities consider nodes and their neighbors only. Instead, a global similarity considers service models as a whole, where similarity measures depend on all service model entities. An issue mentioned previously is that local similarity
measure may bear circularities though dependencies of relations. Both cases require special techniques for similarity computation.

In Figure 5.11, there is an example showing two service model parts with referential cycles. Both models are simplified, i.e. property names and cardinalities are skipped due to lack of space. The part on the left hand side contains the cycle relations between the concepts TransportOrder and SourceAddress, TransportOrder and DestinationAddress, TransportOrder and Good, SourceAddress and PickupTime. These relations might be termed, for instance, as hasGood (from TransportOrder to Good) and hasTransportOrder (from Good to TransportOrder). However, relation names are neglected here. Existence of cyclic relations make local similarity measure techniques unfeasible.

![Diagram showing two typical ontologies showing referential cycles](image)

**Figure 5.11 Two typical ontologies showing referential cycles (simplified, i.e. no relation names due lack of space)**

An intuitive way dealing with circular dependencies involves iteration, i.e. aggregation in an iterative way by refining the last computed value at each step. Iterative computation is illustrated in Figure 5.12.

![Diagram showing iterative computation of similarity function](image)

**Figure 5.12 Iterative computation of similarity function**

There are two methods that cope with circularities. The first one, i.e. similarity flooding, describes a process for similarity computation within a graph. The second method, i.e. similarity equation fixed point, transforms similarity definitions in a set of equations first and uses numerical analysis techniques for solving these equations. Both methods are examined in the following.
5.4.3.1 Similarity flooding

Similarity flooding [Melnik et al. 2002] proposes a generic graph matching algorithm which determines corresponding nodes in a graph. The algorithm is based on the assumption that two nodes of a graph have to be similar if their adjacent nodes tend to be similar. The implementation of the algorithm proceeds as follows:

1. Pre-stage: The ontology has to be transformed into a directed graph G, where the nodes are the concepts and the properties (or relations respectively) are the edges of the graph.
2. Weighting: Assigning weights w to the edges. These weights are usually \(1/n\), where \(n\) is the number of outcoming edges of the source node.
3. Initiation: Assign an initial similarity \(\sigma_0\) to each node pair, e.g. gained from a basic matching algorithm.
4. Computation: Compute \(\sigma_{i+1}\) for each node pair with a chosen formula.
5. Normalization: Normalize \(\sigma_{i+1}\) by dividing with the maximum value.
6. Termination/Iteration: Stop after a predetermined number of steps (or if there is no similarity change more than a predefined threshold) or, otherwise, go to step 4.

The formula mentioned in step 4 suggested by [Melnik et al. 2002] is a weight linear aggregation function. This function calculates the next value of the iteration \(\sigma^{i+1}(x,x')\) considering the initial value \(\sigma^0(x,x')\) and the sum of each related edges inverse weight \(w(x,x',y,y')\) multiplied by the edge value the step before \(\sigma^i(y,y')\). This results in the following definition:

Let \(x\) and \(x'\) be two nodes of two graphs \(G\) and \(G'\), and \(y_k\) all nodes connected with \(x\) directly (from \(k=1\) to \(K\)) and \(y'_l\) all nodes connected with \(x'\) directly (from \(l=1\) to \(L\)). Furthermore, let \(w(x,x',y_k,y'_l)\) be the inverse weight of the number of outcoming edges considering \(x, x'\), and \(y_k, y'_l\), and \(\sigma^i(y_k,y'_l)\) the value of \(y_k\) and \(y'_l\) of the previous iteration, the weight linear aggregation function is defined as follows:

\[
\sigma^{i+1}(x,x') = \sigma^0(x,x') + \sum_{k=1}^{K} \sum_{l=1}^{L} w(x,x',y_k,y'_l) \times \sigma^i(y_k,y'_l)
\]

**Definition 5.39** Weight linear aggregation function for similarity flooding

For example, taking the two service model from Figure 5.11 and considering each concept as a node and each relation as an edge there are two directed graph G and G’ (step 1). Actually, the original similarity flooding algorithms works on properties with same names only. However, since property names can vary heavily and, thus, it is possible that there are no equal properties at all, it is assumed that all properties have the same name in this example. Before weighting (step 2) one has to consider that each node \(x\) of G can be combined with each node \(x'\) of G’. Since Figure 5.11 shows 5 nodes for each graph, there are 25 possible combinations.
Each of the two nodes of a node combination $(x, x')$ has one or many edges leading nodes that form related node combinations $(y_k, y'_l)$. For example, considering Figure 5.11 there is the node combination "TransportOrder-ShipmentRequest". "TransportOrder" has a direct connection to "SourceAddress", "DestinationAddress", and "Good", and "ShipmentRequest" has a direct connection to "Customer", "Consignee" and "Item". Thus $(x, x') = ["TransportOrder-ShipmentRequest"]$ has $3 \times 3 = 9$ related node combinations $(y_k, y'_l) = ["SourceAddress-Customer", "SourceAddress-Consignee", "SourceAddress-Item", "DestinationAddress-Customer", "DestinationAddress-Consignee", "DestinationAddress-Item", "Good-Customer", "Good-Consignee", "Good-Item"]$, which leads to a weight of $1/9$ for each combination. This and all possible combinations including all weights are illustrated in Figure 5.13.

Figure 5.13 Possible node-edge combinations including weights for similarity flooding algorithm

[Melnik et al. 2002] does not describe how to deal with several edges linking the same pair of nodes. However, one can imagine to aggregate weights using the weight sum algorithms, for instance. For initiation (step 3) the initial similarity $\sigma^0$
is taken from the normalized weight sum based on Manhattan distance as a similarity with \( w_{\text{label}} = \frac{3}{4}, w_{\text{comments}} = \frac{1}{4} \) shown in Table 5.28. The application of the formula mentioned above (step 4) to the node combination "TransportOrder-ShipmentRequest" leads to the following result:

\[
\sigma^1(x, x') = 0.30 + \left(\frac{0.15}{4} + \frac{0.15}{2} + \frac{0.12}{2} + \frac{0.06}{1} + \frac{0.06}{1} + \frac{0.03}{2} + \frac{0.25}{2} + \frac{0.42}{1} + \frac{0.30}{1}\right) = 1.52
\]

\( \sigma^1 \) for all combinations \((x, x')\) results in Table 5.29.

<table>
<thead>
<tr>
<th></th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport order</td>
<td>1.52</td>
<td>0.62</td>
<td>0.25</td>
<td>0.31</td>
<td>0.22</td>
</tr>
<tr>
<td>source address</td>
<td>0.97</td>
<td>0.85</td>
<td>0.22</td>
<td>0.34</td>
<td>0.19</td>
</tr>
<tr>
<td>destination address</td>
<td>0.22</td>
<td>0.13</td>
<td>0.09</td>
<td>0.07</td>
<td>0.06</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.29</td>
<td>0.39</td>
<td>0.27</td>
<td>0.63</td>
<td>0.30</td>
</tr>
<tr>
<td>good</td>
<td>0.25</td>
<td>0.32</td>
<td>0.45</td>
<td>0.22</td>
<td>0.41</td>
</tr>
</tbody>
</table>

*Table 5.29* Similarity flooding first iteration (\( \sigma^1 \))

Normalization (step 5) is fulfilled by dividing each value by the maximum value of the matrix, which lead Table 5.30.

<table>
<thead>
<tr>
<th></th>
<th>shipment request</th>
<th>customer</th>
<th>consignee</th>
<th>pickup period</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport order</td>
<td>1.00</td>
<td>0.41</td>
<td>0.16</td>
<td>0.21</td>
<td>0.14</td>
</tr>
<tr>
<td>source address</td>
<td>0.64</td>
<td>0.56</td>
<td>0.15</td>
<td>0.22</td>
<td>0.13</td>
</tr>
<tr>
<td>destination address</td>
<td>0.14</td>
<td>0.09</td>
<td>0.06</td>
<td>0.05</td>
<td>0.04</td>
</tr>
<tr>
<td>pickup time</td>
<td>0.19</td>
<td>0.26</td>
<td>0.18</td>
<td>0.41</td>
<td>0.20</td>
</tr>
<tr>
<td>good</td>
<td>0.16</td>
<td>0.21</td>
<td>0.30</td>
<td>0.14</td>
<td>0.27</td>
</tr>
</tbody>
</table>

*Table 5.30* \( \sigma^2 \) similarity flooding (normalized)

Step 6 leads to step 4 again. For example, it is predefined that 20 iteration steps have to be fulfilled in order to terminate the algorithm. After the 20. iteration the following normalized result set appears.
One can extract the expected correspondences from these similarity values.

### 5.4.3.2 Similarity equation

Similarity equation measures the similarity considering multiple features of entities. Basically, each feature of an entity is measured by a distinct similarity function. Afterwards, all feature measures are summarized and weighted. Given two entities $e$ and $e'$, basic structure of similarity equation is:

$$\sigma(e, e') = w_{f_1}\sigma_{f_1}(F_1(e), F_1(e')) + w_{f_2}\sigma_{f_2}(F_2(e), F_2(e')) + \ldots$$

$$+ w_{f_n}\sigma_{f_n}(F_n(e), F_n(e'))$$

Each feature of an entity is measure by a distinct similarity function $\sigma_f$. The similarity function gets the values of each feature of entities $e$ and $e'$ as an input. Values of features are the outcome of the functions $F(e)$ and $F(e')$ respectively. Further, each feature similarity function is weighted by a weight $w_f$. The summation of all weights $w_{f_1} + w_{f_2} + \ldots + w_{f_n} = 1$, which ensures a normalized similarity values, i.e. $\sigma(e, e') \in [0, 1]$.

Taking Figure 5.11 as an example, there are several concept. Each concept has a label and one or many properties. So, measuring the similarity of the concepts $c$ and $c'$ using similarity equation leads to the following formula.

$$\sigma^c(c, c') = w_{l_c}\sigma^c_{l_c}(F_l(c), F_l(c')) + w_{p_c}\sigma^c_{p_c}(F_p(c), F_p(c'))$$

The function $F_l(c)$ returns the label for a particular concepts. So, $\sigma^c_l$ returns the similarity value of the labels of $c$ and $c'$. Further, $F_p(c)$ returns the relations of a particular concept. However, since one concept may have more than one property the similarity measure of concept properties $\sigma^c_p$ should aggregated. This can be done by an aggregation function as proposed in chapter 5.4.2. Last not least there are two variables $w_{l_c}$ and $w_{p_c}$, which weight label and property similarity values.
Further, each property has a label as well as a domain and a range. Labels denote properties. Domain and range are adopted from ontologies, where domains are concepts a property is attached to, and ranges are concepts a property points to (cf. [Noy/McGuinness 2001]). The features label, domain and range lead to the following concept property similarity function.

$$\sigma^p(p, p') = w^p_L \sigma^p_L(F_L(p), F_L(p')) + w^p_D \sigma^p_D(F_D(p), F_D(p')) + w^p_R \sigma^p_R(F_R(p), F_R(p'))$$

The similarity function $\sigma_p$ for properties $p$ and $p'$ contains the similarity function for each feature, i.e. label ($\sigma^p_L$), domains ($\sigma^p_D$) and range ($\sigma^p_R$). Further, each similarity function has a function $F$ that extracts feature values from the property. In particular, $F_L(p)$ returns the label value of property $p$, $F_D(p)$ returns the domain concepts of property $p$, and $F_R(p)$ returns the range concept of property $p$. Last not least, there are weight variables, i.e. $w^p_L$, $w^p_D$, and $w^p_R$, attached to each of the similarity functions.

Since there are circularities and dependencies between similarity definitions these functions cannot be solved easily. In particular, $\sigma_L$ depends on $\sigma^p_L$, which is calculated by using $\sigma_p$, and $\sigma_R$ depends on $\sigma^p_D$ and $\sigma^p_R$, which is calculated through concept similarity measure $\sigma_c$ both. Thus, an iterative approach is needed to resolve such dependencies.

[Elzenat/Valtchev 2004] proposes the OWL Light Aligner (OLA) algorithm in order to resolve circularities for similarity equations. The algorithm starts with some initial values, which can be set manually or gained from other similarity measurement algorithms. According to the example shown in Figure 5.11 existing concept similarity measure are taken from Table 5.28 (i.e. the normalized weight sum based on Manhattan distance as a similarity with $w_{label} = 3/4$, $w_{comments} = 1/4$). Further, since properties are considered to be the same, i.e. have the same labels, all properties have 1.00 as initial value. Initial values of concepts and property similarity are shown in Table 5.32 and Table 5.33 respectively.

<table>
<thead>
<tr>
<th></th>
<th>(A) shipment request</th>
<th>(B) customer</th>
<th>(C) consignee</th>
<th>(D) pickup period</th>
<th>(E) item</th>
</tr>
</thead>
<tbody>
<tr>
<td>transport order (1)</td>
<td>0.30</td>
<td>0.23</td>
<td>0.15</td>
<td>0.12</td>
<td>0.12</td>
</tr>
<tr>
<td>source address (2)</td>
<td>0.18</td>
<td>0.15</td>
<td>0.15</td>
<td>0.15</td>
<td>0.12</td>
</tr>
<tr>
<td>destination address (3)</td>
<td>0.09</td>
<td>0.06</td>
<td>0.06</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>pickup time (4)</td>
<td>0.12</td>
<td>0.30</td>
<td>0.24</td>
<td>0.59</td>
<td>0.27</td>
</tr>
<tr>
<td>good (5)</td>
<td>0.12</td>
<td>0.25</td>
<td>0.42</td>
<td>0.18</td>
<td>0.38</td>
</tr>
</tbody>
</table>

*Table 5.32 OLA initial concept similarity values ($\sigma^c_I$)*
It should be noted that all properties in Table 5.33 are abbreviated by two numbers or two letters respectively, which is due to saving space. Each number and each letter corresponds to a particular concept. The number or letter refers to a concept in Table 5.32. For example, the number combination 1:2 refers to "transport order" - "source address" and denotes the relation between both concepts. Another example is B-D, which refers to "customer" - "pickup period" and denotes this particular property, i.e. the relation between both concepts.

Before starting the iteration of the OLA algorithm one has to set the weighting values. In this case all weight are set equally for each feature, i.e. \( w^c_k = w^p_k = 1/2 \) and \( w^c_k = w^p_k = 1/3 \). The iteration starts with applying the similarity equation for concepts \( \sigma_c(c, c') \) and the similarity equation for properties \( \sigma_p(p, p') \), taking the similarity values from the previous step into account. Previous values for the first iteration are the initial values. For example, concept similarity between "pickup time" (5) and "shipment request" (A) is calculated by the following equation.

\[
\sigma^c_5(5, A) = \frac{1}{2} \sigma^c_5(5, A) + \frac{1}{2} \max(\sigma^c_5(4:2, A:B), \sigma^c_5(4:2, A:C), \sigma^c_5(4:2, A:E)) = 0.56
\]

On the other hand the property similarity between "transport order" - "source address" (1:2) and "shipment request" - "customer" (A:B) is calculated by the following equation.

\[
\sigma^p_1(1:2, A:B) = \frac{1}{3} \sigma^p_1(1:2, A:B) + \frac{1}{3} \sigma^p_1(1, A) + \frac{1}{3} \sigma^p_2(2, B) = 0.48
\]

All concept similarity measures of the first iteration are shown in Table 5.34, and all property similarity measures of the first iteration are shown in Table 5.35 respectively.
The results shown above are from the first iteration. After several iterations each similarity value converges to a particular one. Usually this happens after 10 iteration steps, i.e. after the 10th iteration step values do not change significantly anymore. Thus, similarity values after the 10th iteration step can be taken as end results. Similarity results after the 10th iteration are shown in Table 5.36 and Table 5.37 respectively.
All in all, similarity equation is an adequate solution for similarity measures between concepts with several features bearing dependencies and possible circularities. Thus, similarity equation can be is feasible for measuring similarity between service entities that have multiple features as mentioned in Table 5.20.

### 5.4.4 Summary on matching strategies

Matching strategies address the composition of basic matching algorithms. The composition of matching algorithms lead to a new matching algorithm by definition. Basically there are two ways of composition, i.e. sequential and parallel composition. Within a sequential composition one matching algorithm is processed leading to an alignment, which works as an input parameter for a second matching algorithm. In parallel composition two (or more) matching algorithms are processed resulting in two (or more) alignments that need to be aggregated to one single alignment. Aggregation can be accomplished using different aggregation techniques, e.g. maximum similarity aggregation, weight product aggregation, Minkowski distance aggregation, Manhattan distance aggregation and more.

The composition of matching algorithms in sequential or parallel order leads to better matching results. However, that problem that had been mentioned in section 5.3.2.2 (i.e. the problem of mutual influences between entities resulting from relational circulations) is not solved by matching algorithm composition only. In order to solve this issue global matching techniques had been introduced in this chapter. One of these approaches, i.e. similarity equation, had been identified as a proper solution for service entities.

Based on the matching algorithms of section 5.3 and 5.4 the BSM-ESM matching solution for this thesis is introduced in the following.
5.5  BSM-ESM matching solution

Several matching techniques have been mentioned so far, reaching from basic techniques (i.e. name-based techniques and structure-based techniques) to matching strategies (i.e. composition, aggregation, global computation and extraction). Since none of these techniques work in isolation, one has to combine them in a proper way in order get proper matching results. However, the selection and combination of matching techniques depend on service model, the use case, the domain, and other circumstance. For instance, a matching solution for the SCMM based models as suggested in this thesis would be different from a matching solution for other models, such as WSMO [Roman et al. 2006b], OWL-S [Martin et al. 2004] or USDL [Barros et al. 2011d]. This is due to the fact that different models contain different features and thus different matching algorithms are feasible, i.e. different basic techniques and different strategies. Further, depending on the model and the use case different weights and thresholds are applied. And last not least, depending on the application domain of the service model there are different external terminological resources. All in all, it should be noted that there are many different ways of matching solution.

Thus, the matching solution proposed here is just one example. It is well-grounded and justified for SCMM based models. Further, the matching solution is focused on logistics application domain and a use case in this area of transportation. According to the matching levels defined in section 5.2 the overall matching solution is subdivided in a matching solution on service entity property level (cf. 5.5.1), a matching solution on service entity level (cf. 5.5.2), a matching solution on service concept level (cf. 5.5.3), and a matching solution on service level (cf. 5.5.4).

5.5.1  Matching solution on service entity property level

Matching on service entity property level is based on properties of service entities as introduced in chapter 5.2. Properties are defined for each service concept. Instantiation of service concepts, i.e. service entities, have service entity properties with values. Matching features of service concepts and their relation to service concept properties have been introduced in chapter 5.3.2.2 (cf. Table 5.20) already. In this chapter, matching techniques are assigned to each feature. Basically there are seven different feature types: (1) string, (2) set of strings, (3) text, (4) enumeration, (5) cardinality, (6) service concept, (7) set of service concepts.

(1) Strings

Service concept entities have a name from type string as defined in the SCMM. String-based matching techniques have been introduced an examined in detail in chapter 5.3.1.2. Assuming that strings have already been normalized (i.e. syntactic
normalization is not necessary) substring similarity as defined in section 5.3.1 is considered as a viable technique for measuring string similarity.

(2) Set of strings

Service concept entities\(^{40}\) have a set of labels. The SCMM defines labels as a set of strings. Since labels are denoted as synonyms for service concept entities extrinsic language-based techniques are appropriated for measuring similarity. However, it is assumed that all available synonyms are encompassed by a label and that there is no external resource involved. In this case a synonym substring similarity is appropriated as defined in Definition 5.24.

(3) Text

Service concept entities\(^ {41}\) have a description providing a textual statement about each entity. A description is a text containing multiple words. Measuring similarity of texts is addressed by cosine similarity. However, since descriptions may differ (e.g. "location-based handling” vs. "location based handling") and my contain stopwords (e.g. "to", "the", "of", "on" or "a"), syntactic normalization (cf. chapter 5.3.1.2) and linguistic normalization (cf. chapter 5.3.1.3) is applied previously. Cosine similarity as defined in Definition 5.21 is applied after normalization.

(4) Enumeration

An enumeration is a set of named constants. There are three obvious and one hidden enumeration defined in the SCMM. The three obvious ones are ResourceInvolvementType, StakeholderInvolvementType, and DataType. The hidden enumeration is defined by three resource types HumanResource, MaterialResource, and ImmaterialResource. These types are ServiceConcepts, but are denoted as enumerations here, since all of them are inherited by the same super concept (i.e. Resource) and show a discrete set of named constants. In order to measure similarity between enumeration values used within service concept entities, one has to distinguish between whether one or multiple values can be set within an entity.

A resource has one resource type only and a DataProperty has one data type (defined by the hasRange property) as well. In both cases case it is appropriated to predefine a similarity matrix as propose in the data type similarity section of chapter 5.3.2.1. In case data type predefined values from Table 5.15 are appropriated and will be applied. In case of resource type a matching value of 1 is predefined if types are the same and 0 of types are not the same.

A Resource and Stakeholder service concept entities may have multiple involvement types (i.e. ResourceInvolvementType and StakeholderInvolvementType

\(^{40}\) 'Most’ means that almost all service entities have a name. However, there are service entities that work as a container for other entities and thus do not have a name, such as Resource and InvolvedResource.

\(^{41}\) 'Most’ means that almost all service entities have a name. However, there are service entities that work as a container for other entities and thus do not have a name, such as Resource and InvolvedResource.
respectively). So, in this case string equality will be applied for each combination. Further, combinations are aggregated using Jaccard similarity. Jaccard similarity is feasible for measuring similarity of sets (cf. 5.3.1.3).

(5) Cardinality

Cardinalities are used as LowerBound and UpperBound definitions in ObjectProperties and DataProperties as defined in the SCMM. Measuring cardinality similarity has been addressed in chapter 5.3.2.1. So, the cardinality similarity algorithm will be applied in this case.

(6) Service concept

Service concept entities contain relations to other entities. Relations between service concepts are defined by a service concept property that refer to other service concepts. In this case there is one service entity that refers to a single other service entity. For instance, a Resource refers to an OntologyConcept (using the property hasOntologyConcept). Problems of measuring similarity in case of interrelated service concepts had been introduced in chapter 5.4.3. A solution is provided by similarity equation and an iterative approach using OWL Light Aligner (OLA) (cf. 5.4.3.2). These algorithms are applied here.

(7) Set of service concepts

In contrast to (6) service concept entities may refer to multiple entities. For instance, a ConcreteService may contain one to many Actions, i.e. the ConcreteService "OrderReceipt" has four Actions "CreateShipmentRequest", "ProcessShipmentRequest", "ActivateShipmentRequest" and "ConfirmShipmentRequest". So, similarity equation and OLA iteration approach (cf. 5.4.3.2) can be applied in the first stage leading to multiple matching results. Thus, multiple matching results have to be aggregated using Jaccard similarity (cf. 5.3.1.3), which results in one single similarity value. Jaccard similarity aggregation is applied using a threshold of 0.4. This means, service concept entities are considered to be the equal if similarity values are greater or equal 0.4 during Jaccard aggregation. With other words, entities are considered as intersection within Jaccard aggregation if their matching value is greater or equal 0.4.

Table 5.38 provides an overview for matching features and corresponding matching technique. The table can be read as follows: the matching technique (fourth column) is appropriated for matching a particular feature (second column) from a particular type (third column), which belongs to a particular service concept (first column).
<table>
<thead>
<tr>
<th>Service Concept</th>
<th>Matching feature</th>
<th>Type</th>
<th>Matching technique</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract-Service</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>String</td>
<td>Cosine similarity</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>SuperAbstract-Service</td>
<td>Set(AbstractService)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>SubAbstract-Service</td>
<td>Set(AbstractService)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>Concrete-Service</td>
<td>Set(ConcreteService)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.20</td>
</tr>
<tr>
<td>Concrete-Service</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>String</td>
<td>Cosine similarity</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>SuperAbstract-Service</td>
<td>AbstractService</td>
<td>Similarity equation</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>Action</td>
<td>Set(Action)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.20</td>
</tr>
<tr>
<td>Action</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>String</td>
<td>Cosine similarity</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>Involved-Resource</td>
<td>Set(Involved-Resource)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.40</td>
</tr>
<tr>
<td>Involved-Resource</td>
<td>Resource</td>
<td>Resource</td>
<td>Resource entity similarity</td>
<td>0.60</td>
</tr>
<tr>
<td></td>
<td>InvolvementType</td>
<td>Set(Resource-InvolvementType)</td>
<td>String equality with Jaccard similarity aggregation (threshold 1.00)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>StateBegin</td>
<td>ResourceState</td>
<td>Similarity equation</td>
<td>0.15</td>
</tr>
<tr>
<td></td>
<td>StateEnd</td>
<td>ResourceState</td>
<td>Similarity equation</td>
<td>0.15</td>
</tr>
<tr>
<td>Resource</td>
<td>Ontology-Concept</td>
<td>OntologyConcept</td>
<td>OntologyConcept entity similarity</td>
<td>0.80</td>
</tr>
<tr>
<td></td>
<td>ResourceType</td>
<td>{HumanResource, MaterialResource, ImmaterialResource }</td>
<td>Predefined similarity matrix</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>ResourceState</td>
<td>Set(ResourceState)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td>Resource-State</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>String</td>
<td>Cosine similarity</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td>Resource</td>
<td>Resource entity similarity</td>
<td>0.40</td>
</tr>
<tr>
<td>Ontology-Concept</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>String</td>
<td>Cosine similarity</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>SuperOntology-Concept</td>
<td>Set(Ontology-Concept)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>SubOntology-Concept</td>
<td>Set(Ontology-Concept)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>ObjectProperty</td>
<td>Set(ObjectProperty)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td></td>
<td>DataProperty</td>
<td>Set(DataProperty)</td>
<td>Similarity equation with Jaccard similarity aggregation (threshold 0.40)</td>
<td>0.10</td>
</tr>
<tr>
<td>Object-Property</td>
<td>Name</td>
<td>String</td>
<td>Substring similarity</td>
<td>0.05</td>
</tr>
<tr>
<td></td>
<td>Label</td>
<td>Set(String)</td>
<td>Substring similarity with maximum value aggregation</td>
<td>0.45</td>
</tr>
</tbody>
</table>
5.5.2 Matching solution on service entity level

Matching on service entity level means calculating similarity for each BS service entity with each ES service entity, where both entities must belong to the same service concept. Since, matching on service entity property level leads to single value for each matching feature one can aggregate these values by using a normalized weight sum aggregation function (cf. chapter 5.4.2.2). This function is defined as follows.

Let $\sigma_{SE}$ be the similarity value between two service entities $e_{BS}$ and $e_{ES}$, where both entities belong to the same service concept, let $\sigma_{SEP}$ be the similarity value on service entity property level of a particular matching feature $i$, let $n$ be the number of features of a particular service entity, and let $w_{SEP}$ be the weight of that feature with $\sum w_{SEP} = 1$, the normalized weight sum aggregation function for matching on service entity level is defined as:

$$
\sigma_{SE} = \sum_{i=1}^{n} (w_{SEP} * \sigma_{SEP})
$$

**Definition 5.40** Normalized weight sum aggregation function for matching on service entity level

Using this function needs several weights $w_i$, which have to be defined for each matching feature. As defined the sum of all weights that belong to one service concept has to be 1. Weights should be set in a way that matching feature that are more relevant have higher weights and matching feature that are less relevant have lower one. However, the definitions of weights depend on the use cases and analy-
sis scenarios. In order to provide an example, Table 5.38 shows example weights in the fifth column. Weights that are shown in Table 5.38 highlight the label feature, i.e. synonyms of service entities. All other weights are spread equally.

5.5.3 Matching solution on service concept level

Matching on service concept level means that matching values on entity level are aggregated according to the service concept they belong to. The aggregation function for each service concept is defined as follows.

Let $\sigma_{SC}$ be the similarity value of a particular service concept, let $\sigma_{SE}(e_{BS}, e_{ES})$ the matching value on service entity level of a particular $e_{BS}$-$e_{ES}$ combination, where $e_{BS}$ and $e_{ES}$ belong to the same service concept, let $i$ be the index and $n$ be the total number of $e_{BS}$, let $j$ be the index and $m$ be the total number of $e_{ES}$, and let $\max_t$ be a function that returns maximum values above a defined threshold $t$, the aggregation function for matching on service concept level is defined as:

$$
\sigma_{SC} = \frac{\sum_{i=1}^{n} \max_t(\sigma_{SE}(e_{BS}, e_{ES}), ..., \sigma_{SE}(e_{BS}, e_{ES}), ..., \sigma_{SE}(e_{BS}, e_{ES}))}{n}
$$

with $\max_t(x_1, ..., x_n) = \begin{cases} 
\max_t(x_1, ..., x_n), & \text{if } x \geq t \\
0, & \text{if } x < t
\end{cases}$

Definition 5.41 Aggregation function for matching on service concept level

The threshold within the maximum function is necessary, since service entity matching values may contain irrelevant results. By applying a threshold $t$ matching value below $t$ are not considered. A threshold of 0.4 is applied in the remainder of this thesis.

5.5.4 Matching solution on service level

Matching on service level means that matching values on service concept level are aggregated leading to an overall similarity value for a particular BSM-ESM combination. The aggregation function is defined as follows.

Let $\sigma_{S}$ be the similarity value between the service model of a business service $sm_{BS}$ and the service model of an electronic service $sm_{ES}$, let $\sigma_{SC_i}$ be the similarity value on service concept level of a particular service concept $i$, let $n$ be the number of service concepts defined in the SCMM, and let $w_{SC_i}$ be the weight of that service concept with $\Sigma(w_{SC_i}) = 1$, the aggregation function for overall service matching is defined as:

- 167 -
\[
    \sigma_{SE} = \sum_{i=1}^{n} (w_{SC_i} \cdot \sigma_{SC_i})
\]

*Definition 5.42 Aggregation function for overall matching result on service level*

Weights for each service concept can be set differently. It is assumed that all service concepts have the same weight within aggregation on service level in the remainder of this thesis.

### 5.6 Summary

Based on business service models and electronic services models that have been described in the previous chapter, this chapter is dedicated to the matching of service. Service matching algorithms are adopted from ontology matching [Euzenat/Shvaiko 2007]. Ontology matching is concerned with measuring semantic heterogeneity between two ontologies. Adopting ontology matching algorithms is possible, since ontologies and service models are treated equally within this thesis (cf. chapter 3).

This chapter starts with an examination about matching terminology, such as comparison, similarity, alignment, etc. Later, basic matching algorithms and matching strategies are examined in detail. Each matching algorithm is applied by a logistics example in consideration of SCMM based service models. Based on distinct matching levels, which had been defined in section 5.2, this chapter provides matching solution for each matching level at the end. The overall matching solution proposed here is one example, which is well-grounded and justified for SCMM based models. Further, the matching solution is focused on logistics application domain and a use case in this area of transportation. The next chapter is dedicated to the visualization of matching results.
6 Visualization

Since the similarity measure leads to many numbers on different levels, the results cannot be interpreted easily. Thus, in order to provide decision support, one needs visualization. This chapter is dedicated to the visualization of service matching results. First, there is a general introduction to visualization of data and information. Next, visualization methods are examined and evaluated using distinct requirements. Last, visualization are applied for service matching results. Further, analysis suggestions are given.

6.1 Visualization of data and information

“Visualization is the process of transforming information into a visual form, enabling users to observe the information. The resulting visual display enables the scientist or engineer to perceive visually features which are hidden in the data but nevertheless are needed for data exploration and analysis.” [Gershon 1992] As such, visualization plays an important role for the usage of computers in general. A visualization method “… is a systematic, rule-based, external, permanent, and graphic representation that depicts information in a way that is conducive to acquiring insights, developing an elaborate understanding, or communicating experiences.” [Lengler/Eppler 2007]

There are many visualization methods existing. [Lengler/Eppler 2007] provides a structured view of existing visualization methods. In analogy to the periodic table of elements (from Russian chemist Dmitri Mendelejew in 1869 (cf. [Zolotov 2007]), Lengler and Eppler [Lengler/Eppler 2007] suggest a periodic table of visualization methods. Visualization methods are classified in five dimensions:

- Complexity (from low to high): Complexity refers to the number of rules applied for the usage of visualization. Further, it refers to the number of interdependencies of elements.
- Application area: The application area dimension is sub-divided into data representation, information representation, concept representation, metaphor representation, strategy visualization, and compound knowledge visualizations:
  - Data representation refers to the “visual representation of quantitative data in schematic form” [Friendly/Denis 2005]
  - Information representation refers to “the use of interactive visual representations of data to amplify cognition. This means that the data is transformed into an image; it is mapped to the screen space. The image can be changed by users as they proceed working with it.” [Card et al. 1999]
  - Concept representation refers to the representation of knowledge. Usually this is done by a “2-D graphical display where concepts (usually represented within boxes or circles), connected by directed arcs encoding brief relationships (linking phrases) between pairs of concepts. These relationships usual-
ly consist of verbs, forming propositions or phrases for each pair of concepts.” [Cañas et al. 2005]

- Metaphor representations fulfill a dual function. Visual metaphors “[…] position information graphically to organize and structure it. [… And] they convey an insight about the represented information through the key characteristics of the metaphor that is employed.” [Eppler/Burkhard 2005]

- Strategy visualization refers to “the systematic use of complementary visual representations to improve the analysis, development, formulation, communication, and implementation of strategies in organizations.” [Lengler/Eppler 2007]

- Compound knowledge visualizations refer to a combination representations from the aforementioned areas. As such they can become “complex knowledge maps that contain diagrammatic and metaphoric elements, conceptual cartoons with quantitative charts, or wall sized pictures.” [Lengler/Eppler 2007]

- Point of view (detail, overview, and detail and overview): Detail means that individual items are highlighted, overview means that the visualization provides a big picture, and detail and overview means that details and overview are provided at same time.

- Type of thinking aid (convergent vs. divergent): Convergent means that the representations server the purpose for reducing complexity. Instead, divergent means that complexity is added in order to highlight details.

- Type of representation (process vs. structure): Process refers to a sequential representation including steps and a timeline. Structure refers to hierarchy or causal network representations.

In analogy to the periodic table of elements, the periodic table proposed by [Lengler/Eppler 2007] contains visualization methods instead of “elements”, i.e. each cell of the table is dedicated to a visualization method. A visualization method contains a name and an abbreviation (e.g. pie chart is abbreviated Pi). The complexity and the application area dimension are assigned to the vertical and horizontal scale of the periodic table, i.e. complexity is assigned to the “period” (vertical) scale and the application areas are assigned to the “group” (horizontal) scale. In that way, less complex visualization methods are on top of the period table and the more complex visualization methods are at the bottom of the table. Application areas form groups, which are horizontally arranged and distinguished by certain colors. The dimensions point of view, type of thinking aid, and type of representation are represented by distinct signs and font colors. The periodic table of visualization methods including a legend of signs and colors is depicted in Figure 6.1.
For visualization of matching results between BS and ES, visualization methods from the application area data visualization and information visualization are of interest only. Since there is no sequential flow or timeline, process visualizations are not of interest. Instead, structure visualization methods are in focus. So, structure visualization for data and information are examined next.

6.2 Data and information visualization methods

In order to conclude about the appropriateness of visualization methods for BS-ES matching result representation, methods for data and information visualization proposed by [Lengler/Eppler 2007] are examined here. As already mentioned structure visualization are of interest only. Before examining each method in detail, one has to preselect, which methods are of interest in general and which of the methods can be neglected.

Considering information visualization semantic network (Se) [Sowa 1991], hyperbolic trees (Hy) [Lamping et al. 1995], entity relationship diagrams (E) [Chen 1976] and Cone tree (Cn) [Robertson et al. 1991] are not of interest, since these visualization methods do not contain any quantifications and cannot be used for matching result visualization. Venn diagrams (Ve) [Lewis 1918] and clustering (Cl) [Tan et al. 2006a] are used in set theory, which is not of interest for visualization here. Although sankey diagrams (Sa) [Schmidt 2008] are not considered as...
structure visualization, they contain sequences and, thus, are not of interest here either. Data maps (Da) [Burrough 1986] are not considered any further, since there is no need for geographic data visualization. Considering data visualization spectrograms (Sp) [Haykin 1995] are used for showing the spectral density of signals, such as light or sounds and, thus, are not of interest here. Instead, the visualization methods continuum (C), table (Tb), cartesian coordinates (Ca), scatter plot (Sc), pie chart (Pi), line chart (L), area chart (Ac), bar chart (B), histogram (Hi), tukey box plot (Tk), radar chart (R), parallel coordinates (Pa) and tree map (Tp) are examined in next.

6.2.1 Continuum (C)

Continuum (C) visualizations present values using a one-dimensional scale. According to [Lengler/Eppler 2007] this is a very simple visualization method, but very useful for illustration of relations between values that belong to the same scale. An example of a continuum is the temperature scale [Wikipedia 2010].

6.2.2 Tables (Tb)

Tables are widely used in data analysis and research. A table contains an arrangement of rows and columns. The intersection of a row and a column is a cell. There are simple tables and multi-dimensional tables. A simple table contains a head row including column identifiers, which can be names, words, phrases, or numerical indices. Further, a simple table contains multiple rows underneath. Each cell contains a particular value. A multi-dimensional table contains identifiers for each dimension. For instance, a 2-dimensional table has a head row and a head column including identifiers for both. In 2-dimensional tables each row-column intersection refers to a single cell-value. Row, column and value form an injective relation. Further, rows and columns may be grouped, segmented, or arranged in many different ways, or even nested recursively [Leech et al. 2005]. Several examples of 2-dimensional tables had been shown in chapter 5 already. Tables can provide overview and details [Lengler/Eppler 2007], which make tables a simple and flexible tool for data representation.

6.2.3 Cartesian coordinates (Ca) and scatter plot (Sc)

A coordinate system is a system which uses one or more coordinates to uniquely determine the position of a point. A coordinate system contains coordinate axes. If coordinate axes are oriented perpendicularly, the coordinate system is called cartesian coordinate system. Usually there are two or three dimensional coordinate system with two (i.e. x, y) or three (i.e. x, y, z) axes respectively. Each axis spans from $-\infty$ to $+\infty$. Usually, axes have an intersection at point 0 [Weisstein]. Two
dimensional cartesian coordinates are used in order to illustrate functions, i.e. the
relation between one value called x and another one called y.

A scatter plot (Sc) [Ults 1999] is a mathematical diagram using a cartesian coordi-
nate system. A scatter plot displays a set of values from two variables. The value
set is displayed as a collection of points. Each point represents one value of the
first variable determining the position on the horizontal axis x and another value
of the second variable determining the position on the vertical axis y. Examples
for scatter plots are given in [Wikipedia 2011a].

6.2.4 Pie chart (Pi)

A pie chart is a circular chart, which illustrates proportions of a whole as sectors
of a circle. Given 100% and several proportions (e.g. 40%, 25%, 20%, and 15%),
a pie chart visualizes each proportions as sector, i.e. as an angle of the circle (e.g.
144°, 90°, 72°, and 54°). The summation of all angles results in 360 degrees. Pie
charts are named for their resemblance to a pie that had been cut into slices. Pie
charts are widely used in business and mass media [Krug 2011]. However, they
are criticized in scientific world, since it is difficult to compare different sectors
across different pie charts. Other plots, such as the line charts, bar charts or tables,
are usually more appropriate (cf. [Cleveland 1984,Spence 2005,Wilkinson 2005]).

6.2.5 Line chart (L) and area chart (Ac)

A line chart is created by connecting a series of points in a coordinate system with
a line. A line chart suggests that represented data is connected. Usually, line charts
are used in order to visualize a trend of data within an interval of time. An area
chart is based on a line chart, but emphasis the area between lines and the x-axis
with colors, hatchings or textures. Each area in the chart represents a quantity.
Usually, area charts are used in order to compare these quantities.

6.2.6 Bar charts (B) and histogram (Hi)

A bar chart is a chart with rectangular bars. Each bar has a length that is propor-
tional to the values it presents. Bars can be plotted horizontally or vertically. Bar
charts are used to represent discrete items that cannot be grouped and that are not
continues. Examples for such items are ‘colors’ or ‘countries’. In contrast, contin-
ues data, such as ‘height’ or ‘weight’, leads to a histogram. Histograms had been
introduced by Karl Pearson [Pearson 1894]. Histograms and bar charts are often
mistaken for each other, since they look similar at first glance. However, Histograms
show continues data, which leads to a slightly different visualization form.
In bar charts all bars have the same width and are separated by a space. In con-
trast, bars in histograms touch each other and may have different width (when
representing different intervals of continues data). The difference between bar charts and histograms is examined in [Willis].

6.2.7 Tukey box plot (Tk)

Tukey box plots (or box plots) are used for depicting groups of numerical data using vertical boxes. Boxes are arranged similar to bars in a bar chart. They can be plotted horizontally or vertically. Each box is divided in five parts showing five numbers: the smallest observation (minimum), a lower quartile, the median, an upper quartile and the largest observation (maximum). Further a box plot might also consider outliers. Spacing between different parts help to indicate the degree of dispersion (spread) and skewness in data, and identify outliers. Details about box plots are available in [McGill et al. 1978] and [Massart et al. 2005].

6.2.8 Radar chart (R)

A radar chart (R) also known as star plot [Chambers 1983] is a visualization method of displaying multivariate data. Radar charts show three or more quantitative variables. Each variable belongs to one axis. All axes starting from the same point. The data length of an axis is proportional to the magnitude of the variable for the data point relative to the maximum magnitude of the variable across all data points. A line is drawn connecting the data values on each axis. Further details about radar charts are explained in [Chambers 1983] and [Croarkin/Tobias 2010].

6.2.9 Parallel coordinate (Pa)

Similar to a radar chart (R), parallel coordinate (Pa) can display multivariate data. It shows multiple quantities variables in a coordinate system. Each variable relates to one coordinate. All coordinates are parallel arranged. Thus, all coordinates span a n-dimensional space. A point in this space represents a value of a variable. Points are connected through a line [Inselberg 1985].

6.2.10 Tree map (Tp)

Tree maps display hierarchical data by using nested rectangles. Each branch of the tree is given a rectangle, in which sub-branches are nested as rectangles in the parent-rectangle. Further, the rectangle of a leaf node has an area proportional to a specified dimension on the data. Usually, leaf nodes are colored in order to show a separate dimension of the data. Tree maps (Tp) make efficient use of space. They can legibly display thousands of items on the screen simultaneously. In that way,
tree maps can visualize patterns that would be difficult to spot in other ways [Shneiderman 2006, Shneiderman/Plaisant 2009].

6.3 Visualization method evaluation

As shown above, there are several visualization methods. However, these are just some examples. There are many other visualization methods or modifications of existing ones. For service matching visualization, one has to evaluate visualization methods. In this section, requirement for visualization are defined, first. Second, existing visualization methods are evaluated considering these requirement.

6.3.1 Visualization method requirements

Basically, the definition of visualization method requirements is about the definition of visualization dimensions. However, first, some general principles for visualization are mentioned [Reuter et al. 1990]:

- A careful choice and expression of scales is essential
- Straight lines are easier to perceive than curves
- Horizontal lines are easier to perceive than oblique lines
- Things that are closer together are easier to compare than things far apart
- Things of equal importance should have a roughly equal visual impact
- Irrelevant material can seriously interfere
- Motion is more effective for conveying 3-dimensional depth than stereopsis or perspective
- Principles of good graphical display are often in conflict with each other, necessitating trade-offs among them

In addition to these principles, the following guidelines apply for this thesis:

- Visualization depends on the raw data and the goal of the analysis.
- Since creation of 3-dimensional visualization is time-consuming and complex (e.g. applying motions), 2-dimensional visualizations are considered here only.
- A set of visualization methods will be used, since there is no single visualization method that fits all concerns. Further, even though, there are many possibilities for visualization, only a distinct set of visualization method is used. This provides clearness, avoids redundancy and eliminates irrelevant material.

These principles and guidelines lead to the requirements for visualization. Visualization dimensions can be deduced from raw data and goals. So, both are considered in more detail.

Regarding raw data there is detailed data, e.g. matching values on service entity property level, and aggregated data, e.g. matching values on service level, as well. Thus, level-of-detail is one dimension for visualization. The detail level is accord-
ing to the matching levels mentioned already, i.e. service matching level, service concept matching level, service entity matching level and service entity property matching level.

The goal of the analysis is to provide decision support for finding suitable BS-ES combinations. Since there are many BS-ES combinations, one has to compare several BS-ES matching results at same time. On the other hand, it is necessary to analyze matching result for one single BS-ES combination in detail. This leads to the second dimension for visualization, i.e. the number of BS-ES matching results. This dimension is categorized into: analysis of a single BS-ES comparison (Single), analysis of a two BS-ES comparisons (Two) and the analysis of n BS-ES comparisons (N).

6.3.2 Visualization method selection

Based on the principles mentioned above and the dimensions, proper visualization methods are selected in this section. This is done by considering the dimension number-of-combinations in the first place and the dimension level-of-detail next to them. Suitable visualization methods are selected based on these dimensions.

6.3.3 Single BS-ES combination

Considering a single BS-ES combination and the corresponding matching results a suitable visualization method for all level-of-details is a table. However, since there is raw data on all levels-of-detail, a nested table is need. The application of a nested table as well as suggestions for analysis are given in chapter 6.4.1.

Another suitable graphical representation for one BS-ES combination is a bar chart. Bar charts provide visualization on three different levels-of-detail. On the one hand, it may show the arrangement of results on service concept level, including an overall aggregated service matching value. Second, a bar chart may provide details about the spreading of certain matching results on service entity level. The application of a bar chart as well as suggestions for analysis are given in chapter 6.4.1.

Other visualization methods are not or less suitable for single BS-ES matching result consideration. Continuum (C) is not applicable, since there is no data that can be arrange in order. Cartesian coordinate (Ca) and scatter plot (Sc) represent two axes and, thus, need two comparison values, which is not suitable considering a single BS-ES result set. Pie charts (Pi) are used to present spreads of several parts from a whole, which is not suitable here. Line charts (L) or area charts (Ac) are similar to bar charts. However, L and Ac suggest that values are connected or form an area, which makes no sense in this case. Tukey box plots (Tk) are similar to bar charts as well, but provide a more complex representation, which is not needed here. Radar charts (R) are not feasible for single comparison, since they
are usually used for multiple comparison purposes. Parallel coordinates (Pa) are used for illustration of points in an n-dimensional space, which is inappropriate here. Tree maps (Tp) are used for displaying hierarchical data with nested rectangles, where the size of the rectangle displays proportion of values. Tp is not feasible, since there are no proportion data in a single BS-ES matching result set.

6.3.3.1 Two BS-ES combinations

The comparison of two BS-ES matching results is needed in order to face two BS-ES combination directly. This can be done on service level, service concept level and service entity level. The visualization of two BS-ES matching results on service entity property would lead to a too complex visualization. However, given that one BS of two BS-ES combinations is the same, one can illustrate matching results using a scatter plot (Sc). In that way, matching values from one BS-ES combination can be represented as x-values and matching values form the second BS-ES combination can be represented as y-values. The application of scatter plots (Sc) for matching results of two BS-ES combinations on different levels, a detailed explanation, and suggestions for analysis are given in 6.4.2.

Other visualization methods are not or less suitable for the comparison of two BS-ES matching result sets. Continuum (C) is not useful, since considering two different value combinations would results in a continuum with two values only. A continuum (C) would result in an opaque amount of data. A table (T) including pure numbers would be feasible. However, a nested table provides a less convenient overview in this case. Pie charts (Pi) are not suitable for the comparison of two values either. Line charts (L), area charts (Ac), bar charts (B), histograms (Hi) and tukey box plots (Tk) are less useful, since comparison between two values is less obvious using these visualization methods. Radar charts (R) can be used, but they are more suitable comparing multiples values. Thus, they are considered in the next section. Parallel coordinates (Pa) are used for illustration of points in an n-dimensional space, which is inappropriate here. Tree maps (Tp) are used for displaying hierarchical data, which is inappropriate here either.

6.3.3.2 Multiple BS-ES combinations

Considering the comparison of several BS-ES combination one should distinguish matching results on service level, service concept level and service entity level. The visualization of multiple BS-ES matching results on service entity property level is not considered, since this visualization would be too complex.

Comparing multiple BS-ES service matching values using a scale leads to a continuum (C) visualization. However, one can illustrated matching results on service level as an ordered list or an ordered table as well. The application of the continuum (C) and table (Tb) for service matching values of multiple BS-ES combina-
tions, a detailed explanation, and suggestions for analysis are given in chapter 6.4.3.

However, continuum (C) and tables (Tb) are not suitable, for visualization of service concept matching results or service entity matching results of multiple BS-ES combinations. In this case, radar charts (R) are appropriated means of visualization. On service concept level each service concepts relates to one axes. In this way aggregated matching values form multiple BS-ES combinations can be illustrated in one chart. Further, on service entity level, one can show all matching results from one service concept in a single chart as well. The application of radar charts (R) for multiple BS-ES combinations, a detailed explanation, and suggestions for analysis are given in chapter 6.4.3.

Other visualization methods are not or less suitable for visualizing multiple BS-ES matching results. Cartesian coordinate (Ca) and scatter plot (Sc) represent two axes only and, thus, are not suitable. Pie charts (Pi) are used to present the spread of several parts from a whole, which is inappropriate here. Line charts (L), area charts (Ac), bar charts (B), histograms (Hi) and tukey box plots (Tk) are less useful, since comparison between multiple values results in a bunch of opaque information. Parallel coordinates (Pa) and tree maps (Tp) are inappropriate either.

### 6.3.3.3 Summary

In order to select appropriated visualization methods one has to consider raw data and analysis goals. The consideration of both lead to two dimensions: number-of-combinations and the level-of-details. Both dimensions are illustrated below. Table 6.1 shows the number-of-combinations categorization as rows and level-of-details categorization as columns. Cells contain visualization methods that have been identified and selected. Application, examples and analysis suggestion are provided in section 6.4.

<table>
<thead>
<tr>
<th></th>
<th>Service level</th>
<th>Service concept level</th>
<th>Service entity level</th>
<th>Service entity property level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single</td>
<td>Table, Bar Chart</td>
<td>Table, Bar Chart</td>
<td>Table, Bar Chart</td>
<td>Table</td>
</tr>
<tr>
<td>Two</td>
<td>Scatter Plot</td>
<td>Scatter Plot</td>
<td>Scatter Plot</td>
<td>-</td>
</tr>
<tr>
<td>Multiple</td>
<td>Continuum, Table</td>
<td>Radar Chart</td>
<td>Radar Chart</td>
<td>-</td>
</tr>
</tbody>
</table>

*Table 6.1 Selected visualization methods*
6.4 Visualization method application and analysis

According to the dimensions number-of-combinations and level-of-details, appropriate visualization methods had been selected so far. These methods are applied in this subchapter in order testify their adequacy. For this purpose sample matching results are applied to these visualization methods and suggestions for interpretation and analysis are given. This chapter is divided into three subchapters. Each subchapter addresses visualization methods of each number-of-combinations category.

6.4.1 Tables and bar charts for single BS-ES comparison

As mentioned previously (cf. 6.3.3), tables are appropriated for raw data representation. Since raw data results are on different level-of-details a nested table is necessary. A table suggested for this purpose including sample data is shown below (cf. Table 6.2).

The table starts with a row on the top including names of one particular BS and one particular ES. Both names are separated by a colon sign. Next to the names, there is the overall matching value. The overall matching value represents the matching value on service level and results from a weighted average calculation of all service concept results. Under the top row there are several tables. Each table relates to one service concept showing detailed matching results. Table 6.2 shows the two service concepts AbstractService and ConcreteService, for instance. Each table has a head row including the labels “Business Service” on the right hand side, “Matching Value” in the middle and “Electronic Service” on the right hand side. The “Matching Value” is arranged in the middle in order to illustrate that the matching value considers two particular concepts from the BS and the ES. Each head labels contains a number in brackets. The number for the BS and the ES label refers to the number of concepts. For instance (cf. Table 6.2), there are two AbstractServices on BS and ES side each and, thus, the head row contains “Business Service (2)” and “Electronic Service (2)”. The BS and ES concepts are arranged under their corresponding head cells. Since there is a matching value for each BS-ES concept combination there are two rows on BS side and two times two rows on the ES side, i.e. the ES concepts appear as many times as BS concepts exist for each service concept. This may lead to many rows. However, all ES concepts are arranged as an ordered list, in which results that are more relevant appear on top and less relevant ones below. The matching value of each concept combination is shown in the middle as illustrated in Table 6.2. Each middle cell has one matching value and zero, one or two numbers in brackets. The matching value represents the similarity value of a particular BS-ES-concept combination. The first row for a particular BS concept contains a average (Ø) number, which refers to the arithmetic mean of all matching values for a particular BS concept. Further each BS-ES combination (with exception of the last one) has a delta number (Δ). The delta represents the difference between a particular matching value of one BS-ES con-
cept combination to the matching value of the next most relevant BS-ES concept combination. For example (cf. Table 6.2), the ConcreteService concept “ConcreteServiceB1” is matched with three ConcreteService concept from ES side resulting in the matching values 0.90, 0.40, and 0.10. The average number (Ω) is 0.47 (=1/3*(0.90 + 0.40 + 0.10)) as presented in the first cell. Further, the delta (Δ) between the first and the second number is 0.50 (0.90 - 0.40) and the delta (Δ) between the second and the third number is 0.30 (0.40 - 0.10) illustrated in the particular cell. Last not least, the head cell of the middle column, i.e. “Matching Value”, contains another average (Ω) number in brackets. This average number represents the overall matching value for a particular service concept, which results from calculating the average the maximum matching values from each BS concept with all ES concepts. For example (cf. Table 6.2), the overall matching value of ConcreteService concept is 0.72 (=1/3*(0.90 + 0.75 + 0.50)).

<table>
<thead>
<tr>
<th>BSM1 : ESM1 - Overall: Ø0.63</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractService</td>
</tr>
<tr>
<td>Business Service (2)</td>
</tr>
<tr>
<td>AbstractServiceB1</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>AbstractServiceB2</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

| ConcreteService             |
| Business Service (3)        | Matching value (Ω 0.72) | Electronic Service (3) |
| ConcreteServiceB1           | 0.90 (Ω 0.47, Δ 0.50)   | ConcreteServiceE1      |
|                            | 0.40 (Δ 0.30)           | ConcreteServiceE2      |
|                            | 0.10                    | ConcreteServiceE3      |
| ConcreteServiceB2           | 0.75 (Ω 0.55, Δ 0.15)   | ConcreteServiceE1      |
|                            | 0.65 (Δ 0.40)           | ConcreteServiceE3      |
|                            | 0.25                    | ConcreteServiceE2      |
| ConcreteServiceB3           | 0.50 (Ω 0.30, Δ 0.30)   | ConcreteServiceE2      |
|                            | 0.20 (Δ 0.00)           | ConcreteServiceE1      |
|                            | 0.20                    | ConcreteServiceE3      |

Action

...
The numbers and matching values shown in Table 6.2 are useful for detailed exploration. However, in order to get a visual overview, a graphical representation is desirable. As mentioned in section 6.3.3 bar charts are appropriated visualization methods for that purpose. Starting with the aggregated matching values on service concepts, Figure 6.2 represents a corresponding bar chart.

![Figure 6.2 Bar chart for a single BS-ES matching results set on service level and service concept level](image)

The bar chart (cf. Figure 6.2) shows the average matching value for each service concept. Each bar has a label including the name of the service concept (e.g. AbstractService) at the bottom of the chart, and each bar has a numeric value in top of the bar, which represents the matching value. Further, the chart contains the overall average value (in this case 0.52) as a horizontal line. In that way one can conclude, which service concept is above and which one is below the overall matching value. In case of Figure 6.2, the Action, Resource and ObjectProperty have lower matching results, i.e. BS and ES have less terminological correspondences in these areas. One the other hand, BS and ES correspond by their AbstractService representation.

Further, since matching values, average values and delta values on service entity level are hard to compare using a table (cf. Table 6.2), a graphical representation of these values is necessary as well. Figure 6.3 shows the matching values considering ConcreteService entities from Table 6.2 as a bar chart.

---

42 Values in the bar chart are example data and do not correspond the values in Table 6.2.
As shown in Figure 6.3 there are three ConcreteService from BS side, i.e. ConcreteServiceB1, ConcreteServiceB2, and ConcreteServiceB3. The matching value of each combination is represented by one bar each, whereupon the highest matching value is blue colored, the second highest is red colored and the lowest has a green color. Further, the average for each particular ConcreteService from business side is represented as a point including the average number. Using this visualization one can show the spread of matching values and, thus, can conclude about their qualities. For example, the first BS:ConcreteService has a very high matching value (i.e. 0.9) with one of the ES:ConcreteServices. Since the second matching value has 0.4 and the third 0.1 only, one can infer that the first value is very strong. With other words the matching between ConcreteServiceBS1 and ConcreteServiceE1 is high and has a high quality, i.e. it is likely that these services are similar. On the other hand, the highest matching value ConcreteServiceB2 is 0.75, which is still high. However, the second matching value is 0.65, which is 0.1 less than the highest one. Thus, one can infer that it is uncertain which of both, ConcreteServiceE1 or ConcreteServiceE3, is more similar with ConcreteServiceB2. Because of uncertainty the quality of the matching results is weak for ConcreteServiceBS2. An indicator for the quality of the matching result is the difference between the maximum and the average value. A higher difference indicates high quality. A low difference indicates that there are several similar matching results, which cause uncertainty about the conclusion whether a proper matching has been found.
6.4.2 Scatter plots for two BS-ES comparisons

Comparing two BS-ES combinations with each other directly one can use scatter plots as suggested in chapter 6.3.3.1. This works under the condition that the BS of both combinations is the same only. In that way all matching results of one BS-ES combination can be represented as x-values and all matching results of the second one can be represented as y-values. Assuming the following sample data (cf. Table 6.3), the corresponding scatter plot is shown in Figure 6.4.

<table>
<thead>
<tr>
<th>Service Level</th>
<th>BS1:ES1</th>
<th>BS1:ES2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall (Ø)</td>
<td>0.63</td>
<td>0.45</td>
</tr>
<tr>
<td>AbstractService (Ø)</td>
<td>0.70</td>
<td>0.55</td>
</tr>
<tr>
<td>BS1::AbstractServiceB1</td>
<td>0.75</td>
<td>0.80</td>
</tr>
<tr>
<td>BS1::AbstractServiceB2</td>
<td>0.65</td>
<td>0.30</td>
</tr>
<tr>
<td>ConcreteService (Ø)</td>
<td>0.72</td>
<td>0.37</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB1</td>
<td>0.90</td>
<td>0.50</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB2</td>
<td>0.75</td>
<td>0.20</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB3</td>
<td>0.50</td>
<td>0.40</td>
</tr>
</tbody>
</table>

Table 6.3 Sample result set of two BS-ES combinations

![Scatter plot comparison of two BS-ES matching result set on service level, service concept level, and service entity level](image)

*Figure 6.4 Scatter plot comparison of two BS-ES matching result set on service level, service concept level, and service entity level*

The scatter plot shown in Figure 6.4 contains the matching results of BS1:ES1 as y-values and the matching results of BS1:ES2 as x-values. All samples values, i.e.
the overall matching value, the aggregated value for AbstractService and ConcreteService as well as the corresponding detailed matching values, are shown in the coordinate system. Each type has a certain color and a certain sign for better distinction. In order to decide, which combination has higher values there is a diagonal line, which divides the coordinate system into two areas. Values above the diagonal are more related to the BS-ES combination on the y-axis (in this case BS1:ES1) and values below the diagonal are more related to the BS-ES combination on the x-axis (in this case BS1:ES2). Further it is meaningful to define thresholds in order to define relevance of matching results. In case of Figure 6.4 the threshold is set to 0.5. Diagonal and threshold form six areas (i.e. Q1a, Q1b, Q2, Q3a, Q3b, Q4) with a distinct meaning:

- Q1a means that the matching value is more related to BS-ES combination on the y-axis (e.g. BS1:ES1), but less relevant, because of not reaching the required threshold
- Q1b means that the matching value is more related to BS-ES combination on the x-axis (e.g. BS1:ES2), but less relevant, because of not reaching the required threshold
- Q2 means that the matching value is related to BS-ES combination on the y-axis (e.g. BS1:ES1), since it exceeds the y-threshold. Further, it is not relevant for the BS-ES combination on the x-axis (e.g. BS1:ES2), since it is not reaching the required x-threshold.
- Q3a means that the matching value is more related to BS-ES combination on the y-axis (e.g. BS1:ES1). However, it is relevant in both combinations, since both matching values exceed the required threshold
- Q3b means that the matching value is more related to BS-ES combination on the x-axis (e.g. BS1:ES2). However, it is relevant in both combinations, since both matching values exceed the required threshold
- Q4 means that the matching value is related to BS-ES combination on the x-axis (e.g. BS1:ES2), since it exceeds the x-threshold. Further, it is not relevant for the BS-ES combination on the y-axis, since it is not reaching the required y-threshold.

Considering the example in Figure 6.4, the overall matching values as well as all ConcreteService values are located in Q2, and, thus, are related to the combination BS1:ES1. One AbstractService is located in Q2 and another one in Q3b, which results in an average AbstractService result in Q3a. Thus, AbstractService matching results are more related to BS1:ES1, even though they are above the x-threshold of BS1:ES2. All in all the matching results are higher for the BS1:ES1 combination as illustrated in Figure 6.4. That means, comparing BS1:ES1 and BS1:ES2, the BS-ES combination BS1:ES1 shows a higher terminological correspondence.
6.4.3 Continuum and radar charts for multiple BS-ES comparisons

Considering the comparison of multiple BS-ES combination the visualization methods continuum, table and radar view have been selected. Continuum is feasible in order to visualize overall service matching values using a sequential one-dimensional axis. For example, ten BS-ES combinations and their corresponding overall matching results shown in Table 6.4 result into the continuum representation shown in Figure 6.5.

| BS1:ES1  | 0,63 |
| BS1:ES2  | 0,45 |
| BS1:ES3  | 0,31 |
| BS1:ES4  | 0,75 |
| BS1:ES5  | 0,09 |
| BS1:ES6  | 0,01 |
| BS1:ES7  | 0,27 |
| BS1:ES8  | 0,15 |
| BS1:ES9  | 0,05 |
| BS1:ES10 | 0,21 |

*Table 6.4 Sample overall matching result ten BS-ES combinations*

*Figure 6.5 Continuum comparison of multiple BS-ES matching result sets on service level*

Figure 6.5 shows that the BS-ES combination BS1:ES4 has the highest matching value and BS1:ES2 has the second highest one. Further, seven of ten overall matching values are under 0.35. Using this continuum visualization, one can concluded that the ES candidates with the highest probability of being relevant for BS1 are ES1, ES2 and ES4. Thus, further consideration should focus a detailed examination of matching results of these three ES candidates.

A examination of matching values on service concept level and service entity level considering multiple BS-ES combinations can be done by using radar charts. Based on sample matching results of multiple BS-ES combination shown in Table 6.5, the following radar chart applies (cf. Figure 6.6).
### Table 6.5 Sample service concept matching result of five BS-ES combinations

<table>
<thead>
<tr>
<th>Service Concept</th>
<th>BS1:ES1 (Ø0.63)</th>
<th>BS1:ES2 (Ø0.45)</th>
<th>BS1:ES3 (Ø0.31)</th>
<th>BS1:ES4 (Ø0.20)</th>
<th>BS1:ES5 (Ø0.09)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbstractService</td>
<td>0.71</td>
<td>0.43</td>
<td>0.33</td>
<td>0.17</td>
<td>0.10</td>
</tr>
<tr>
<td>ConcreteService</td>
<td>0.53</td>
<td>0.66</td>
<td>0.23</td>
<td>0.14</td>
<td>0.12</td>
</tr>
<tr>
<td>Action</td>
<td>0.38</td>
<td>0.45</td>
<td>0.24</td>
<td>0.23</td>
<td>0.11</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>0.55</td>
<td>0.32</td>
<td>0.32</td>
<td>0.19</td>
<td>0.20</td>
</tr>
<tr>
<td>Resource</td>
<td>0.44</td>
<td>0.41</td>
<td>0.21</td>
<td>0.31</td>
<td>0.04</td>
</tr>
<tr>
<td>OntologyConcept</td>
<td>0.54</td>
<td>0.39</td>
<td>0.28</td>
<td>0.45</td>
<td>0.03</td>
</tr>
<tr>
<td>ObjectProperty</td>
<td>0.42</td>
<td>0.35</td>
<td>0.22</td>
<td>0.38</td>
<td>0.05</td>
</tr>
<tr>
<td>DataProperty</td>
<td>0.56</td>
<td>0.41</td>
<td>0.22</td>
<td>0.34</td>
<td>0.02</td>
</tr>
</tbody>
</table>

### Figure 6.6 Radar chart comparison of multiple BS-ES combinations on service concept level

The radar chart in Figure 6.6 has eight axes, i.e. one axis for each service concept. Each axis starts from 0 and ranges to a maximum of 1. Each service concept matching value of each BS-ES combination refers to a distinct point of one axis. All values, i.e. all axes points, are connected by a colored line. Each color refers to a certain BS-ES combination. In that way, each BS-ES result set forms a polygon. Each polygon has a distinct area. Thus, in order to find the highest matching result one has to find the polygon with the biggest area size. In case of Figure 6.6 this is the polygon of the BS1:ES1 matching result set. Beside the size of the polygon one conclude about matching results for particular service concepts. For example,
the BS1:ES1 combination has the highest matching value at an service level, but in the area of ConcreteService and Action the combination BS1:ES2 has higher matching results. One may infer the same information from a table as well. However, using a radar chart this information is provided at a glance.

Similar to visualizing aggregated service concept values one can illustrate matching values on service entity level. The following table (cf. Table 6.6) shows an example of the maximum values of a ConcreteService matching result set. There is one BS called BS1 and there are five ES. ConcreteServices of BS1 are shown in the head column. ConcreteService entities of ES1 to ES5 are shown in the head row. The values represent the maximum matching value of each BS-ES ConcreteService combination on service entity level.

<table>
<thead>
<tr>
<th></th>
<th>ES1</th>
<th>ES2</th>
<th>ES3</th>
<th>ES4</th>
<th>ES5</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS1::ConcreteServiceB1</td>
<td>0.80</td>
<td>0.50</td>
<td>0.32</td>
<td>0.19</td>
<td>0.10</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB2</td>
<td>0.75</td>
<td>0.20</td>
<td>0.21</td>
<td>0.31</td>
<td>0.12</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB3</td>
<td>0.60</td>
<td>0.40</td>
<td>0.28</td>
<td>0.51</td>
<td>0.13</td>
</tr>
<tr>
<td>BS1::ConcreteServiceB4</td>
<td>0.72</td>
<td>0.32</td>
<td>0.27</td>
<td>0.34</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 6.6 Sample maximum results from ConcreteService matching between one BS and five ES

The visualization of the sample data shown in Table 6.6 results in a radar chart shown in Figure 6.7. The interpretation of Figure 6.7 is similar to the interpretation given for Figure 6.6.
6.5 Summary

After the examination of service matching algorithms in chapter 5, this chapter was dedicated to the visualization of matching results. First, there was a general introduction about the visualization of data and information. Second, based on the periodic table of visualization of [Lengler/Eppler] existing visualization methods have been examined and pre-selected. Third, general principles and requirements for matching result visualization had been introduced. Requirements are based on the definition of visualization dimensions. Two dimension had been identified by an analysis of raw data and visualization goals. These dimension are: level-of-detail and number of BS-ES combinations that should be visualized in one plot. Based on the dimension characteristics visualization methods had been evaluated and a set of methods had been selected. Fourth, each selected visualization method had been applied using sample data. Further, analysis suggestions had been given for each example.

Next, the tool that supports service modeling (cf. chapter 3 and 4), service matching (cf. chapter 5) and the visualization of matching results according to this chapter will be introduced in chapter 7.
7 Tool support

In order to apply each part of this thesis in combination a tool is needed. A tool should support modeling, must apply matching algorithms and have to provide visualization of matching results. A tool for modeling business and electronic service according to the SCMM has already been introduced in chapter 4.5. So service preselection tool, which is introduced here, integrates EMF and, thus, can be used for modeling as well. However, main focus of the service preselection tool are the application of matching algorithms and the visualization of matching results. The overall tool architecture including basic constituents is presented first (cf. subchapter 7.1). Afterwards, each part of tool will be explained in detail.

7.1 Service preselection tool architecture

The service preselection tool architecture follows a typical three tier architecture [Eckerson 1995]. Three tiers are presentation tier, logic tier and data tier.

- The presentation tier represents the user interface and is the top-most level of the tool. Main function of the presentation tier is to translate tasks and results to something the user can understand. The service preselection tool provides a service modeling editor and visualization of results on that level.

- The logic tier processes commands, coordinates the application, makes logical decisions, and performs calculations. Main part of the service preselection tool on this tier is matching logic, i.e. applying matching algorithms that calculate matching results. Other functions such as visualization logic that provides data for the presentation layer and data logic that connects to the data tier are provided on this tier as well.

- The data tier provides capabilities of retrieving and storing information from a database or a file system. This information is processed in the logic tier and eventually provided on presentation level. The data tier of the service preselection tool contains of an object model, which is according to the SCMM. Data is serialized and stored as XML files on file system.

Basically, the service preselection tool is based on Eclipse platform [D'Anjou et al. 2005]. Eclipse provides an integrated development environment (IDE) and is an extensible platform for building IDEs at same time. The platform provides core mechanisms for controlling a set of tools working together to support programming tasks. However, tools can be contributed by third parties into the platform as well. Contributors can add functionality by wrapping their tools into pluggable components, i.e. into Eclipse plug-ins (cf. [Clayberg/Rubel 2004]). As such plug-ins add new functionality to Eclipse. The service preselection tool is an Eclipse plug-in. The advantage of being an Eclipse plug-in is that core Eclipse functionality, such as user interface components, project management, file loading, storing and printing, as well as functionality of other Eclipse plug-ins, such as EMF, can
be used out of the box. The tool architecture is represented in Figure 7.1 and explained below.

**Figure 7.1 Tool architecture**

Figure 7.1 shows the three layers presentation tier, logic tier and data tier. Within each tier there are several packages (denoted as rectangles) that provide distinct functionality. However, there are two rectangles that indicate tier spanning functionality. As an Eclipse plug-in the service preselection tool uses basic Eclipse functionality, such as file handling and basic user interface components. Further, since Eclipse Modeling Framework (EMF) has been identified as a proper tool for service modeling (cf. chapter 4.5), the service preselection tool uses EMF functionality, such as an SCMM based object model and an editor component that provides editing SCMM based service models. The object model provides access to business and electronic service information de-/serialized as XML files. Business and electronic service information is needed within the matching logic component. Matching logic implements the service matching algorithms from chapter 5 and provides matching results. Matching results are processed by the presentation logic component, which provides information to the presentation tier. The presentation tier visualizes matching results using the visualization techniques mentioned in chapter 6. For this purpose, table view visualization (cf. 6.4.1) is implemented using a HTML [Raggett et al. 1999] representation. All other visualizations, such as bar chart, scatter plot and radar chart, are implemented using an external library called JFreeChart [Gilbert/Morgner 2011]. The interaction between components is indicated by arrows in Figure 7.1. For service modeling purposes the editor component interacts with the object model, and, in turn, the object model stores and loads information from file system. Matching logic retrieves information from the object model and provides matching results to the presentation logic. Presentation
logic controls all components of presentation tier. View changes and changes of
settings, i.e. changing thresholds or weights, on presentation level perform interac-
tion with the logic layer.

The architecture of the service preselection tool has been explained above. Next,
tool functionalities are presented.

7.2 Service preselection tool functionality

The service preselection tool contains three main functional areas: (1) creating,
loading, editing and saving SCMM based models, (2) matching service models
and visualizing matching results, and (3) adjusting thresholds and weights. This
chapter starts with a general introduction about the tools user interface (cf. 7.2.1)
followed by two subchapters about creating, loading, selecting, editing and saving
SCMM based models (cf. 7.2.2 and 7.2.3). Afterwards, there are four subchapters
about visualizing matching results. These four subchapters are according to the
visualization techniques mentioned in chapter 6, i.e. table view (cf. 7.2.4), bar
view (cf. 7.2.5), scatter plot view (cf. 7.2.6), and radar view (cf. 7.2.7). Last not
least, adjusting weights and thresholds within the tool is explained at the end (cf.
7.2.8).

7.2.1 General

After opening the service preselection tool there is a typical Eclipse based inter-
face including a menu and a toolbar. On the top right one can select the perspec-
tive. A perspective is a container for a set of views [Springgay 2001]. In order to
use the service preselection tool one has to choose the "Service Preselection Tool"
perspective. After selecting the perspective all service preselection tool views are
loaded into the main window as shown in Figure 7.2. These views can be split into
three areas. The first area is the service selection area on the right hand side. There
are two views, one view for business service and another one for electronic ser-
vice. In this area business and electronic service models can be loaded and se-
lected. Details about loading and selecting are provided in subchapter 7.3.2. The
second area, i.e. the view selection area, is located on top of the main window and
contains six buttons and one pull down menu. The buttons are used for switching
views. There are six views available: tree view, table view, bar view, scatter plot
view, radar view, and weight and threshold settings view. The pull down menu is
used for selecting details level of particular views. All views will be explained in
subchapter 7.2.3 to 7.2.8 in detail. Views are shown in the main area, which is
underneath the view selection area. The three areas are marked by red rectangle in
Figure 7.2.
7.2.2 Creating and loading SCMM based models

Creating a new business or electronic service model from scratch is done by a typical Eclipse project and file creation process. During file creation one has to select the file type SCMM with file extension .scmm. Afterwards a new model is created and the editor view of the model is opened. This is the same view as the tree view, which will be mentioned in the following subchapter. Details about project and file creation and about EMF editors can be taken from [D’Anjou et al. 2005] and [Steinberg et al. 2009].

Loading an existing SCMM based model from file system is done by clicking the load button of the business service view (cf. (1) in Figure 7.3) or the electronic service view (cf. (2) in Figure 7.3) respectively. After selecting models from file system both appear in alphabetic order. When selecting one particular business service matching algorithms are triggered for each electronic service combination. Matching values on service entity property level (cf. 5.5.1), service entity level (cf. 5.5.2), service concept level (cf. 5.5.3) and service level (cf. 5.5.4) are calculated. The overall matching result, i.e. matching value on service level, is shown in the electronic service view next to the electronic service name. The electronic services are ordered by their overall matching value.
7.2.3 Tree view

Tree view is used for showing details about a particular service model. Further, it is used for creating and editing service models as well. For showing the tree view one has to select a business or an electronic service (in the service selection area) and click on the tree view button (in the view selection area). This will show the selected service model in the main area similar to Figure 7.4. In case of Figure 7.4 there are two service models in split window side by side. On the left hand side there is a business service model and on the right hand side there is an electronic service model. The tree view represents the basic EMF editor for editing models based on Ecore meta-models (cf. [Steinberg et al. 2009]). As shown in Figure 7.4 siblings, parents and child relations are presented in terms of tree nodes. New siblings or child nodes can be created by right-click on a particular node. Since the editor takes care of being conform to the meta-model definition, i.e. the SCMM, particular nodes can be added only. Selecting a node shows information about that node, i.e. node properties, within the properties area. Here one can add property information. After having finished editing a particular service model one can save the model, which means that the service model is serialized into XML and stored on the file system. One can create service models from scratch in a similar way.
7.2.4 Table view

Table view represents table visualization as mentioned earlier (cf. 6.4.1). Table view shows matching results on all level of details, i.e. on service level, on service concept level, on service entity level and on service entity property level. In order to show the table view one has to select a business service and a particular electronic service in the service selection area. Afterwards one has to select table view button (in view selection area) and the table view appears as shown in Figure 7.5. This view is according to the table visualization as presented in chapter 6.4.1. The overall matching value is shown on top. Matching results on service concept level and all other levels are presented in tables, i.e. one table for each service concept. For instance, there is the AbstractService table, which contains all abstract service entities from the business service model on the left hand side, matching results on different level in the middle, and all abstract services entities of the electronic service on the right hand side. Each abstract service has its own row and contains name, description and labels (in curly brackets) of the entity. The results presented in the table range from matching values on service concept level in the top row (cf. (1) Figure 7.5), matching values on service entity level (cf. (2) Figure 7.5), and matching values on service entity property level (cf. (3) Figure 7.5). Further, matching results for each entity combination can be shown by collapsing a particular table row using the ‘+’ button (cf. (4) Figure 7.5).
7.2.5 Bar view

Bar view is according to the bar chart visualization in chapter 6.4.1. Similar to table view one has to select a business service and an electronic service first. Second, one has to click on bar view in the view selection area. The bar view shows matching results on service level, i.e. overall matching result, on service concept level and on service entity level. Matching results on service level and service concept level are shown in Figure 7.6. The overall matching result is depicted as a straight horizontal line. Each bar represents a particular service concept matching value. Further, each bar contains a tooltip, which shows matching result as a number value. Additionally, there is a horizontal line that indicated the threshold.
In order to show matching results on service entity level, one has to choose a particular service concept using the pull down menu in the view selection area. Afterwards, matching results on service entity level are illustrated as shown in Figure 7.6. On the x-axis there are the entities from the business service, each bar represents one entity combination, and the y-axis shows the matching value for that combination. Further, there is an horizontal line, which shows the matching value on service concept level (cf. "Average Matching Value" in Figure 7.7), and there is a horizontal line for threshold illustration. In case of Figure 7.7 the AbstractService service concept has been selected. There are the AbstractService entities OrderManagement and Transportation from the business service model and the AbstractService entities OrderManagement and TransportManagement from the electronic service model. Thus, there are four combinations leading to four bars in the chart. Two of the bars are above the threshold, but two of them are below the threshold. Matching values below threshold are highlighted red. The tooltip for each bar indicates the matching value as a number and the entity name on electronic service side (cf. "Transportation: 0.34" in Figure 7.7). Further, as introduced in chapter 6.4.1 (cf. Figure 6.3) already, there is a black dot illustrating the average matching value for each entity combination of a particular business service entity.
7.2.6 Scatter plot view

Scatter plot view is according to the scatter plot visualization explained in chapter 6.4.2. Scatter plot visualization is used for analyzing and comparing matching results of two BS-ES combinations, where the business service of both combinations is the same. So, in order to apply scatter plot view one has to select one business service and two electronic services. Afterwards one has to select scatter plot view in the view selection area. As shown in Figure 7.8 matching results of one combination are presented as values on the x-axis, and matching results of the second combination are presented as values of the y-axis. Similar to Figure 6.4 in chapter 6.4.2 scatter plot view shows a diagonal line and thresholds as horizontal and vertical lines. This divides the coordinate system into six areas with distinct meaning. Details about these areas had been explained in chapter 6.4.2 already. Scatter plot view in Figure 7.8 compares matching values on service level and on service concept level. In order to compare matching values on service entity level one has to select a particular service concept using the pull down menu in view selection area. As an example, Figure 7.9 shows matching results of Ontology-Concept entities of both combinations. Each point of the scatter plot coordinate system has a mouse over tooltip, which contains the name of the entity (or the service concept respectively) as well as its x-value and its y-value as a decimal number (cf. "StandardPackage (0.409, 0.213)" in Figure 7.9).

The overall matching value on service level is depicted by a red rectangle.
Figure 7.8 Scatter plot (Overall)

Figure 7.9 Scatter plot (OntologyConcept)
7.2.7 Radar view

Radar view seeks in analyzing several BS-ES combinations at the same time. This is according to radar view visualization techniques explained in chapter 6.4.3. So, in order to invoke radar view one has to select one business service and two or more electronic services. Further, one has to click on the radar view button in the view selection area. Radar view appears in the main area of the service preselection tool as shown in Figure 7.10. Here, each axis represents a particular service concept. Matching values refer to a particular point of the corresponding axis. Further, neighboring points of a particular combination are connected, which results in a polygon with a distinct area. Each BS-ES combination is represented by a distinct color. Figure 7.10 shows matching results on service concept level only. In order to compare matching values on service entity level (as required in 6.4.3) one has to select a particular service concept in the pull down menu in view selection area. As an example, Figure 7.11 shows matching results of OntologyConcept entities of three combinations. Each point of the radar view has a mouse over tooltip, which contains the name of the corresponding electronic service entity as well as the matching value for this particular combination (cf. "ShipmentRequest (0.81)" in Figure 7.11).

Figure 7.10 Radar chart (Overall)
After having explained all views for matching result visualization, a view for configuring weight and threshold properties is introduced next.

### 7.2.8 Weight and threshold configuration

Weights and thresholds are an integral part of the matching solution (cf. chapter 5.5). Since it is important to adjust weights and thresholds in order to emphasize or to hide certain aspects, weight and threshold settings can be configured within the service preselection tool. To do so, one has to select the Weight/Threshold button in the view selection area - the configuration view appears as shown in Figure 7.12. Within this view there is a list with several key-value pairs. Each key represents the label of a particular configuration property and each value represents its decimal number. Basically there are three groups of properties.

- **Thresholds.** Thresholds are used in different parts of the matching solution (cf chapter 5.5). The overall threshold is labeled as "NEIGHBOR_THRESHOLD" and can be configured by a decimal number between 0 and 1.
- **Weights for service level aggregation.** Weight for service level aggregation had been mentioned in chapter 5.5.4. Service level aggregation weights are labeled "OVERALLWEIGHT_[SERVICECONCEPTNAME]". Thus, there is one weight for each service concept. The value for each weight can be a decimal number between 0 and 1. However, the sum of all service level aggregation weights must not exceed 1.
Weights for service entity aggregation. Weight for service entity aggregation had been mentioned in chapter 5.5.2 and in Table 5.38. These weights start with the name "WEIGHT_". There is one weight group for each service concept, containing weights for each service concept property. The value for each weight can be a decimal number between 0 and 1. However, the sum of all weights in one group, i.e. for one particular service concept, must not exceed 1.

Beside the configuration of weights and thresholds, configuration settings can be save and loaded. This can be done by using the buttons in the top right corner (cf. red rectangle in Figure 7.12). After adjusting the settings and clicking the save button, matching algorithms are invoked once again. All matching results are recalculated based on the new configuration settings.

![Figure 7.12 Weight and threshold configuration view](image)

7.3 Summary

This chapter was dedicated to tool support of the overall approach. Tool support is needed for modeling services, for applying matching algorithms and for matching result visualization. The service preselection tool, which had been introduced here, supports each topic. It combines results from different areas of this thesis and provides applicability of the overall approach from one source. This chapters explained the tool architecture and the tools' functionality from user point of view. The tool is a prerequisite for validation purposes. Validation is examined in the next chapter (cf. chapter 8).
8 Validation

This chapter is dedicated to the validation of the matching approach as required (cf. [Hevner et al. 2004]). Validation is defined as a “process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements” [IEEE 1990]. A related term of validation is verification. In contrast to validation, verification is defined as “a process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase” [IEEE 1990]. Rather than verifying, this chapter is focused on validation, i.e. whether the approach satisfies defined requirements.

Before validation, it should be noted that assumptions, models, methods, and techniques, which are constituents of the matching approach, are rather young and topics of ongoing research. Among others, this includes:

- The representation of companies business in terms of business services and the representation of software functions in terms of electronic services.
- Service repositories that include a large amount of business service and electronic services.
- Domain specific linguistic databases that support matching algorithms as external resources.

Service representation and service modeling has been addressed in chapter 3 and chapter 4. Since there is no intersubjective service meta-model and no intersubjective method for modeling services that fits this thesis, the SCMM and method for BSM and ESM modeling has been derived from existing approaches. Although both have already been applied successfully (cf. chapter 4), neither the SCMM nor the method have been used in large scale so far. This leads to the fact that there is no service repository existing in real world business environments, which contains large amounts of BSM and ESM. So validating the matching approach cannot be based on existing data, and, thus, has to be rather explorative and based on use case scenarios.

Scenarios used for thesis validation have been developed within InterLogGrid project together with major industrial partners (cf. [InterLogGrid 2011]). InterLogGrid is a research project, which aims at applying grid-computing [Foster/Kesselman 2004] for logistics companies. This comprises providing logistics services running on a grid-based infrastructure. In course of this project, project members created a logistics partner handbook [Netzwerk Logistik Leipzig-Halle e.V. 2010]. The handbook includes competences and resources of 27 logistics companies of Middle Germany44. Further, InterLogGrid project members made a survey about used software products within logistics companies. Results of the survey are presented in [Arnold et al. 2010]. The partner handbook and the

---

44 Middle Germany is a region in Germany comprising the states Saxony, Saxony-Anhalt and Thuringia. Middle Germany is one of the prospering logistics regions of Germany [Jünemann 2005].
logistics software product survey provide the basis for deriving logistics grid services within InterLogGrid. Use case scenarios provided here are conducted with both reports. Logistics grid services from InterLogGrid are adapted and used for validation.

This chapter is structured as follows. Chapter 8.1 introduces use case scenarios. Based on that one BS and nine ES are required, which are derived from real world and explained in chapter 8.2. Afterwards, matching algorithms are applied according to the use case scenarios. This leads to matching results, which are interpreted and analyzed in chapter 8.3. This chapter ends with a summary (cf. chapter 8.4), which recaps main results.

8.1 Use case scenarios

This section provides explanation of the use case scenario. Within the scenario there is one business service and many electronic services. The objective is to find the ES from a set of ESs that supports the BS. There are three different scenarios:

1. Find the ES from a set of ESs that fit the concerns of the BS. The set of ESs contains one ES from logistics domain and two ESs from other domains. This scenario is applied and analyzed in 8.3.1.
2. Find the ES from a set of ESs that fit the concerns of the BS. The set of ESs contains three ESs from logistics domain. However, two of them are different to the BS use case, and one of them is similar to the BS use case. This scenario is applied and analyzed in 8.3.2.
3. Show differences between four ESs that are similar to the BS use case, but are different in naming, structure and their features. This scenario is applied and analyzed in 8.3.3.

Based on these scenarios required BS and the ESs are explained below.

8.2 Use case services

According to the three scenarios nine services are needed. First of all, a business service from logistics order management is introduced in section 8.2.1. For the first scenario, one ES from the logistics domain and two ESs from other domains are need. The two ES from other domains are acquired from customer relationship management software product (cf. section 8.2.2) and from an enterprise resource planning service (cf. section 8.2.3).

For the second scenario, three services from logistics domain are needed. For this purpose three ES are acquired from SAP® TM. First, there is one ES that is concerned with invoice management (cf. 8.2.4). Second, there is a transport planning ES (cf. 8.2.5). And third, there is a transport order management ES (cf. 8.2.6).
The third scenario is concerned with services that show similar use cases. Since the BS relates to order management, there are three modification of the transport order management ES introduced in section 8.2.7 to 8.2.9. The first modification is renaming all service, stakeholder, actions, resource, concepts and properties. The resulting ES is presented in 8.2.7. The second modification is restructuring some concepts of the transport order management ES, which is shown in 8.2.8. The third modification is, removing feature from the order management ES (cf. 8.2.9).

Each service is introduced with a short text and a figure. However, the figure shows just some sketchy details. Full service description, which is according to the SCMM, is provided in tables of Appendix E to Appendix M. Furthermore, there is a XML representation of each service, which results as an output of the EMF model editor.

### 8.2.1 Business service: OrderInquiry

Order Inquiry is a business service that is provided by a fictional logistics company. The company provides transport services. Beside of physical transportation a transport service contains several sub-services, such as order management, route planning, accounting, etc. As a representative of order management Order Inquiry is concerned with receiving, processing and confirming transport requests from customers. Order Inquiry starts with receiving a request from a customer, processing this request and sending a confirmation message back to the customer.

Applying the explanation to the concepts defined in SCMM, OrderInquiry is a concrete service that belongs the AbstractService OrderManagament. In turn, OrderManagement belongs to the AbstractService Transport. OrderInquiry comprises two actions called ReceiveTransportRequest and ConfirmTransportRequest. The first action creates a TransportRequest. TransportRequest is a ImmaterialResource and further detailed by a TransportRequest ontology concept. The second action processes the TransportRequest, creates a ConfirmationMessage and sends this message to the customer. ConfirmationMessage is a ImmaterialResource as well. Further details of ConfirmationMessage is provided by the ConfirmationMessage ontology concept.
The services as well as action and resources are illustrated in Figure 8.1. Due to lack of space, details about their synonym and descriptions as well as details about stakeholders, ontology concepts and properties are provided in Appendix E (Table E.1 to Table E.7). This information has been acquired from [Gudehus 2010] and [Logitrans GmbH 2011].

### 8.2.2 Electronic service: CustomerEntry

Customer Entry is an electronic service adopted from the software product Microsoft Dynamics CRM [Microsoft Corporation 2011b, Microsoft Corporation 2011a]. Thus, CustomerEntry belongs to the AbstractService CustomerRelationshipManagement. The service contains an action called EnterCustomerData, which has a CustomerData object as an output. The service is depicted in Figure 8.2 briefly. Details about synonyms and description of all concepts defined in the SCMM a provided in Appendix F (Table F.1 to Table F.7).
8.2.3 Electronic service: SendInvoice

SendInvoice is an electronic service that has been adopted from a Web service called Invoice Place [Invoice Place 2007]. Details about the service structure had been taken from OpenERP [OpenERP 2011]. OpenERP is a Enterprise Resource Planning software product. SendInvoice is ConcreteService, which is related to the AbstractServices Invoicing and EnterpriseResourcePlanning. SendInvoice has three actions: CreateInvoice, CompleteInvoice and SendInvoice. Each Action processes the invoice data. This service is depicted in Figure 8.3. Details about the ES SendInvoice is provided in Appendix G in tabular form (Table G.1 to Table G.7).

![Figure 8.3 Electronic service: SendInvoice](image)

8.2.4 Electronic service: SendCustomerInvoice

SendCustomerInvoice is an electronic service from SAP® TM [SAP AG 2011b] and, hence, related to logistics. The service description is adopted from SAP® TM graphical user interface and [Lauterbach et al. 2009]. SendCustomerInvoice belongs to the AbstractService InvoiceMangement, which, in turn, belongs to TransportManagement. SendCustomerInvoice is a ConcreteService and contains four actions: CreateCustomerInvoice, ProcessCustomerInvoice, CompleteCustomerInvoice and SendCustomerInvoice. Each of these actions is concerned with the CustomerInvoice data object, which is an ImmaterialResource. A brief overview of SendCustomerInvoice is provided in Figure 8.4. Details about this service is provided in tabular form in Appendix H (Table H.1 to Table H.7).
Electronic service: PlanTransportationActivities

PlanTransportationActivities is an SAP® TM electronic service as well. PlanTransportationActivities belongs to Planning within TransportManagement. According to [Lauterbach et al. 2009] there is one action that is concerned with the creation of a TransportationPlan taking available FreightUnits, VehicleResources and TransportationResource into account. A brief overview of PlanTransportationActivities is provided in Figure 8.5. Details are provided in Appendix I (Table I.1 to Table I.7).

Electronic service: OrderReceipt

OrderRequest is the third electronic service that is derived from SAP® TM [Lauterbach et al. 2009]. OrderRequest belong to OrderManagement and contain four actions: CreateShipmentRequest, ProcessShipmentRequest, ActivateShipmentRequest, and ConfirmShipmentRequest. Each of these actions works on the ShipmentRequest data object, which is defined as an ImmaterialResource. A brief
overview of OrderReceipt is given in Figure 8.6. Details are provided in Appendix J (Table J.1 to Table J.7).

Figure 8.6 Electronic service: OrderReceipt

The next three Services (cf. 8.2.7 to 8.2.9) are modification of the OrderReceipt electronic service.

8.2.7 Electronic service: OrderReceipt (renamed)

As mentioned before the electronic service presented here is a modification of OrderReceipt (cf. 8.2.6). In order validate matching algorithms and the tool this service has same functionality as OrderReceipt, but uses different names. For example, the resource ShipmentRequest is called TransportationRequest. Further, actions CreateShipmentRequest, ProcessShipmentRequest, ActivateShipmentRequest and ConfirmShipmentRequest are renamed into EnterTransportationRequest, ProcessTransportationRequest, CompleteTransportationRequest, and ConfirmTransportationRequest. The service is illustrated in Figure 8.7. Details such as Stakeholder, OntologyConcepts, ObjectProperties, and DataProperties as well as their names, label and descriptions are provided in Appendix K (Table K.1 to Table K.7). As shown in the Appendix K names and descriptions are different to Appendix J. Differences in names had been gained from synonym words used in [Lauterbach et al. 2009] and from the WordNet [Princeton University] synonym database.
8.2.8 Electronic service: OrderReceipt (renamed)

This electronic service is a restructured modification of OrderReceipt (cf. 8.2.6). Restructuring means that concepts are removed, added or substituted by other concepts. For instance, the action ActivateShipmentRequest is removed and substituted by the action ConfirmShipmentRequest, which comprises confirmation and activation as well. This is illustrated in Figure 8.8. However, there are more modifications. The modification are: First, a stakeholder called OrderManager is added. Second, the ObjectProperties hasGrossWeight and hasGrossSize from ItemData and their connected OntologyConcepts GrossWeight and GrossSize are removed. Instead, the DataProperties of GrossWeight and GrossSize (i.e. hasWeightValue, hasWeightUnit, hasBreadth, hasLength, hasHeight, and hasSizeUnit) are connected to ItemData, directly. Third, the ObjectProperty hasBusinessPartner is removed. Instead, business partners are connected directly via hasShipper, hasConsignee and hasCustomer including their corresponding OntologyConcepts Shipper, Consignee, and Customer. Fourth and fifth, the ObjectProperties hasRequestedDates and hasAcceptableDate including their corresponding OntologyConcepts are removed. Instead, corresponding DataProperties, i.e. hasDateFrom, hasDateTo, hasEarliestDate and hasLatestDate, are connected with PickUpDeliveryPeriod directly. Details about this service are provided in Appendix L (Table L.1 to Table L.7).
8.2.9 Electronic service: OrderReceipt (modified feature)

In the third version of the OrderReceipt basic structure and naming is the same as described in 8.2.6. However, the ES in 8.2.6 does provide support for four ways of transportation (i.e. truck, ship, air, and rail). So the modification is that this service supports one way of transportation only. Sea cargo, air cargo and rail cargo are not provided anymore in this case. Thus, the VehicleResources Ship, Aircraft, and Train are removed. Since this modification is done on OntologyConcept level, the brief illustration of the ES (cf. Figure 8.9) is the same as the illustration provided in 8.2.6 (cf. Figure 8.6). However, details are provided in Appendix M (Table M.1 to Table M.7).

8.3 Application and analysis

This section is dedicated to the application of the three scenarios mentioned in 8.1. Each scenario starts with an introduction of the involved services and state expected result. Afterwards, matching algorithms are applied using the preselection
tool. Next, results are analyzed. If all results meet the expectations the approach is regarded as validated.

### 8.3.1 Electronic service from different domains

Within the first use case scenario there is the BS OrderInquiry (cf. 8.2.1) from logistics domain and there are three ES, which are from different domains each. The first ES is CustomerEntry (cf. 8.2.2) from customer relationship management. The second ES is SendInvoice (cf. 8.2.3), which has been derived from an enterprise resource planning software product. SendInvoice belongs to the accounting domain. Third, there is ES OrderReceipt (cf. 8.2.6) from the logistics domain. The objective is to apply the matching algorithms from chapter 5 using the tool from chapter 7 and analyze matching results. The expected result is that the ES from logistics has higher matching values than the two non-logistics ESs.

Applying the matching algorithms leads to the matching results shown in the table below. Table 8.1 shows the BS OrderInquiry in the upper top row and the three ES (i.e. CustomerEntry, SendInvoice and OrderReceipt) in the lower top row. The table has four columns. The first column contains the label of the matching value. This table contains two types of matching values. The overall matching value is presented in the first row. All matching values on service concept level are shown in row two to nine. Column two to four contain the matching values of the BS-ES-combinations.

<table>
<thead>
<tr>
<th>Business service:</th>
<th>OrderInquiry (cf. 8.2.1)</th>
<th>CustomerEntry (cf. 8.2.2)</th>
<th>SendInvoice (cf. 8.2.3)</th>
<th>OrderReceipt (cf. 8.2.6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>0.19</td>
<td>0.18</td>
<td>0.64</td>
<td></td>
</tr>
<tr>
<td>AbstractService</td>
<td>0.00</td>
<td>0.00</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>ConcreteService</td>
<td>0.00</td>
<td>0.00</td>
<td>0.90</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>0.00</td>
<td>0.00</td>
<td>0.65</td>
<td></td>
</tr>
<tr>
<td>Resource</td>
<td>0.00</td>
<td>0.00</td>
<td>0.42</td>
<td></td>
</tr>
<tr>
<td>OntologyConcept</td>
<td>0.21</td>
<td>0.14</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td>ObjectProperty</td>
<td>0.34</td>
<td>0.29</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>DataProperty</td>
<td>0.61</td>
<td>0.60</td>
<td>0.61</td>
<td></td>
</tr>
<tr>
<td>Stakeholder</td>
<td>0.33</td>
<td>0.39</td>
<td>0.75</td>
<td></td>
</tr>
</tbody>
</table>

*Table 8.1 Overall and service concept level matching results (ES from different domains)*

The overall matching results illustrated using continuum visualization (cf. 6.2.1) is shown in Figure 8.10.
As shown in the table and the figure above, ES OrderReceipt from the logistics domain has the highest overall matching value. This meets the expectation, i.e. a BS from the logistics domain and an ES from the logistics domain have higher matching than ES from other domains. Service concept level matching results from Table 8.1 are illustrated using radar chart visualization (cf. 6.2.8). As shown in Figure 8.11 almost all matching values on service concept level of ES OrderReceipt (red color) are higher than the values of ES CustomerEntry (green color) and ES SendInvoice (blue color). While the ES OrderReceipt matching values show results >0 for all service concept, the matching value of the other two ES show zero results. This is due to the fact that matching values on service concept level are calculated using a weight average algorithm that takes values in to account, which are above a specific threshold only. Since the threshold is 0.4 and all matching values on AbstractService level, ConcreteService level, Action level and Resource level are below 0.4, the weight average value is zero for these service concepts.

Although the overall result is clear, the matching values on DataProperty level are very close to each other. With other words, the matching results of ES CustomerEntry and ES SendInvoice are as high as the matching results of ES OrderReceipt
on DataProperty level. This is due to the fact that DataProperties, such as hasCity, hasZip, hasCountry, hasEmail, etc., are more general and less domain specific. Hence, it is likely that DataProperties of the BS appear in ES CustomerEntry, ES SendInvoice and ES OrderReceipt as well. Instead, ObjectProperties, Ontology-Concept and others are more domain specific and, thus, matching values for these differ more than matching values on DataProperty level.

8.3.2 Different electronic services from logistics domain

In the second use case scenario, there is the BS OrderInquiry (cf. 8.2.1) and there are three ES, which are from logistics domain. The first ES is SendCustomerInvoice (cf. 8.2.4), the second ES is PlanTransportationActivities (cf. 8.2.5), and the third ES is OrderReceipt (cf. 8.2.6) once again. All three ES are derived from the same software products (i.e., SAP® TM), and, thus, belong to the same domain. However, each of the ES provides different functionality. The expectation by applying the matching algorithm is that the BS-ES combination with similar functionality (i.e., BS OrderInquiry and ES OrderReceipt) has higher matching values.

Applying the matching algorithms leads to the matching results shown in Table 8.2. The structure of Table 8.2 is similar to Table 8.1.

<table>
<thead>
<tr>
<th>Business service:</th>
<th>OrderInquiry (cf. 8.2.1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic service:</td>
<td>SendCustomerInvoice (cf. 8.2.4)</td>
</tr>
<tr>
<td>Overall</td>
<td>0.28</td>
</tr>
<tr>
<td>AbstractService</td>
<td>0.59</td>
</tr>
<tr>
<td>ConcreteService</td>
<td>0.00</td>
</tr>
<tr>
<td>Action</td>
<td>0.00</td>
</tr>
<tr>
<td>Resource</td>
<td>0.00</td>
</tr>
<tr>
<td>OntologyConcept</td>
<td>0.25</td>
</tr>
<tr>
<td>ObjectProperty</td>
<td>0.43</td>
</tr>
<tr>
<td>DataProperty</td>
<td>0.60</td>
</tr>
<tr>
<td>Stakeholder</td>
<td>0.41</td>
</tr>
</tbody>
</table>

| Table 8.2 Overall and service concept level matching results (different ES from same domain) |

The overall matching results from Table 8.2 illustrated using continuum visualization (cf. 6.2.1) leads to Figure 8.12.
As shown in table and figure above, ES OrderReceipt has the highest overall matching result, i.e. 0.64, with the BS OrderInquiry. Since both services deal with the inquiry of transportation requests, i.e. BS OrderInquiry and ES OrderReceipt provide similar functionality, the result meets the expectations. Analyzing matching results using radar chart (cf. 6.2.8) on service concept level leads the Figure 8.13.

All matching values on service concept level of ES OrderReceipt (red color) are higher than the values of ES SendCustomerInvoice and ES PlanTransportationActivities. The matching values on AbstractService level of all given ESs is higher and closer to each other than previously (i.e. in 8.3.1). This is natural since all ESs here are from the same software product and, thus, share the same root AbstractService “TransportManagement”. Instead, the matching value on ConcreteService level is quite separated. The matching result on OntologyConcepts level, ObjectProperties level and DataProperties level are very close related once again. The explanation is similar to the explanation in 8.3.1. DataProperties such as “hasCity”, “hasCountry” and “hasZip” for an address are commonly used across services. Thus, matching values are very close to each other. Further, OntologyConcept
and ObjectProperty are very close related since all given ES are from the same software product and, thus, share similar concepts.

Figure 8.14 shows matching results on OntologyConcept level, for instance. There are general concepts such as AddressData, SourceAddress and DestinationAddress that can be found in all ES. Further, there are logistics specific concepts, such as AirCargo, SeaCargo, TruckComplete, that can be found in two ESs. These services deal with logistics and, thus, there are similar OntologyConcepts even though functionalities of ESs are different. The difference in functionality is indicated by the differences of matching values on Action level, Resource level, and Stakeholder level.

8.3.3 Similar electronic service from logistics domain

In the third use case scenario there is the BS Order Inquiry (cf. 8.2.1) and there are three modifications of the ES Order Receipt (cf. 8.2.6). The first modification is renaming all concepts (cf. 8.2.7). The second modification is restructuring certain concepts. As mentioned in 8.2.8, restructuring means that concepts are removed, added or substituted by other concepts. Last not least, the third modification is done by removing one distinct feature. As explained in 8.2.3 this is done by removing the concepts sea cargo, air cargo and rail cargo. Thus, this ES provides support for truck cargo transport only. The expectation is that the ES that is renamed and the ES that is restructured should have marginal different matching values than the original ES OrderReceipt, since they provided the same functionality. However, the original ES OrderReceipt and the ES with the modified feature should show differences since both are different in their features.

Applying the matching algorithms leads to the matching results shown in Table 8.3. The structure of Table 8.3 is similar to Table 8.1 and 8.2.
Matching ES OrderReceipt and its renamed version with the BS OrderInquiry lead to matching results shown in column two and three of Table 8.3. The overall matching results are almost the same (0.63 vs. 0.63). Scatter plot visualization (cf. Figure 8.15) of matching values on service concept level shows that these values are very close to diagonal line. This indicates that they are very close to each other. There are only marginal differences due to the fact that names are different. The meaning of concepts is still the same, i.e. concept synonyms and description are similar and relations are the same as well, which leads similar matching results. Matching results meet the expectation.

Matching ES OrderReceipt and its restructured version with the BS OrderInquiry lead to matching results shown in column two and four of Table 8.3. The overall...
matching results are very close (0.63 vs. 0.65). Further, in scatter plot visualization (cf. Figure 8.16) all values are very close to diagonal line once again. There are only marginal differences due to the fact that there are some structural differences. However, the meaning of concepts is still the same, i.e. concept synonyms and description are the same and relations are similar, which leads similar matching results. Matching results meet the expectation.

Matching ES OrderReceipt and its missing feature version with BS OrderInquiry lead to results shown in column two and five of Table 8.3. As shown in the table, there is a difference on OntologyConcept level (cf. 0.47 vs. 0.41), which results from missing concepts Ship, Aircraft, and Train in the missing feature version of ES OrderReceipt. However, without knowing that these features are missing the difference is not easy to detect, since the overall matching result is similar (cf. 0.63 vs. 0.62). All other matching results on service concept level are the same. Thus, when ES are similar in their functionality one has to analyze concepts on different level in detail in order to detect differences. In this use case scenario radar view on OntologyConcept level provides a good visualization for that purpose. As shown in Figure 8.17 missing OntologyConcepts are illustrated with as lower matching values.
8.4 Summary

The validation of the matching approach is based on three use case scenarios. Basically, there is one BS and there are several ES within each scenario. Scenario one contains ES from different domains. Scenario two contains ES from logistics domain with different functionality. Scenario three contains ES from logistics with similar functionality. Required services for the use case scenario had been derived using the method mentioned in chapter 4. Furthermore, BS and ES are in accordance to the logistics partner handbook of InterLogGrid (cf. [Netzwerk Logistik Leipzig-Halle e.V. 2010] and [Arnold et al. 2010]). Each use case scenario defines certain expectations. After applying the matching algorithms according to the use case scenarios, matching results of each BS-ES combination had been analyzed in detail. The expectation had been met in each case. Consequently, the matching approach is validated successfully. However, since matching results can be influenced by different circumstances (e.g. proper description, weight settings, threshold settings, external resource, etc.), variations are possible. This will be discussed in chapter 9.3 and within the future work section (chapter 9.4).
9 Conclusion and outlook

This is the final chapter of the thesis. This chapter provides the conclusion and outlook. First of all there is a summary in order to recap the overall approach (cf. 9.1). Second, section 9.2 highlights main contribution. Third, findings of the thesis are discussed critically in section 9.3 and, lastly, section 9.4 addresses open topics and further research.

9.1 Summary

This thesis studied the matching of business service and electronic service by measuring the heterogeneity of their descriptions. The thesis is based on the assumption that the lower the heterogeneity between business and electronic service descriptions the higher the functional match between them. With other words, lower heterogeneity leads to higher probability that the electronic service supports the business service in its function.

There are three different kinds of heterogeneity: syntactic, terminological and semiotic heterogeneity. Syntactic heterogeneity occurs when using different language and different syntactic rules during service description, terminological heterogeneity occurs by differences in perspectives, granularity and coverage, and semiotic heterogeneity occurs by different interpretation of service descriptions. Syntactic heterogeneity can be prevented by defining common syntactic rules prior to the service description. Terminological heterogeneity cannot be prevented per se, but can be measured. Semiotic heterogeneity is out of the scope of this thesis.

After the introduction chapter about the research area and research methodology, chapter 2 starts with background knowledge about business and electronic service definitions, service-orientation, and the research framework this thesis is embedded in. Further, chapter 2 includes motivation, assumptions, general requirements and the conceptual approach. The conceptual approach explains main constituents of research and provides the thesis’ pathway.

One of the general requirements states that business services and electronic service have to be described based on the same syntactic rules in order prevent syntactic heterogeneity. Based on this requirement, chapter 3 is dedicated to service representation. First, two general service representation techniques, i.e. models and ontologies, are examined. Second, with a focus on functional aspects existing service representation approaches, i.e. existing conceptual service models and service ontologies, are examined in detail. Third, based on distinct requirements and based on the existing approaches a service concept meta-model (SCMM) had been derived. The SCMM provides the basis for service descriptions within this thesis.
Business and electronic service need to be described according to the SCMM. So chapter 4 is dedicated to the question: How to derive information for business (BSM) and electronic service models (ESM)? This question regards methodologies, methods, and techniques for service modeling, which are defined first in chapter 4. Afterwards, existing methodologies from ontology engineering are examined in detail. As a result there is a method for BSM and ESM modeling. This method had been applied within a use case. Chapter 4 ends with a section about tool support for service modeling. Eclipse Modeling Framework (EMF) has been identified as a proper tool for modeling BSM and ESM.

Measuring terminological heterogeneity is addressed in chapter 5. Measuring terminological heterogeneity relies on existing ontology matching algorithms. There are basic matching algorithms and matching strategies. Both had been examined in detail, i.e. each algorithms has been proven with an example from logistics. After a detailed evaluation of matching algorithms, chapter 5 ended with a proposed matching algorithm for SCMM based service models. The matching algorithms is core of the terminological measuring solution.

Since the application of matching algorithms leads to results that cannot be interpreted easily, visualization of matching results is necessary. Chapter 6 provides techniques for visualization. First, visualization and general techniques had been introduced. Second, visualization techniques had been evaluated according to distinct requirements. According to this proper visualization techniques have been selected. Further, the application of the visualization techniques had been proven using sample data. Furthermore, sample interpretations and analysis for different scenarios had been provided.

Tool support has been introduced in chapter 4 already. However, a tool that encompasses service modeling, the application of matching algorithms and visualization is needed in order to apply the overall matching approach. Such tool has been implemented and presented in chapter 7. Main features of the tool are creating and editing of service models based on SCMM, applying matching algorithms including the adjustment of matching parameters (i.e. weights and thresholds), applying matching algorithms, and the application of visualization techniques.

In chapter 7 chapter the matching approach was validated. Based on three use case scenarios and nine use case services, which had been derived from real world business and real software products, service models, matching algorithm, visualization and tool had been demonstrated. Within each scenarios certain expectation had been defined previously. All expectation had been fulfilled, and, thus, the overall matching approach was validated successfully.

9.2 Main contribution

Preselection of electronic services by given business service based on measuring semantic heterogeneity contributes to service-oriented research area. More pre-
cisely, this thesis addresses service discovery, which is one topic of service-oriented research. This thesis presents an approach for measuring the heterogeneity between business services and electronic services using service matching algorithms. These algorithms result in matching values that can be aggregated, visualized and analyzed. With the assumption that a large amount of business services and electronic service exist, aggregation, vitalization and analysis of matching results facilitates decision making during electronic service selection. Given one particular business service and several electronic services, the amount of electronic services that possibly support that business service is reduced. This eases decision making during electronic service evaluation, i.e. the selection of electronic services that support a particular business service.

Main contribution of this thesis is the facilitation of electronic service evaluation by providing models, methods, techniques and tools:

- Models: This thesis provides a service concept meta-model (SCMM), which is focused on functional aspects and can be used for the representation of business and electronic services.
- Methods: This thesis provides a method for service modeling, which can be applied for business and electronic services as well.
- Techniques: This thesis provides two main techniques. First, there are algorithms for measuring heterogeneity between business and electronic services resulting in matching values on different levels. Second, this thesis provides techniques for visualization matching values on different level.
- Tools: This thesis provides tools for service modeling, for applying matching algorithms and for visualizing matching results.

Next, thesis results are discussed critically.

9.3 Discussion

This thesis contributes to service discovery topic, which is one topic of service-oriented research. The contributes has been stated above. However, the results and findings of the thesis should be discussed critically as well. So, this section is dedicated to critical voices concerning the assumptions and results of the thesis. Critical statements are listed and discussed below.

SI: “Business services cannot be conceptualized and formalized.”

This statement predicates that business services, i.e. real world services, such as hair cutting, plumbing and transportation, are unstructured, unpredictable and change often in operation, and, thus, cannot be formalized using a finite model. Though, it is true that most of the business services are unstructured so far, that it is hard to predict each case, and that operations of business services might vary. However, it is still possible to conceptualized and formalized business services using appropriated models, methods and modeling techniques. Existing models
had been mentioned in chapter 3 in detail. Further, a meta-model for business services that serve the purpose of this thesis had been derived as well.

S2: “Business services and electronic services cannot be described by using the same meta-model.”

Protectionist of this statement argue that business services, i.e. real world services, and electronic service, i.e. software, exist in two different worlds, and, thus, have to be modeled using different meta-concepts. Different meta-concepts lead to different meta-models. However, assuming that electronic services serve the purpose of a business services, assuming that both contain business semantics (cf. A4 and A5 in chapter 2.4), and assuming that electronic services need to reproduce business services in order to support business operations (cf. A7 in chapter 2.4), this statement holds not true. Further, as one of the basic assumptions of chapter 2.4, A6 states that business services and electronic services can be described base on the same conceptual model.

S3: “Terminological similarity between business services and electronic services does not mean that ES support the BS. There are ES that does not have terminological similarities, but still support BS.”

Protectionist of this statement could argue in two ways. First, business services and electronic services might have terminological similarities, but this does not mean that the electronic service supports the business service. Second, there are business services and electronic services that have no terminological similarities, but the electronic service still supports the business service. The first argument is covered by assumption A7 and justified by two independent statements (cf. chapter 2.4). It is assumed that the less the terminological heterogeneity of BS and ES the higher the probability that the ES supports the BS. Sure, it is hard to provided evidence for that. However, suppose that there is a BS that contains the term "invoice" in its description, it is quite logical that an ES that contains the term "invoice" as well, is more likely to support the BS than an ES that does not contain the term "invoice". The second argument states that there are ES that do not have terminological similarities, but still support BS. For instance, a hospital (BS) might use a warehouse management system (ES) for managing the occupancy of beds. Although this might be possible, there are several good reasons for hospital not to use a warehouse management system, but a hospital management system instead. Just to name a few reasons: stability of the system, extensibility of the system, maintenance of the system, running an oversized system on higher costs, running an undersized system that provides less functionality, integration with other systems, and more.

S4. “There many possibilities for a conceptual service models. A different service concept meta-model would lead to a different matching result.”

This thesis does not claim to provide the one-and-only meta-model for services. Although the SCMM is based on the evaluation of 10 existing service models (cf. chapter 3), there are viable alternatives. A different SCMM would lead to different BS models and ES models, to a different setup of matching algorithms, to differ-
ent matching parameters and, thus, to different matching results. These matching results would not be comparable with the results of this thesis. So, it is important to use one SCMM in order to get comparable results. It is not important which SCMM as long as it represents services properly.

S5. “There many possibilities for BSM and ESM description. Different services description will lead to a different matching result.”

This statement argues that different people describe same services differently, even if the services are according to the SCMM. These differences lead to different matching results. However, it is assumed that all BSM and ESM descriptions are in a similar range of granularity and level of detail. Further, it is assumed that BS and ES are described fully, honestly and realistically. Under these assumptions, matching results still differ, but not significantly. As shown in chapter 8.3.3 differences in structure and naming lead to similar matching results.

S6: “Different matching algorithms, different external resources, different weights, and different threshold settings lead to different matching results.”

This thesis does not claim to provide the one-and-only matching algorithm. Although the matching algorithms provided in this thesis are based on a detailed evaluation (cf. chapter 5), there are alternatives. Another matching algorithm would lead to a different matching results. However, matching results from different algorithms are not comparable with the results in this thesis. So, it is important to use one particular matching algorithm in order to get comparable results. This holds true for different weights and different thresholds. Although weights and thresholds can be modified easily using the service preselection tool (cf. chapter 7.2.8), it is important to use one setup for one particular analysis.

After this subchapter, where critical statement concerning assumptions and results of the thesis had been discussed, future work is presented next.

9.4 Further research

This section presents further research and continues work. As already mentioned (cf. chapter 8), basic constituents of the overall approach are rather young and topics of ongoing research. Among others this includes service representation in general including conceptual service models, methods and techniques used for eliciting service representation, algorithms for service matching, and, last not least, the application of the overall approach. Thus, continues work is subdivided into four areas: service representation, service modeling, service matching, and application.

Service representation

Research in the area of business service representation and electronic service representation is dedicated the question of how to structure and model companies
business and software in terms of services in general. This thesis assumes that it is possible to represent companies business as a set of business service (cf. A1 in chapter 2.4) and that it is possible to represent software as a set of electronic service (cf. A2 in chapter 2.4). However, there is no proof for these assumption yet. Further research may address these topic.

This thesis presents a service concept meta-model (SCMM), which can be used for business service and electronic service representation. The SCMM is based on an evaluation of 10 existing service models. However, since the statement “There is no one correct way to model […] - there are always viable alternatives.” [Noy/McGuinness 2001] holds true for service models as well, it would be interesting to know how the overall approach works with another SCMM. A different SCMM could be more specific, could be more general, or could highlight different aspects. So, future work may apply different meta-models and compare matching results between these models.

Service modeling

The thesis provides a modeling method, which is used for extracting information for business service models and electronic service models. This method has been adapted from ontologie engineering. However, other service modeling methods are possible. So, one of the next endeavors are the application of other modeling methods, the examination of resulting models and the comparison of matching results after applying matching algorithms with these models.

Further, there are techniques for automatic information extraction, e.g. text mining [Berry/Kogan 2010]. Such techniques might extract proper business service models from existing documents. Other techniques might extract information from existing software products automatically. For instance, it might be possible to gain OntologyConcepts, ObjectProperties and DataProperties from object models of existing software products. Research in this area is useful, since it is costly to gain proper business services models and electronic services models manually.

Another direction of research is collaborative modeling [Renger et al. 2008]. Through collaborative modeling different stakeholders take part in the modeling process in order to create a shared understanding about a model. As one of the next steps, research in collaborative service modeling is useful, since it might lead to a shared understanding of the service representation. Shared understanding means higher acceptance and higher stability.

Service matching

Service matching algorithms had been adapted from ontology matching. Since ontology matching is under ongoing research, there is ongoing research in service matching as well. There are various existing matching algorithms. New matching algorithms appear each year. Several matching algorithms had been introduces in chapter 3. Further research might apply different matching algorithms and compare matching results in order to prove stability and variability of matching results.
Beside the matching algorithms mentioned in this thesis there are learning methods, which have not been examined yet. Learning methods usually operate in two phases: (1) learning and training phase and (2) the matching phase. Within the first phase positive and negative results are sorted. This can be done manually. Within the second phase matching algorithms are applied using the information from the learning phase. In this way matching results can be improved by iteration, i.e. applying several learning/training and matching phases in combination. Well-known learning methods are Bayes learning [McCallum/Nigam 1998], WHIRL learning [Cohen/Hirsh 1998], neural networks [Li/Clifton 1994], decision trees [Xu/Embley 2003], and stacked generalization [Doan et al. 2004]. Machine learning is a topic of future research.

Another topic for next steps in research is a deep examination about weights and thresholds. Weight and thresholds are used for highlighting and filtering certain aspects. An examination of matching results by applying different weights and thresholds using same models and matching algorithms will reveals effects of weights and thresholds in detail. This may have an effect on the decision about the matching level of a particular BS-ES-combination.

Application

As mentioned earlier, the SCMM and resulting models (i.e. BSM and ESM) are topics of ongoing research. This leads to the fact that there is no service repository existing in real world business environments, which contains large amounts of BSM and ESM. So, in order to prove the applicability in real world the overall approach should have to be applied in a large scale scenario. This is a topic for further research.

Furthermore, the approach presented in this thesis is focused on finding an ES by a given BS, which is termed as a BS-ES matching. However, it might be interesting to apply the same approach for ES-BS, BS-BS or ES-ES matching. An ES-BS matching would reveal, which part of a software product will be used and which part of the software will not be used within a particular BS. One may conclude about wasted resources or unused potentials of software within a company. An ES-ES matching reveals differences of ES. One may conclude about the complementary and the substitution ES. Analogously, a BS-BS matching reveals differences of BS.

Last not least, this thesis is focused on functionality of BS and ES. However, there are non-functional properties as well, such as accessibility, usability, capacity, scalability, robustness, security, quality etc. Non-functional properties are used to judge about the operation of services, rather than specific behaviors. So, future work could prove the applicability of the approach for non-functional properties using different models, methods and matching algorithms.
References


Albahari, B., Drayton, P., Merrill, B., C# essentials, O'Reilly Media, 2002.


Aristoteles, Metaphysica, Athens, 350 B.C.


Christensen, E., Meredith, G., Weerawarana, S., Web Services Description Language (WSDL) 1.1, 2001.


Gilbert, D., Morgner, T., JFreeChart - Homepage, http://www.jfree.org/jfreechart/,


Grüninger, M., Fox, M.S., Methodology for the design and evaluation of ontologies, in: Proceedings 1995, S.


Haykin, S., Advances in spectrum analysis and array processing (vol. III), 1995.


Kluge, R., Preselection of Electronic Services by Given Business Services Based on Semantic Concept Correspondence, in: Proceedings 2nd International


Leymann, F., Web services flow language (WSFL 1.0), May, 2001.


Maynard, D., Term Recognition using Combined Knowledge Sources, Manchester Metropolitan University, 2000.


OASIS Open, Reference Model for Service Oriented Architecture 1.0; Committee Specification, (2006)


Platon, Politeia, Athen, 375 B.C.


Tan, P.N., Steinbach, M., Kumar, V., Chapter 8. Cluster Analysis: Basic Concepts and Algorithms, (Hrsg.), Introduction to data mining, Pearson Addison Wesley Boston, 2006a, S.

Tan, P.N., Steinbach, M., Kumar, V., Introduction to data mining, Pearson Addison Wesley Boston, 2006b.


World Wide Web Consortium (W3C), Resource Description Framework (RDF) - W3C Homepage, http://www.w3.org/RDF/,


Appendix A
Appendix B

The full XMI code can be found on the CD supplied.
d:\ServicePreselectionTool\de.hem.scmm\model\SCMM.ecore
Appendix C
Appendix D

The full XML code can be found on the CD supplied.

d:\ValidationScenario\ES_TM_OrdReceipt.scmm
- Concept BusinessPartner
  - Concept Shipper
  - Concept Consignee
  - Concept Customer
  - Concept Property hasAddress
- Concept Address
  - Data Property hasStreet
  - Data Property hasNumber
  - Data Property hasCity
  - Data Property hasZIP
  - Data Property hasCountry
- Concept Resource
  - Concept VehicleResource
    - Concept Truck
    - Concept Ship
    - Concept Aircraft
    - Concept Train
  - Concept TransportationUnit
    - Concept Container
    - Concept RailwayCar
- Concept Location
  - Concept Property hasAddress
- Concept PickupDeliveryPeriod
  - Concept Property hasRequestedDates
    - Concept RequestedDates
      - Data Property hasDateFrom
      - Data Property hasDateTo
  - Concept Property hasAcceptableDates
    - Concept AcceptableDates
      - Data Property hasEarliestDate
      - Data Property hasLatestDate
Appendix E

BS Transport Management - Order Inquiry

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transport</td>
<td>transportation, transport</td>
<td>Transport or transportation is the movement of people or goods from one location to another.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Order Management</td>
<td>order management</td>
<td>Order management comprises order entry and processing.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Order Inquiry</td>
<td>order inquiry, order receipt, receive order</td>
<td>Order inquiry comprises a set of actions that transforms a transport request into a transport order.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table E.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Customer</td>
<td>client, customer, ordering party</td>
<td>The person who sends the transportation request.</td>
<td>C</td>
<td>I 3</td>
</tr>
<tr>
<td>2</td>
<td>Order Manager</td>
<td>order manager, receiver, agent</td>
<td>The person who receives and processes the order inquiry.</td>
<td>R</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Service Provider</td>
<td>service provider, provider</td>
<td>The provider of the order inquiry service.</td>
<td>P</td>
<td>3</td>
</tr>
</tbody>
</table>

Table E.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Receive Transport Request</td>
<td>receive transport request</td>
<td>Receives a transport order and stores it for further consideration.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Confirm Transport Request</td>
<td>confirmation transport request, confirm transport request</td>
<td>A confirmation message is sent after transport order sheet has been completed and prices have been calculated.</td>
<td>3</td>
</tr>
</tbody>
</table>

Table E.3 Action

---

47 The full XML code can be found on the CD supplied. d:\ValidationScenario\BS_TrnspOrdMng.scmn
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>Involvement Type</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportRequest</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>Requested</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>TransportRequest</td>
<td>I</td>
<td>Input</td>
<td>Requested</td>
<td>Confirmed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ConfirmationMsg</td>
<td>I</td>
<td>Output</td>
<td>NoState</td>
<td>Sent</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table E.4 Resource**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportRequest</td>
<td>transport request, shipment request, transportation request</td>
<td>A transport request is a request for the transportation service provider of shipping units, freight or items.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AdressData</td>
<td>Represents address data.</td>
<td>Represents address data.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>FreightTerm</td>
<td>Freight term, inco term</td>
<td>Freight terms identify the party responsible for the payment of freight.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ServiceType</td>
<td>service type</td>
<td>The type of the transportation service, i.e. whether it is, for instance, direct way, express, truck, air or sea cargo.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ConfirmationMessage</td>
<td>confirmation message</td>
<td>A confirmation message is sent to the customer after all data is collected and confirmed by the transportation service provider.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ConsignerData</td>
<td>consigner data, consigner address data, consigner address, consigner</td>
<td>The data about the consigner of the transport order.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SourceAddress</td>
<td>source address, source location</td>
<td>The source location if different from the consigner address/location.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DestinationAddress</td>
<td>destination address, destination address, destination location</td>
<td>The address at the destination location.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>PickupTime</td>
<td>pickup time</td>
<td>Represents the pickup time.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>AdditionalData</td>
<td>additional data</td>
<td>Represents additional data to the transport order.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Good</td>
<td>good, item, freight, cargo</td>
<td>Good means the item or freight that will be transported from location A to location B.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DirectWay</td>
<td>direct way</td>
<td>Direct way means that the cargo is transported directly without any intermediate stops.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Express</td>
<td>express</td>
<td>The good will be transported in an express way.</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>TruckComplete</td>
<td>truck complete</td>
<td>A complete truck will needed to transport the cargo.</td>
<td>4</td>
</tr>
<tr>
<td>15</td>
<td>TruckMixedCargo</td>
<td>truck mixed cargo</td>
<td>A truck will transport the good. There are intermediate stopps since the truck will transport goods from other customers as well.</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>AirCargo</td>
<td>air cargo, plane</td>
<td>The cargo will be transported via air plane.</td>
<td>4</td>
</tr>
</tbody>
</table>
The cargo will be transported via sea or ship respectively.

The transportation will be fulfilled as a standard transportation using a standard package.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasConsignerData</td>
<td>consigner data, consigner address data, consigner address, consigner</td>
<td>Comprises at least three business partners, such as shipper, consignee, and the customer.</td>
<td>[0,1]</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>hasSourceAddress</td>
<td>source address</td>
<td>Represents source address data, if different from consigner data.</td>
<td>[0,1]</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>hasDestinationAddress</td>
<td>destination address, destination location</td>
<td>The address data of the destination location where the item/good/freight will be delivered.</td>
<td>[0,1]</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>hasPickupTime</td>
<td>pickup time</td>
<td>The pickup timer of the good.</td>
<td>[1,1]</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>hasAdditionalData</td>
<td>additional data</td>
<td>Represents additional data to the transport order.</td>
<td>[0,1]</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>hasGoods</td>
<td>good, item</td>
<td>Information about the freight, i.e. good or item, that will be transported.</td>
<td>[0,-1]</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>hasTransportOrder</td>
<td>transport order</td>
<td>The transport order, which contains date from transport request.</td>
<td>[1,1]</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>hasAddressData</td>
<td>address data, contact data</td>
<td>Address and contact data of the consigner.</td>
<td>[1,1]</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>hasAddressData</td>
<td>address data, source address data</td>
<td>The address data of the source location.</td>
<td>[1,1]</td>
<td>7</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>hasAddressData</td>
<td>address data, destination address data</td>
<td>The address data of the destination location.</td>
<td>[1,1]</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>hasFreightTerm</td>
<td>freight term, inco term</td>
<td>Freight terms identify the party responsible for the payment of freight.</td>
<td>[1,1]</td>
<td>10</td>
<td>3</td>
</tr>
<tr>
<td>12</td>
<td>hasServiceType</td>
<td>service type</td>
<td>Defines the type of the service.</td>
<td>[1,1]</td>
<td>10</td>
<td>4</td>
</tr>
</tbody>
</table>

Table E.5 OntologyConcept

Table E.6 ObjectProperty
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasDate-TimeFrom</td>
<td>date time from, date from, time from</td>
<td>The date and time where the good/freight/item can be picked up from.</td>
<td>[1,1]</td>
<td>9</td>
<td>DateTime</td>
</tr>
<tr>
<td>2</td>
<td>hasDate-TimeTo</td>
<td>date time to, date to, time to</td>
<td>The date and time until the good/freight/item has to be picked up.</td>
<td>[1,1]</td>
<td>9</td>
<td>DateTime</td>
</tr>
<tr>
<td>3</td>
<td>hasCargo-Insurance</td>
<td>cargo insurance</td>
<td>Indicates whether there is a cargo insurance included or not.</td>
<td>[1,1]</td>
<td>10</td>
<td>Boolean</td>
</tr>
<tr>
<td>4</td>
<td>hasCargo-Value</td>
<td>cargo value</td>
<td>The value of the cargo.</td>
<td>[1,1]</td>
<td>10</td>
<td>Decimal</td>
</tr>
<tr>
<td>5</td>
<td>hasCargo-ValueCurrency</td>
<td>cargo value currency</td>
<td>The currency of the cargo value.</td>
<td>[1,1]</td>
<td>10</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasID</td>
<td>id, identifier</td>
<td>The identifier of the good.</td>
<td>[1,1]</td>
<td>11</td>
<td>Integer</td>
</tr>
<tr>
<td>7</td>
<td>hasPackagingAmount</td>
<td>packaging amount, amount, amount of packages</td>
<td>The amount of packages or goods that should be transported.</td>
<td>[1,1]</td>
<td>11</td>
<td>Integer</td>
</tr>
<tr>
<td>8</td>
<td>hasPackagingType</td>
<td>package type, packaging type</td>
<td>Defines the type of the package, i.e. whether it is a container, a pallet, a box, etc.</td>
<td>[1,1]</td>
<td>11</td>
<td>String</td>
</tr>
<tr>
<td>9</td>
<td>hasWeight</td>
<td>weight, kg, kilogramm, good weight</td>
<td>The weight of the good in kilogram.</td>
<td>[1,1]</td>
<td>11</td>
<td>String</td>
</tr>
<tr>
<td>10</td>
<td>hasSize</td>
<td>size, volume</td>
<td>The height, length, width or the volume.</td>
<td>[1,1]</td>
<td>11</td>
<td>String</td>
</tr>
<tr>
<td>11</td>
<td>hasDescription</td>
<td>description, note</td>
<td>The description of the good.</td>
<td>[1,1]</td>
<td>11</td>
<td>String</td>
</tr>
<tr>
<td>12</td>
<td>hasCompanyName</td>
<td>company name, company</td>
<td>The name of the company.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>13</td>
<td>hasContactPerson</td>
<td>contact person, person name</td>
<td>The contact person at the location.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>14</td>
<td>hasStreet-Number</td>
<td>street, number, house number, street number</td>
<td>The street and the house number of the address.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>15</td>
<td>hasZIP</td>
<td>zip</td>
<td>The zip code at the address.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>16</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city of the address.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>17</td>
<td>hasCountry</td>
<td>country,</td>
<td>The country name of</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>DataProperty</td>
<td>Description</td>
<td>Min</td>
<td>Max</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>hasPhone</td>
<td>phone, telephone</td>
<td>1</td>
<td>1</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>hasFax</td>
<td>fax</td>
<td>0</td>
<td>1</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>hasEmail</td>
<td>email</td>
<td>1</td>
<td>1</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

*Table E.7 DataProperty*
Appendix F

ES CRM – CustomerEntry

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CustomerRelationshipManagement</td>
<td>customer relationship management</td>
<td>Customer relationship management (CRM) is a widely-implemented strategy for managing a company’s interactions with customers, clients and sales prospects.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>CustomerEntry</td>
<td>customer entry, customer creation</td>
<td>Customer entry is a service that enters customer data into the system, i.e. stores customer data.</td>
<td>C</td>
<td>1</td>
</tr>
</tbody>
</table>

Table F.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operator</td>
<td>operator, customer care manager</td>
<td>The person who operates the service.</td>
<td>R</td>
<td>2</td>
</tr>
</tbody>
</table>

Table F.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EnterCustomerData</td>
<td>This action comprises receiving customer data and enter it into the system.</td>
<td>2</td>
</tr>
</tbody>
</table>

Table F.3 Action

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CustomerData</td>
<td>I</td>
<td>Created</td>
<td>Empty</td>
<td>Filled</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table F.4 Resource

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Customer-Data</td>
<td>customer data, client data</td>
<td>Customer comprises data about a certain customer.</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>ContactDetails</td>
<td>concat details, contact</td>
<td>The contact details of the customer.</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Address</td>
<td>address, location</td>
<td>The address deteal of a customer.</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Address</td>
<td>address type, type</td>
<td>The address type specifies the type of the ad-</td>
<td>-</td>
</tr>
</tbody>
</table>

48 The full XML code can be found on the CD supplied. 
\d:\ValidationScenario\ES_CRM_CstmrEntr.scmm
Type of address

<table>
<thead>
<tr>
<th>ID</th>
<th>Type</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Home</td>
<td>home address, home</td>
<td>Home means the home address.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Work</td>
<td>work address, work</td>
<td>Work means work address.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Additional</td>
<td>additional address, additional</td>
<td>Additional means an additional address.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table F.5 OntologyConcept

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasContactDetails</td>
<td>contact details, contact data</td>
<td>Comprises at least three business partners, such as shipper, consignee, and the customer.</td>
<td>[1,1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>hasAddress</td>
<td>address</td>
<td>This property contains address details about the contact.</td>
<td>[1,-1]</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>hasAddressType</td>
<td>address type</td>
<td>The address type specifies the type of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Table F.6 ObjectProperty

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasPhoneNumber</td>
<td>phone number, telephone number</td>
<td>The contact phone number.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>2</td>
<td>hasMobileNumber</td>
<td>mobile number, mobile telephone number, mobile phone number</td>
<td>The mobile number of the contact.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasFaxNumber</td>
<td>fax, fax number</td>
<td>The fax number of the contact.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasEmail</td>
<td>email, email address</td>
<td>The email address of the contact.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasStreet</td>
<td>The street of the address.</td>
<td>The street of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasNumber</td>
<td>street number, house number, number, number</td>
<td>The house number of the address location.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>7</td>
<td>hasCity</td>
<td>city, town</td>
<td>city of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>hasZIP</td>
<td>zip</td>
<td>zip code of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>9</td>
<td>hasCountry</td>
<td>country</td>
<td>country of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
</tbody>
</table>

Table F.7 DataProperty
Appendix G

ES ERP – SendInvoice*

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Enterpr -</td>
<td></td>
<td></td>
<td>Enterprise resource planning (ERP) manages the use of resources, such as materials, money and humans/employees, within business processes of a company.</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Invoic-</td>
<td></td>
<td></td>
<td>Invoicing is concerned with all invoice relevant activities within a company.</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>SendIn-</td>
<td></td>
<td></td>
<td>This service send an invoice/bill to the customer.</td>
<td>C</td>
</tr>
</tbody>
</table>

**Table G.1 Service**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Account-</td>
<td></td>
<td></td>
<td>The person who manages all accounting relevant concerns, i.e. sending invoices.</td>
<td>R</td>
</tr>
<tr>
<td>2</td>
<td>Invoice-</td>
<td></td>
<td></td>
<td>The invoice is sent to the invoice recipient, i.e. the customer.</td>
<td>N</td>
</tr>
</tbody>
</table>

**Table G.2 Stakeholder**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateIn-</td>
<td></td>
<td></td>
<td>The invoice is created based on an order which has already been processed.</td>
</tr>
<tr>
<td>2</td>
<td>CompleteIn-</td>
<td></td>
<td></td>
<td>The invoice is processed and completed within this action, i.e. all invoice relevant parameters are filled in the invoice document.</td>
</tr>
<tr>
<td>3</td>
<td>SendInvoice</td>
<td>send invoice, submit invoice, mail invoice</td>
<td>The invoice is send to the recipient/customer/payer via mail or email.</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table G.3 Action**

*The full XML code can be found on the CD supplied.

d:\ValidationScenario\ES_ERP_SndInvc.scmn
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Invoice</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Invoice</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Completed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Invoice</td>
<td>I</td>
<td>Changed</td>
<td>Completed</td>
<td>Sent</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

### Table G.4 Resource

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Invoice</td>
<td>invoice, bill, invoice document</td>
<td>An invoice or bill is a commercial document issued by a seller to the buyer, indicating the products, quantities, and agreed prices for products or services the seller has provided the buyer.</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Supplier</td>
<td>supplier, distributor, provider, vendor, manufacturer</td>
<td>The supplier is the party who distributes the product.</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Customer</td>
<td>client, customer, addressee, payer</td>
<td>The customer, i.e. the person who is the addressee of the invoice.</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Payment-Method</td>
<td>payment method, method of payment</td>
<td>The method of payment.</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Product</td>
<td>product, item, good</td>
<td>Represents one particular product</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>CashOn-Delivery</td>
<td>cod, cash on delivery</td>
<td>Pay when the product is delivered.</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>Check</td>
<td>check</td>
<td>Pay via check.</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>CreditCard</td>
<td>credit card</td>
<td>Pay via credit card.</td>
<td>4</td>
</tr>
<tr>
<td>9</td>
<td>DirectBank Transfer</td>
<td>direct bank transfer, wire bank transfer</td>
<td>Pay via wire bank transfer.</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table G.5 OntologyConcept

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card. ID</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>has SUP</td>
<td>supplier, distributor, provider, vendor, manufacturer</td>
<td>The supplier is the party who distributes the product.</td>
<td>[1,1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>has Customer</td>
<td>client, customer, addressee, payer</td>
<td>The customer, i.e. the person who is the addressee of the invoice.</td>
<td>[1,1]</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>has PossiblePaymentMethods</td>
<td>payment method, possible payment method</td>
<td>The payment method that are available in order to pay the delivered product.</td>
<td>[1,1]</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>has Product</td>
<td>product, item,</td>
<td>Products that are delivered.</td>
<td>[1,1]</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>
good verified to the customer.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasSupplierBusinessName</td>
<td>supplier business name, supplier name, company name, business name</td>
<td>The business name of the supplier.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>2</td>
<td>hasContactName</td>
<td>contact name, person name</td>
<td>The name of the contact person on the supplier side.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasBusinessTelephone</td>
<td>telephone, business telephone</td>
<td>The telephone number of the contact person on the supplier side.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasBusinessFax</td>
<td>fax, business fax</td>
<td>The fax number of the contact person on the supplier side.</td>
<td>[0,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasBusinessEmail</td>
<td>email, business email</td>
<td>The business email of the contact person on the supplier side.</td>
<td>[0,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasBusinessAddress</td>
<td>business address, supplier address</td>
<td>The business address of the supplier.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>7</td>
<td>hasTaxID</td>
<td>tax identification number, tax id</td>
<td>The tax id of the supplier.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>hasBankName</td>
<td>bank name, suppliers bank name</td>
<td>The suppliers bank name.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>9</td>
<td>hasBankCode</td>
<td>bank code, suppliers bank code</td>
<td>The bank code of the suppliers bank.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>10</td>
<td>hasAccountName</td>
<td>account name, bank account name, suppliers bank account name</td>
<td>The suppliers bank account name.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>11</td>
<td>hasAccountNumber</td>
<td>bank account number, account number, suppliers bank account number</td>
<td>The suppliers bank account number.</td>
<td>[1,1]</td>
<td>2</td>
<td>String</td>
</tr>
<tr>
<td>12</td>
<td>hasCustomer</td>
<td>customer</td>
<td>The business name of the customer.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>Column Name</td>
<td>Description</td>
<td>Type</td>
<td>Constraints</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>-------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>merBusinessName</td>
<td>business name, customer name</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasContactName</td>
<td>contact name, person name</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasBusinessTelephone</td>
<td>telephone, business telephone</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasBusinessFax</td>
<td>fax, business fax</td>
<td>[0,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasBusinessEmail</td>
<td>email, email address</td>
<td>[0,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasBusinessAddress</td>
<td>business address, address</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasDeliveryAddress</td>
<td>delivery address</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasProductCode</td>
<td>product code</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasDescription</td>
<td>description, product description</td>
<td>[0,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasQuantity</td>
<td>quantity, amount</td>
<td>[1,1]</td>
<td>Integer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasPricePerUnit</td>
<td>price, price per unit</td>
<td>[1,1]</td>
<td>Decimal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasUnit</td>
<td>unit</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasTax</td>
<td>tax, tax percentage</td>
<td>[1,1]</td>
<td>Decimal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasDiscount</td>
<td>discount, discount percentage</td>
<td>[0,1]</td>
<td>Decimal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasNote</td>
<td>note, comment</td>
<td>[0,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasInvoiceDate</td>
<td>invoice date</td>
<td>[1,1]</td>
<td>DateTime</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasCurrency</td>
<td>currency</td>
<td>[1,1]</td>
<td>String</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hasPostagePrice</td>
<td>shipping costs, postage price, postage cost, mail</td>
<td>[1,1]</td>
<td>Decimal</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>DataProperty</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>hasPostageTax, shipping tax, postage tax</td>
<td>hasPostageTax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The tax which is included in the postage.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>hasTotalPrice, total price, sum price</td>
<td>hasTotalPrice</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The total price of the invoice.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>hasTotalTax, total tax, sum tax</td>
<td>hasTotalTax</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>The total tax of the invoice.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>hasDescription, description</td>
<td>hasDescription</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>An optional description of the invoice.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>hasDeliveryInstructions, comments, delivery instructions</td>
<td>hasDeliveryInstructions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Some optional comments for delivery.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table G.7 DataProperty*
Appendix H

ES SAP TM – InvoiceManagement

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportManagement</td>
<td>transport management, transportation management</td>
<td>Transport management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>InvoiceManagement</td>
<td>invoice management</td>
<td>This service provides capabilities around invoicing, i.e. creation, sending and management of invoices or bills.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Send-CustomerInvoice</td>
<td>send customer invoice, sending customer invoice, send customer bill</td>
<td>This service is concerned with sending an invoice to the customer.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table H.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AccountingManager</td>
<td>accounting manager, invoice manager, operator</td>
<td>The accounting manager is responsible for creation and sending of invoices.</td>
<td>Ctrl, R</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Customer</td>
<td>client, customer, payer</td>
<td>The client receives the invoice and has to pay the bill.</td>
<td>N</td>
<td>3</td>
</tr>
</tbody>
</table>

Table H.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateCustomerInvoice</td>
<td>create customer invoice, create customer bill, create invoice, create bill</td>
<td>This action is concerned with the creation of the customer invoice.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>ProcessCustomerInvoice</td>
<td>process customer invoice, process customer bill, process invoice, process bill</td>
<td>This action is concerned with the processing of the customer invoice.</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CompleteCustomerInvoice</td>
<td>complete customer invoice, complete customer invoice, complete invoice</td>
<td>This action is concerned with the completion of the customer invoice.</td>
<td>3</td>
</tr>
</tbody>
</table>

The full XML code can be found on the CD supplied.
d:\ValidationScenario\ES_TM_SendCstInvc.scmm
voice complete customer bill, complete invoice, complete bill

|   | SendCustomerInvoice | send customer invoice, send customer bill, send invoice, send bill | This action is concerned with the sending of the invoice to the customer. | 5 |

**Table H.3 Action**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CustomerInvoice</td>
<td>I</td>
<td>Created</td>
<td>New</td>
<td>NoState</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>CustomerInvoice</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Modified</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>CustomerInvoice</td>
<td>I</td>
<td>Changed</td>
<td>Modified</td>
<td>Completed</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>CustomerInvoice</td>
<td>I</td>
<td>Changed</td>
<td>Completed</td>
<td>Completed</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table H.4 Resource**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CustomerInvoice</td>
<td>customer invoice, customer bill, client bill, client invoice</td>
<td>A customer invoice is an invoice that is sent to the customer (i.e. the client or ordering party).</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Customer</td>
<td>customer, consigner, addresser</td>
<td>The party or business partner that represents the customer, addressee or consigner of the good, item or freight.</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Address</td>
<td>address</td>
<td>Represents all data items of an address.</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Location</td>
<td>location, address</td>
<td>A location is a place somewhere in the universe which is defined by an address.</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>ItemData</td>
<td>item data, goods data, freight data</td>
<td>Comprises detailed definitions of the goods being transported. It comprises goods type, goods description, product numbers, marks/numbers, and so on. In additional subnotes you can enter any measurement value (weights, volumes, numbers, dimensions), value indications, dangerous goods indication, customs details, and packaging details. All fields, with the exception of the goods description and quantity indication, are optional.</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>GrossWeight</td>
<td>gross weight, item weight, good weight, weight</td>
<td>The weight of the item or good that has to be transported.</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>GrossSize</td>
<td>gross size, item size, good size, size, volume</td>
<td>This property includes the size of the item or good that will be transported. The gross size is specified by the volume, i.e. breadth, length, and height, of the item.</td>
<td>-</td>
</tr>
</tbody>
</table>

**Table H.5 OntologyConcept**
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItem-Data</td>
<td>item data, good data, freight data</td>
<td>Comprises detailed definitions of the goods being transported.</td>
<td>[1,1]</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>hasCustomer</td>
<td>customer, client, payer</td>
<td>The name and address of the customer.</td>
<td>[1,1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>hasAddress</td>
<td>address</td>
<td>A customer does always has an address.</td>
<td>[1,1]</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>hasGross-Weight</td>
<td>gross weight, good weight, item weight, weight</td>
<td>The weight of the item or good that has to be transported.</td>
<td>[0,1]</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>5</td>
<td>hasGross-Size</td>
<td>gross size, item size, good size, size</td>
<td>This property includes the size of the item or good that will be transported.</td>
<td>[0,1]</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>6</td>
<td>hasSourceLocation</td>
<td>source location, customer location, consignor location, addressee location</td>
<td>This is the location from where the good or item will be shipped.</td>
<td>[1,1]</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>hasDestinationLocation</td>
<td>destination location, consignee location, addressee location</td>
<td>This is the location where the good or item will be shipped to.</td>
<td>[1,1]</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>hasAddress</td>
<td>address, location</td>
<td>A location has an address.</td>
<td>[1,1]</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table H.6 ObjectProperty**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItem-Number</td>
<td>item number, good number, number, identifier, item identifier, good identifier</td>
<td>The number or the identifier of the item that will be transported.</td>
<td>[1,1]</td>
<td>5</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>hasItem-Description</td>
<td>description, item description</td>
<td>Some text that describes the item or good.</td>
<td>[0,1]</td>
<td>5</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasPick-upDate-Time</td>
<td>pickup date time</td>
<td>Date and time when the item was picked up.</td>
<td>[1,1]</td>
<td>5</td>
<td>DateTime</td>
</tr>
<tr>
<td>4</td>
<td>hasDeliveryDate-Time</td>
<td>delivery date time</td>
<td>Date and time when the item was delivered.</td>
<td>[1,1]</td>
<td>5</td>
<td>DateTime</td>
</tr>
<tr>
<td>5</td>
<td>hasInvoiceDate</td>
<td>invoice date</td>
<td>The date when the invoice is created.</td>
<td>[1,1]</td>
<td>1</td>
<td>DateTime</td>
</tr>
<tr>
<td>6</td>
<td>hasTotalDistance</td>
<td>total distance</td>
<td>The total distance of</td>
<td>[0,1]</td>
<td>1</td>
<td>Decimal</td>
</tr>
<tr>
<td></td>
<td>Property</td>
<td>Description</td>
<td>Min</td>
<td>Max</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----</td>
<td>-----</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>hasTotalAmount</td>
<td>total amount, total quantity</td>
<td>[0,1]</td>
<td>1</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>hasPricePerItem</td>
<td>price per item</td>
<td>[0,1]</td>
<td>1</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>hasTotalPrice</td>
<td>total price of the invoice</td>
<td>[1,1]</td>
<td>1</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>hasStreet</td>
<td>street</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>hasNumber</td>
<td>street number, house number, number</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>hasCity</td>
<td>city, town</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>hasZIP</td>
<td>zip, zip code, area code, zip area code</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>hasCountry</td>
<td>country, country code</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>hasWeightValue</td>
<td>weight, weight value</td>
<td>[1,1]</td>
<td>6</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>hasWeightUnit</td>
<td>weight unit</td>
<td>[1,1]</td>
<td>6</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>hasBreadth</td>
<td>breadth, good breadth, item breadth</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>hasLength</td>
<td>length, good length, item length</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>hasHeight</td>
<td>height, good height, item height</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>hasLengthUnit</td>
<td>length unit</td>
<td>[1,1]</td>
<td>7</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

Table H.7 DataProperty
Appendix I

ES SAP TM – PlanTransportationActivities51

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportManagement</td>
<td>transport management,</td>
<td>Transport management comprises management and coordination of moving good</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>transportation management</td>
<td>from one location to another one.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Planning</td>
<td>planning, transport planning</td>
<td>Planning creates a transportation plan that executes transportation</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>requirements with available transportation capacities.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Plan-TransportationActivities</td>
<td>plan transportation activities</td>
<td>This action takes transportation requirements and transportation capacities and makes a transportation plan out of it.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table I.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Planner</td>
<td>transportation service prov.,</td>
<td>The planner plans the transportation.</td>
<td>P, R, I</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>order mngr</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table I.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PlanTransportationActivities</td>
<td>plan transportation activities</td>
<td>This action takes transportation requirements and transportation capacities and makes a transportation plan out of it.</td>
<td>5</td>
</tr>
</tbody>
</table>

Table I.3 Action

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transportation-Plan</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>FreightUnit</td>
<td>M</td>
<td>Input</td>
<td>NoState</td>
<td>NoState</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>VehicleResource</td>
<td>M</td>
<td>Input</td>
<td>NoState</td>
<td>NoState</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>TransportationUnit</td>
<td>M</td>
<td>Input</td>
<td>NoState</td>
<td>NoState</td>
<td>1</td>
<td>12</td>
</tr>
</tbody>
</table>

Table I.4 Resource

51 The full XML code can be found on the CD supplied.
d:\ValidationScenario\ES_TM_PlnTrnsAct.scmm
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FreightUnit</td>
<td>freight unit</td>
<td>A freight unit is a combination of good that are transported together.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Resource</td>
<td>resource</td>
<td>Resources are divided into transportation units and vehicle resources. You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TransportationPlan</td>
<td>transportation plan</td>
<td>A transportation plan comprises one to many transportation activities.</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>TransportationActivity</td>
<td>transportation activity</td>
<td>Transportation activity creates the connection between transportation requirements and capacities.</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Location</td>
<td>location, address</td>
<td>A location is a place somewhere in the universe which is defined by an address.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Address</td>
<td>address</td>
<td>Represents all data items of an address.</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>VehicleResource</td>
<td>vehicle resource, vehicle, active vehicle, means of transport, means of transport</td>
<td>Vehicle resources are moving resources, including trucks, planes and ships, which can transport goods between location. A vehicle resource is always an active vehicle, which can be driven independently.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Truck</td>
<td>truck, lorry, road train</td>
<td>A truck or lorry is a motor vehicle designed to transport cargo.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Ship</td>
<td>ship, vessel</td>
<td>A ship transports goods via sea, rivers or lakes.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Aircraft</td>
<td>aircraft, plane</td>
<td>An aircraft is an aircraft capable of flight using forward motion that generates lift as the wing moves through the air.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Train</td>
<td>train, railway</td>
<td>A train is a connected series of vehicles for rail transport that move along a track (permanent way) to transport cargo or passengers from one place to another.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>TransportationUnit</td>
<td>transportation unit</td>
<td>The unit with which the good or item will be transported.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Container</td>
<td>container, compartment</td>
<td>A container that can be transported via ship, train or truck.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>RailwayCar</td>
<td>railway car</td>
<td>A railroad car is a vehicle on a rail transport system (railroad or railway) that is used for the carrying of cargo or passengers.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TransportationActivityType</td>
<td>transportation activity type</td>
<td>This concept defines the type of the transportation activity.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Loading</td>
<td>loading</td>
<td>This type refers to a freight unit and a transportation capacity and indicates that the freight unit is loaded into the transportation capacity at the</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Name</td>
<td>Labels</td>
<td>Description</td>
<td>Card. ID</td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td>----------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>17</td>
<td>Unloading</td>
<td>unloading</td>
<td>This type refers to a freight unit and a transportation capacity and indicates that the freight unit is unloaded from the transportation capacity at the specific location.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Transportation</td>
<td>transportation</td>
<td>This indicates that the activity moves a set of freight units and a transportation capacity from A to B.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Coupling</td>
<td>coupling</td>
<td>This type refers to an active and a passive vehicle resource and indices that both resources are coupled from each other and subsequently can move together.</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Uncoupling</td>
<td>uncoupling</td>
<td>This type refers to an active and a passive vehicle resource and indices that both resources are uncoupled from each other and subsequently cannot move together.</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>AssignTransportationUnit</td>
<td>assign transportation unit</td>
<td>This type relates to a vehicle resource (active or passive) and a transportation unit and indicates that the assignment of transportation unit to the vehicle has been done.</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>UnassignTransportationUnit</td>
<td>unassign transportation unit</td>
<td>This type relates to a vehicle resource (active or passive) and a transportation unit and indicates that the assignment of transportation unit to the vehicle has been undone.</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>AssignDriver</td>
<td>assign driver</td>
<td>This type relates to an active vehicle resource and a driver and indicates that the driver is assigned to a vehicle.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>UnassignDriver</td>
<td>unassign driver</td>
<td>This type relates to an active vehicle resource and a driver and indicates that the assignment of the driver to the vehicle has been undone.</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>ActivityState</td>
<td>activity state, transportation activity state</td>
<td>A planned transportation activity has a status that defines how to handle further planning with the activity.</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>PlanningFixed</td>
<td>planning fixed</td>
<td>The activity must not be deleted. The assignment of drivers and other resource must not be changed.</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Released</td>
<td>released, planning released</td>
<td>The allocation of freight units and defined times must not be changed anymore.</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Blocked</td>
<td>blocked, planning blocked</td>
<td>Activity is locked for executing the transportation.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1.5 OntologyConcept

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card. ID</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasTransportationActivities</td>
<td>transportation activities</td>
<td>A transportation plan has three to many transportation activities. The three mandatory activities are loading, transport and unloading.</td>
<td>[3,-1]</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>ID</td>
<td>hasActivityType</td>
<td>activity type, transportation activity type</td>
<td>This attributes defines the type of the transportation activity.</td>
<td>[1,1]</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>----</td>
<td>----------------</td>
<td>---------------------------------------------</td>
<td>---------------------------------------------------------------</td>
<td>------</td>
<td>---</td>
<td>----</td>
</tr>
<tr>
<td>3</td>
<td>hasActivityState</td>
<td>activity state, transportation activity state</td>
<td>A planned transportation activity has a status that defines how to handle further planning with the activity.</td>
<td>[0,1]</td>
<td>4</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>hasTransportationCapacity</td>
<td>resource, transportation capacity, transportation resource, transportation unit</td>
<td>Each activity has at least one transportation capacity, i.e. a transportation resource or unit.</td>
<td>[1,1]</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>hasFreightUnit</td>
<td>freight unit</td>
<td>An activity may refer to a freight unit.</td>
<td>[0,1]</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>hasLocation</td>
<td>location</td>
<td>The location of the activity.</td>
<td>[0,1]</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>hasAddress</td>
<td>address, location</td>
<td>a location has an address</td>
<td>[1,1]</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

Table I.6 ObjectProperty

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasStartDateTime</td>
<td>start time, start date, start date time</td>
<td>The start of the activity.</td>
<td>[0,1]</td>
<td>4</td>
<td>DateTime</td>
</tr>
<tr>
<td>2</td>
<td>hasEndDateTime</td>
<td>end time, end date, end date time</td>
<td>The end of the activity.</td>
<td>[0,1]</td>
<td>4</td>
<td>DateTime</td>
</tr>
<tr>
<td>3</td>
<td>hasStreet</td>
<td>street</td>
<td>The address street</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasNumberOf</td>
<td>street number, house number, number</td>
<td>The number of the house in the street.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasZIP</td>
<td>zip, zip code, area code, zip area code</td>
<td>The zip code of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>7</td>
<td>hasCountry</td>
<td>country, country code</td>
<td>The country of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
</tbody>
</table>

Table I.7 DataProperty
Appendix J

ES SAP TM – OrderReceipt Order Management

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportManagement</td>
<td>transport management, transportation management</td>
<td>Transport management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Order-Management</td>
<td>order management</td>
<td>Order management and order receipt are generally the beginning of an operational process in transportation management.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Order-Receipt</td>
<td>order receipt, order inquiry, receive order, receive request</td>
<td>Order receipt comprises a series of key operations that can be performed in acceptance a shipment request.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportationServiceProvider</td>
<td>transportation service provider, order manager</td>
<td>The transportation service provider manages the process of order from order entry to order confirmation.</td>
<td>P, Ctrl, R</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>OrderingParty</td>
<td>client, ordering party</td>
<td>The ordering party sends a request and gets order confirmation.</td>
<td>C, I</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateShipmentRequest</td>
<td>create shipment request, enter shipment request, shipment request entering, create shipment request, enter transport request, transport request entering</td>
<td>A shipment request is created either by directly creating it or by copying a reference shipment request.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>ProcessShipmentRequest</td>
<td>shipment request processing, shipment request completion, transport request</td>
<td>If the shipment request has not been completely and consistently created during the first entry, it can be completed in a second or further processing step.</td>
<td>3</td>
</tr>
</tbody>
</table>

52 The full XML code can be found on the CD supplied. d:\ValidationScenario\ES_TM_OrdReceipt.smm
processing, transport request completion

3 Activate-ShipmentRequest
activate shipment request, complete shipment request, activate transport request, complete transport request
When the shipment request is full described it can be activated.

4 Confirm-ShipmentRequest
confirm shipment request, approve shipment request, confirm transport request, approve transport request
After shipment request activation it is confirmed to the customer.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Changed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>Activated</td>
<td>Activated</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Activated</td>
<td>Confirmed</td>
<td>Confirmed</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>shipment request, transport request, freight request, transport</td>
<td>A shipment request is an agreement between a transportation service provider and a ordering party regarding the transportation of goods or transportation equipment from an issuing partner or location to a receiving partner or location according to the agreed conditions.</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>BusinessPartner</td>
<td>business partner</td>
<td>The subnotes for the business partners, aside from the mandatory specifications Shipper, Consignee and Customer, allow further parties involved in the transportation operation to be defined. These may be, for example, predetermined service providers (e.g. if requested by the customer), agents, or invoice recipients.</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Address</td>
<td>address</td>
<td>Represents all data items of an address.</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Resource</td>
<td>resource</td>
<td>Resources are divided into transportation units and vehicle resources. You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Location</td>
<td>location, address</td>
<td>A location is a place somewhere in the universe</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>PickupDeliveryPeriod</td>
<td>pickup delivery period</td>
<td>Pickup and delivery period defines the periods for pickup and delivery of freight, goods or items.</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ItemData</td>
<td>item data, goods data, freight data</td>
<td>Comprises detailed definitions of the goods being transported. It comprises goods type, goods description, product numbers, marks/numbers, and so on. In additional subnotes you can enter any measurement value (weights, volumes, numbers, dimensions), value indications, dangerous goods indication, customs details, and packaging details. All fields, with the exception of the goods description and quantity indication, are optional.</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>GrossWeight</td>
<td>gross weight, good weight, weight</td>
<td>The weight of the item or good that has to be transported.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>GrossSize</td>
<td>gross size, item size, good size, size, volume</td>
<td>This property includes the size of the item or good that will be transported. The gross size is specified by the volume, i.e. breadth, length, and height, of the item.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Shipper</td>
<td>shipper, transportation service provider, transporting party, carrier, forwarder</td>
<td>The party that ships the good. Shipper is also known as a carrier or freight forwarder.</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Consignee</td>
<td>consignee, addressee</td>
<td>The party or business partner that is the consignee or addressee of the good, item or freight.</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Customer</td>
<td>customer, consignor, addresser</td>
<td>The party or business partner that is the consignor, addresser or consignee of the good, item or freight.</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>VehicleResource</td>
<td>vehicel resource, vehicle, active vehicle, active means of transport, means of transport</td>
<td>Vehicle resources are moving resources, including trucks, planes and ships, which can transport goods between location. A vehicle resource is always an active vehicle, which can be driven independently.</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>TransportationUnit</td>
<td>transportation unit</td>
<td>The unit with which the good or item will be transported.</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Truck</td>
<td>truck, lorry, road train</td>
<td>A truck or lorry is a motor vehicle designed to transport cargo.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Ship</td>
<td>ship, vessel</td>
<td>A ship transports goods via sea, rivers or lakes.</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Aircraft</td>
<td>aircraft, plane</td>
<td>An aircraft is capable to flight using forward motion that generates lift as the wing moves through the air.</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Train</td>
<td>train, railway</td>
<td>A train is a connected series of vehicles for rail transport that move along a track (permanent way) to transport cargo or passengers from one place to another.</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Container</td>
<td>container, com-</td>
<td>A container that can be transported via ship, train</td>
<td></td>
</tr>
</tbody>
</table>
partment or truck.

20 Railway-Car railway car A railroad car is a vehicle on a rail transport system (railroad or railway) that is used for the carrying of cargo or passengers. 14

21 RequestedDates requested dates Requested dates define a time period from when to when a freight or cargo can be picked up. -

22 AcceptableDates acceptable dates Acceptable dates define a time period from when (min) to when (max) a freight or cargo can be picked up. -

23 Note note Within a shipment request you can enter any number of language dependent notes. Notes are categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print and communicate. -

24 Dossier dossier A dossier allows you to enter document references, information on required documents, and file attachments (e.g. scanned documents for shipment request) and make them accessible to processors. -

<table>
<thead>
<tr>
<th>Table 1.5 OntologyConcept</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>11</td>
</tr>
<tr>
<td>12</td>
</tr>
<tr>
<td>13</td>
</tr>
<tr>
<td>14</td>
</tr>
<tr>
<td>15</td>
</tr>
</tbody>
</table>

Table J.6 ObjectProperty
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItem-Number</td>
<td>item number, good number, number, identifier, good identifier</td>
<td>The number or the identifier of the item that will be transported.</td>
<td>[1,1]</td>
<td>7</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>hasItem-Description</td>
<td>description, item description</td>
<td>Some text that describes the item or good.</td>
<td>[0,1]</td>
<td>7</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasStreet</td>
<td>street</td>
<td>The name of the street of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasNumber</td>
<td>street number, house number, number</td>
<td>The number of the house in the street.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasZIP</td>
<td>zip, zip code, area code, zip area code</td>
<td>The zip code of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>7</td>
<td>hasCountry</td>
<td>country, country code</td>
<td>The country of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>hasDate-From</td>
<td>date from</td>
<td>The date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>21</td>
<td>DateTime</td>
</tr>
<tr>
<td>9</td>
<td>hasDateTo</td>
<td>date to</td>
<td>The date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>21</td>
<td>DateTime</td>
</tr>
<tr>
<td>10</td>
<td>hasEarliestDate</td>
<td>earliest date, earliest date from</td>
<td>The earliest date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>22</td>
<td>DateTime</td>
</tr>
<tr>
<td>11</td>
<td>hasLatestDate</td>
<td>latest date, latest date to</td>
<td>The latest date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>22</td>
<td>DateTime</td>
</tr>
<tr>
<td>12</td>
<td>hasWeightValue</td>
<td>weight, weight value</td>
<td>Comprises the value of the good weight or item weight respectively.</td>
<td>[1,1]</td>
<td>8</td>
<td>Decimal</td>
</tr>
<tr>
<td>13</td>
<td>hasWeightUnit</td>
<td>weight unit</td>
<td>Comprises the weight unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>8</td>
<td>String</td>
</tr>
<tr>
<td>14</td>
<td>hasBreadth</td>
<td>breadth, good breadth, item breadth</td>
<td>The breadth of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
</tr>
<tr>
<td>15</td>
<td>hasLength</td>
<td>length, good length, item length</td>
<td>The length of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
</tr>
<tr>
<td></td>
<td>hasHeight</td>
<td>height, good height, item height</td>
<td>The height of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
</tr>
<tr>
<td>---</td>
<td>-----------</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
<td>-------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>hasLengthUnit</td>
<td>length unit</td>
<td>Comprises the length unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>9</td>
<td>String</td>
</tr>
</tbody>
</table>

*Table J.7 DataProperty*
Appendix K

ES SAP TM – Order Management renamed

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transportation Management</td>
<td>transportation management</td>
<td>Transportation management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Order</td>
<td>order management, order</td>
<td>Order is the beginning of an operational process in transportation management.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Receive Request</td>
<td>order receipt, order inquiry, receive order, receive request</td>
<td>Receive request comprises a series of operations that can be performed in acceptance a transportation request.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table K.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Order-Manager</td>
<td>order manager, transportation provider</td>
<td>The order manager manages the process of order from order entry to order confirmation.</td>
<td>P, Ctrl, R</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Customer</td>
<td>client, orderer, customer, ordering party</td>
<td>The customer sends a request and gets an order confirmation.</td>
<td>C, I</td>
<td>3</td>
</tr>
</tbody>
</table>

Table K.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>EnterTransportationRequest</td>
<td>enter transport request, create shipment request, enter transportation request, enter shipment request, create transport</td>
<td>A transport request is entered either by directly creating it or by copying a reference business object.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>ProcessTransportationRequest</td>
<td>process transportation request, process shipment request</td>
<td>If the transportation request has not been completely and consistently created during the first entry, it can be completed in a second or further processing step.</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>CompleteTransportationRequest</td>
<td>complete transportation request</td>
<td>When the request is fully described and no further processing is necessary, it is completed.</td>
<td>3</td>
</tr>
</tbody>
</table>

The full XML code can be found on the CD supplied.

\d:\ValidationScenario\ES_TM_OrdReceipt_a_renamed.scmn
When transportation request has been completed so far and planned or priced if necessary, so that a confirmation can be sent to the ordering party, the order confirmation step is performed.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>Involvement/Type</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportationRequest</td>
<td>I</td>
<td>Created</td>
<td>No</td>
<td>Created</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>TransportationRequest</td>
<td>I</td>
<td>Changed</td>
<td>Created</td>
<td>Modified</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>TransportationRequest</td>
<td>I</td>
<td>Changed</td>
<td>Modified</td>
<td>FullDescribed</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>TransportationRequest</td>
<td>I</td>
<td>Changed</td>
<td>FullDescribed</td>
<td>Approved</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table K.3 Action**

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportationRequest</td>
<td>shipment request, transport request, freight request, transportation request</td>
<td>A transportation request is an agreement between a transportation service provider and an ordering party regarding the transportation of goods or transportation equipment from an issuing partner or location to a receiving partner or location according to the agreed conditions.</td>
</tr>
<tr>
<td>2</td>
<td>Business-Participant</td>
<td>business participant, business partner</td>
<td>The subnotes for the business participant, aside from the mandatory specifications Shipper, Consignee and Customer, allow further parties involved in the transportation operation to be defined. These may be, for example, pre-determined service providers (e.g. if requested by the customer), agents, or invoice recipients.</td>
</tr>
<tr>
<td>3</td>
<td>Address</td>
<td>address</td>
<td>Represents all data items of an address.</td>
</tr>
<tr>
<td>4</td>
<td>TransportationResource</td>
<td>transportation resource</td>
<td>Transportation resources are divided into transportation units and vehicle resources. You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in pack-aged form in a container) or you can make a note of special resource requirements from the customer here.</td>
</tr>
<tr>
<td>5</td>
<td>Location</td>
<td>location, address</td>
<td>A location is a place somewhere in the universe which is defined by an address</td>
</tr>
<tr>
<td>6</td>
<td>PickupDeliveryTime</td>
<td>pickup delivery period, pickup delivery time</td>
<td>Pickup and delivery time defines the periods for pickup and delivery of freight, goods or items.</td>
</tr>
<tr>
<td>7</td>
<td>FreightData</td>
<td>item data, goods data, freight data</td>
<td>Comprises detailed definitions of the freight being transported. It comprises goods type, goods</td>
</tr>
</tbody>
</table>

**Table K.4 Resource**
<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FreightWeight</td>
<td>freight weight, item weight, good weight, weight</td>
<td>The weight of the freight that has to be transported.</td>
</tr>
<tr>
<td>FreightSize</td>
<td>freight size, item size, good size, size, volume</td>
<td>This concept includes the size of freight that will be transported. The freight size is specified by the volume, i.e. breadth, length, and height, of the item.</td>
</tr>
<tr>
<td>Carrier</td>
<td>shipper, transportation service provider, transporting party, carrier, forwarder</td>
<td>The party that ships the good. Carrier is also known as a shipper or freight forwarder.</td>
</tr>
<tr>
<td>Addressee</td>
<td>consignee, addressee</td>
<td>The party or business partner that is the addressee of the freight.</td>
</tr>
<tr>
<td>Addressee</td>
<td>customer, consignor, addresser</td>
<td>The party or business partner that is the addressee of the good, item or freight.</td>
</tr>
<tr>
<td>ActiveMeansOfTransport</td>
<td>active means of transport, means of transport, vehicle resource, vehicle, active vehicle,</td>
<td>Active means of transport are moving vehicle resources, including trucks, planes and ships, which can transport goods between location. An active means of transport is a vehicle, which can be driven independently.</td>
</tr>
<tr>
<td>PassiveMeansOfTransport</td>
<td>passive means of transport, transportation unit</td>
<td>Active means of transport are compartments or container that can be moved by an active means of transport.</td>
</tr>
<tr>
<td>Truck</td>
<td>truck, lorry, road train</td>
<td>A lorry is a motor vehicle designed to transport cargo.</td>
</tr>
<tr>
<td>Vessel</td>
<td>ship, vessel</td>
<td>A vessel transports goods via sea, rivers or lakes.</td>
</tr>
<tr>
<td>Plane</td>
<td>aircraft, plane</td>
<td>An plane is capable to flight using forward motion that generates lift as the wing moves through the air.</td>
</tr>
<tr>
<td>Railway</td>
<td>train, railway</td>
<td>A railway is a connected series of vehicles for rail transport that move along a track (permanent way) to transport cargo or passengers from one place to another.</td>
</tr>
<tr>
<td>Compart</td>
<td>container, compartment</td>
<td>A compartment that can be transported via ship, train or truck.</td>
</tr>
<tr>
<td>RailwayCar</td>
<td>railway car, rail car</td>
<td>A railway car is a vehicle on a rail transport system (railroad or railway) that is used for the carrying of cargo or passengers.</td>
</tr>
<tr>
<td>RequestedTime</td>
<td>requested dates, requested time</td>
<td>Requested time defines a time period from when to when a freight or cargo can be picked up.</td>
</tr>
<tr>
<td></td>
<td>Acceptable Time</td>
<td>acceptable dates, acceptable time</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>23</td>
<td>Memo</td>
<td>memo, note</td>
</tr>
<tr>
<td>24</td>
<td>Dossier</td>
<td>dossier</td>
</tr>
</tbody>
</table>

Table K.5 OntologyConcept

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasFreightData</td>
<td>item data, good data, freight data</td>
<td>Comprises detailed definitions of the freight being transported.</td>
<td>[1,1]</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>hasBusinessParticipant</td>
<td>business partner, business participant</td>
<td>Comprises at least three business participants such as shipper, consignee, and the customer.</td>
<td>[3,-1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>hasTransportationResource</td>
<td>resource, transportation resource</td>
<td>Transportation resources are divided into transportation units and vehicle resources. You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td>[0,-1]</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>hasMemo</td>
<td>note, memo</td>
<td>Within a transportation request you can enter any number of language dependent memos. Memos are categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print and communicate.</td>
<td>[0,1]</td>
<td>1</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>Property</td>
<td>Description</td>
<td>cardinality</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------------</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>hasDocuments</td>
<td>document, dossier</td>
<td>[0,1] 1 24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A document allows you to enter document references, information on required</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>documents, and file attachments (e.g. scanned documents for shipment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>request) and make them accessible to processors.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>hasAddress</td>
<td>address</td>
<td>[1,1] 2 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A business participant does always have an address.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>hasAddress</td>
<td>address, address location</td>
<td>[1,1] 5 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>A location has an address</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>hasRequestedTime</td>
<td>requested time, requested dates</td>
<td>[1,1] 6 21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requested time define a time period from when to when a freight or cargo</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>can be picked up.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>hasAcceptableTime</td>
<td>acceptable dates, acceptable time</td>
<td>[1,1] 6 22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Acceptable time define a time period from when (min) to when (max) a</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>freight or cargo can be picked up.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>hasFreightWeight</td>
<td>freight weight, good weight, item weight, weight</td>
<td>[0,1] 7 8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The weight of the freight that has to be transported.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>hasFreightSize</td>
<td>freight size, item size, good size, size, volume</td>
<td>[0,1] 7 9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This property includes the size of freight that will be transported.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The freight size is specified by the volume, i.e. breadth, length, and</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>height, of the item.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>hasSource</td>
<td>source location, customer location, consignor location, addresser location,</td>
<td>[1,1] 7 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>source location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the location from where the freight will be shipped.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>hasPickupPeriod</td>
<td>pickup period, pickup time, pickup time period, pickup date</td>
<td>[1,1] 7 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The time period in which the item or good can be picked up.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>hasDestination</td>
<td>destination location, consignee location, addressee location</td>
<td>[1,1] 7 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>This is the location where the freight will be shipped to.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>hasDeliveryTime</td>
<td>delivery period, delivery time, delivery time period, delivery date</td>
<td>[1,1] 7 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The time in which the item or good can be delivered.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table K.6 ObjectProperty

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasFreightNumber</td>
<td>freight number, good number, number, identifier, item identifier, good identifier</td>
<td>The number or the identifier of the freight that will be transported.</td>
<td>[1,1]</td>
<td>7</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>hasFreightDescription</td>
<td>description, item description</td>
<td>Some text that describes the freight.</td>
<td>[0,1]</td>
<td>7</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasStreet</td>
<td>street</td>
<td>The name of the street of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasNumber</td>
<td>street number, house number, number</td>
<td>The number of the house in the street.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasZIP</td>
<td>zip, zip code, area code, zip area code</td>
<td>The zip code of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>7</td>
<td>hasCountry</td>
<td>country, country code</td>
<td>The country of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>hasDateTimeFrom</td>
<td>date from, date time from</td>
<td>The date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>21</td>
<td>DateTime</td>
</tr>
<tr>
<td>9</td>
<td>hasDateTimeTo</td>
<td>date to, date time to</td>
<td>The date time to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>21</td>
<td>DateTime</td>
</tr>
<tr>
<td>10</td>
<td>hasDateTimeEarliest</td>
<td>earliest date time, date time from</td>
<td>The earliest date time from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>22</td>
<td>DateTime</td>
</tr>
<tr>
<td>11</td>
<td>hasDateTimeLatest</td>
<td>latest date time, date time to</td>
<td>The latest date time to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>22</td>
<td>DateTime</td>
</tr>
<tr>
<td>12</td>
<td>hasWeightValue</td>
<td>weight, weight value</td>
<td>Comprises the value of the good weight or item weight respectively.</td>
<td>[1,1]</td>
<td>8</td>
<td>Decimal</td>
</tr>
<tr>
<td>13</td>
<td>hasWeightUnit</td>
<td>weight unit</td>
<td>Comprises the weight unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>8</td>
<td>String</td>
</tr>
<tr>
<td>14</td>
<td>hasBreadth</td>
<td>breadth, good breadth, item</td>
<td>The breadth of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
</tr>
<tr>
<td></td>
<td>breadth</td>
<td>description</td>
<td>type</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>hasLength</td>
<td>length, good length, item length. The length of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>hasHeight</td>
<td>height, good height, item height. The height of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
<td>Decimal</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>hasLengthUnit</td>
<td>length unit. Comprises the length unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>9</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

*Table K.7 DataProperty*
Appendix L

ES SAP TM – Order Management restructured\textsuperscript{45}

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transportation Management</td>
<td>transport management, transportation management</td>
<td>Transport management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Order Management</td>
<td>order management</td>
<td>Order management and order receipt are generally the beginning of an operational process in transportation management.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Order Receipt</td>
<td>order receipt, order inquiry, receive order, receive request</td>
<td>Order receipt comprises a series of key operations that can be performed in accordance with a shipment request.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

\textit{Table L.1 Service}

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transportation Service Provider</td>
<td>transportation service provider</td>
<td>The transportation provider manages the process of order from order entry to order confirmation.</td>
<td>P</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Ordering Party</td>
<td>client, orderer, customer, ordering party</td>
<td>The ordering party sends a request and gets order confirmation.</td>
<td>C, I</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>Order Manager</td>
<td>order manager, shipment request manager</td>
<td>The order manager is responsible for creating, processing and confirming the shipment request.</td>
<td>R, Ctrl</td>
<td>3</td>
</tr>
</tbody>
</table>

\textit{Table L.2 Stakeholder}

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateShipmentRequest</td>
<td>create shipment request, enter shipment request, create shipment request entering, create shipment request entering, enter transport request, transport request entering</td>
<td>A shipment request is created either by directly creating it or by copying a reference shipment request.</td>
<td>3</td>
</tr>
</tbody>
</table>

\textsuperscript{45} The full XML code can be found on the CD supplied.
d:\ValidationScenario\ES TM_OrdReceipt_b_restructured.scmn

- 284 -
If the shipment request has not been completely and consistently created during the first entry, it can be completed in a second or further processing step.

Within this action the shipment request will be completed and confirmed.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action ID</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Changed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>Confirmed</td>
<td></td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

- **Table L.3 Action**

- **Table L.4 Resource**

A location is a place somewhere in the universe which is defined by an address.
<table>
<thead>
<tr>
<th></th>
<th>PickupDeliveryPeriod</th>
<th>Pickup delivery period</th>
<th>Pickup and delivery period defines the periods for pickup and delivery of freight, goods or items</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>ItemData</td>
<td>item data, goods data, freight data</td>
<td>Comprises detailed definitions of the goods being transported. It comprises goods type, goods description, product numbers, marks/numbers, and so on. In additional subnotes you can enter any measurement value (weights, volumes, numbers, dimensions), value indications, dangerous goods indication, customs details, and packaging details. All fields, with the exception of the goods description and quantity indication, are optional.</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>Shipper</td>
<td>shipper, transportation service provider, transporting party, carrier, forwarder</td>
<td>The party that ships the good. Shipper is also known as a carrier or freight forwarder.</td>
<td>2</td>
</tr>
<tr>
<td>9</td>
<td>Consignee</td>
<td>consignee, addressee</td>
<td>The party or business partner that is the consignee or addressee of the good, item or freight.</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>Customer</td>
<td>customer, consigner, addresser</td>
<td>The party or business partner that is the customer, addresser or consigner of the good, item or freight.</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>VehicleResource</td>
<td>vehicle resource, vehicle, active vehicle, active means of transport, means of transport</td>
<td>Vehicle resources are moving resources, including trucks, planes and ships, which can transport goods between locations. A vehicle resource is always an active vehicle, which can be driven independently.</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>TransportationUnit</td>
<td>transportation unit</td>
<td>The unit with which the good or item will be transported.</td>
<td>4</td>
</tr>
<tr>
<td>13</td>
<td>Truck</td>
<td>truck, lorry, road train</td>
<td>A truck or lorry is a motor vehicle designed to transport cargo.</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>Ship</td>
<td>ship, vessel</td>
<td>A ship transports goods via sea, rivers or lakes.</td>
<td>13</td>
</tr>
<tr>
<td>15</td>
<td>Aircraft</td>
<td>aircraft, plane</td>
<td>An aircraft is capable of flight using forward motion that generates lift as the wing moves through the air.</td>
<td>13</td>
</tr>
<tr>
<td>16</td>
<td>Train</td>
<td>train, railway</td>
<td>A train is a connected series of vehicles for rail transport that move along a track (permanent way) to transport cargo or passengers from one place to another.</td>
<td>13</td>
</tr>
<tr>
<td>17</td>
<td>Container</td>
<td>container, compartment</td>
<td>A container that can be transported via ship, train or truck.</td>
<td>14</td>
</tr>
<tr>
<td>18</td>
<td>RailwayCar</td>
<td>railway car</td>
<td>A railroad car is a vehicle on a rail transport system (railroad or railway) that is used for the carrying of cargo or passengers.</td>
<td>14</td>
</tr>
<tr>
<td>19</td>
<td>Note</td>
<td>note</td>
<td>Within a shipment request you can enter any number of language dependent notes. Notes are categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print</td>
<td>-</td>
</tr>
</tbody>
</table>
A dossier allows you to enter document references, information on required documents, and file attachments (e.g. scanned documents for shipment request) and make them accessible to processors.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItem-Data</td>
<td>item data, good data, freight data</td>
<td>Comprises detailed definitions of the goods being transported.</td>
<td>[1,1]</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>hasShopper</td>
<td>shipper, transportation service provider,</td>
<td>The party that ships the good. Shipper is also known as a carrier or freight forwarder.</td>
<td>[1,-1]</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>hasConsignee</td>
<td>consignee, addressee</td>
<td>The party that is the consignee or addressee of the good, item or freight.</td>
<td>[1,-1]</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>hasCustomer</td>
<td>customer, consignee, addressee</td>
<td>The party that is the customer, addressee or consignee of the good, item or freight.</td>
<td>[1,-1]</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>hasResource</td>
<td>resource</td>
<td>You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td>[0,-1]</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>hasNote</td>
<td>note</td>
<td>Within a shipment request you can enter any number of language dependent notes. Notes are categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print and communicate.</td>
<td>[0,1]</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>7</td>
<td>hasDossier</td>
<td>dossier</td>
<td>A dossier allows you to enter document references, information on</td>
<td>[0,1]</td>
<td>1</td>
<td>20</td>
</tr>
</tbody>
</table>
required documents, and file attachments (e.g. scanned documents for shipment request) and make them accessible to processors.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>hasAddress</td>
<td>address</td>
<td>A business partner does always have an address.</td>
<td>[1,1]</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>hasAddress</td>
<td>address, address location</td>
<td>A location has an address</td>
<td>[1,1]</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>hasSourceLocation</td>
<td>source location, customer location, consignee location, addressee location</td>
<td>This is the location from where the good or item will be shipped.</td>
<td>[1,1]</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>hasPickupPeriod</td>
<td>pickup period, pickup time, pickup time period, pickup date</td>
<td>The period in which the item or good can be picked up.</td>
<td>[1,1]</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>hasDestinationLocation</td>
<td>destination location, consignee location, addressee location</td>
<td>This is the location where the good or item will be shipped to.</td>
<td>[1,1]</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>hasDeliveryPeriod</td>
<td>delivery period, delivery time, delivery time period, delivery date</td>
<td>The period in which the item or good can be delivered.</td>
<td>[1,1]</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

Table L.6 ObjectProperty

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItemNumber</td>
<td>item number, good number, number, identifier, item identifier, good identifier</td>
<td>The number or the identifier of the item that will be transported.</td>
<td>[1,1]</td>
<td>7</td>
<td>Integer</td>
</tr>
<tr>
<td>2</td>
<td>hasItemDescription</td>
<td>description, item description</td>
<td>Some text that describes the item or good.</td>
<td>[0,1]</td>
<td>7</td>
<td>String</td>
</tr>
<tr>
<td>3</td>
<td>hasStreet</td>
<td>street</td>
<td>The name of the street of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>hasNumber</td>
<td>street number, house number, number</td>
<td>The number of the house in the street.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>hasCity</td>
<td>city, town</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>6</td>
<td>hasZIP</td>
<td>zip, zip code</td>
<td>The zip code of the</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>hasCountry</td>
<td>country, country code</td>
<td>The country of the address</td>
<td>[1,1]</td>
<td>3</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>hasDate-From</td>
<td>date from</td>
<td>The date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>DateTime</td>
</tr>
<tr>
<td>9</td>
<td>hasDateTo</td>
<td>date to</td>
<td>The date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>DateTime</td>
</tr>
<tr>
<td>10</td>
<td>hasEarliestDate</td>
<td>earliest date, earliest date from</td>
<td>The earliest date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>DateTime</td>
</tr>
<tr>
<td>11</td>
<td>hasLatestDate</td>
<td>latest date, latest date to</td>
<td>The latest date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>DateTime</td>
</tr>
<tr>
<td>12</td>
<td>has-WeightValue</td>
<td>weight, weight value</td>
<td>Comprises the value of the good weight or item weight respectively.</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
</tr>
<tr>
<td>13</td>
<td>hasWeightUnit</td>
<td>weight unit</td>
<td>Comprises the weight unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>7</td>
<td>String</td>
</tr>
<tr>
<td>14</td>
<td>hasBreadth</td>
<td>breadth, good breadth, item breadth</td>
<td>The breadth of the item or good.</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
</tr>
<tr>
<td>15</td>
<td>hasLength</td>
<td>length, good length, item length</td>
<td>The length of the item or good.</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
</tr>
<tr>
<td>16</td>
<td>hasHeight</td>
<td>height, good height, item height</td>
<td>The height of the item or good.</td>
<td>[1,1]</td>
<td>7</td>
<td>Decimal</td>
</tr>
<tr>
<td>17</td>
<td>hasSizeUnit</td>
<td>Size unit</td>
<td>Comprises the size unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>7</td>
<td>String</td>
</tr>
</tbody>
</table>

*Table L.7 DataProperty*
Appendix M

ES SAP TM – Order Management missing feature

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportMgmt</td>
<td>transport management, transportation management</td>
<td>Transport management comprises management and coordination of moving good from one location to another one.</td>
<td>A</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Order-Management</td>
<td>order management</td>
<td>Order management and order receipt are generally the beginning of an operational process in transportation management.</td>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Order-Receipt</td>
<td>order receipt, order inquiry, receive order, receive request</td>
<td>Order receipt comprises a series of key operations that can be performed in acceptance a shipment request.</td>
<td>C</td>
<td>2</td>
</tr>
</tbody>
</table>

Table M.1 Service

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Type</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TransportationServiceProvider</td>
<td>transportation service provider, order manager</td>
<td>The transportation service provider manages the process of order from order entry to order confirmation.</td>
<td>P, Ctrl, R</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>OrderingParty</td>
<td>client, orderer, customer, ordering party</td>
<td>The ordering party sends a request and gets order confirmation.</td>
<td>C, I</td>
<td>3</td>
</tr>
</tbody>
</table>

Table M.2 Stakeholder

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CreateShipmentRequest</td>
<td>create shipment request, enter shipment request, shipment request entering, create shipment request, enter transport request, transport request entering</td>
<td>A shipment request is created either by directly creating it or by copying a reference shipment request.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>ProcessShipmentRequest</td>
<td>shipment request processing, shipment request completion, transport request</td>
<td>If the shipment request has not been completely and consistently created during the first entry, it can be completed in a second or further processing step.</td>
<td>3</td>
</tr>
</tbody>
</table>

55 The full XML code can be found on the CD supplied.
d:\ValidationScenario\ES_TM_OrdReceipt_c_modified.scmn
processing, transport request completion

<table>
<thead>
<tr>
<th>ID</th>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Activate-ShipmentRequest</td>
<td>When the shipment request is full described it can be activated.</td>
</tr>
<tr>
<td>4</td>
<td>Confirm-ShipmentRequest</td>
<td>After shipment request activation it is confirmed to the customer.</td>
</tr>
</tbody>
</table>

Table M.3 Action

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Type</th>
<th>InvolvementType</th>
<th>State Start</th>
<th>State End</th>
<th>Action</th>
<th>Concept ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Created</td>
<td>NoState</td>
<td>New</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>New</td>
<td>Changed</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Changed</td>
<td>Activated</td>
<td>Activated</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>ShipmentRequest</td>
<td>I</td>
<td>Activated</td>
<td>Confirmed</td>
<td>Confirmed</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table M.4 Resource

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Super ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ShipmentRequest</td>
<td>shipment request, transport request, freight request, transpor</td>
<td>A shipment request is an agreement between a transportation service provider and a ordering party regarding the transportation of goods or transportation equipment from an issuing partner or location to a receiving partner or location according to the agreed conditions.</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Business-Partner</td>
<td>business partner</td>
<td>The subnotes for the business partners, aside from the mandatory specifications Shipper, Consignee and Customer, allow further parties involved in the transportation operation to be defined. These may be, for example, predetermined service providers (e.g. if requested by the customer), agents, or invoice recipients.</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>Address</td>
<td>address</td>
<td>Represents all data items of an address.</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>Resource</td>
<td>resource</td>
<td>Resources are divided into transportation units and vehicle resources. You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>Location</td>
<td>location, address</td>
<td>A location is a place somewhere in the universe.</td>
<td>-</td>
</tr>
</tbody>
</table>
which is defined by an address

<table>
<thead>
<tr>
<th></th>
<th>PickupDeliveryPeriod</th>
<th>pickup delivery period</th>
<th>Pickup and delivery period defines the periods for pickup and delivery of freight, goods or items</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>ItemData</td>
<td>item data, goods data, freight data</td>
<td>Comprises detailed definitions of the goods being transported. It comprises goods type, goods description, product numbers, marks/numbers, and so on. In additional subnotes you can enter any measurement value (weights, volumes, numbers, dimensions), value indications, dangerous goods indication, customs details, and packaging details. All fields, with the exception of the goods description and quantity indication, are optional.</td>
</tr>
<tr>
<td>8</td>
<td>Gross-Weight</td>
<td>gross weight, item weight, good weight, weight</td>
<td>The weight of the item or good that has to be transported.</td>
</tr>
<tr>
<td>9</td>
<td>GrossSize</td>
<td>gross size, item size, good size, size, volume</td>
<td>This property includes the size of the item or good that will be transported. The gross size is specified by the volume, i.e. breadth, length, and height, of the item.</td>
</tr>
<tr>
<td>10</td>
<td>Shipper</td>
<td>shipper, transportation service provider, transporting party, carrier, forwarder</td>
<td>The party that ships the good. Shipper is also known as a carrier or freight forwarder.</td>
</tr>
<tr>
<td>11</td>
<td>Consignee</td>
<td>consignee, addressee</td>
<td>The party or business partner that is the consignee or addressee of the good, item or freight.</td>
</tr>
<tr>
<td>12</td>
<td>Customer</td>
<td>customer, consignor, addresser</td>
<td>The party or business partner that is the customer, addresser or consignor of the good, item or freight.</td>
</tr>
<tr>
<td>13</td>
<td>VehicleResource</td>
<td>vehicle resource, vehicle, active vehicle, active means of transport, means of transport</td>
<td>Vehicle resources are moving resources, including trucks, planes and ships, which can transport goods between location. A vehicle resource is always an active vehicle, which can be driven independently.</td>
</tr>
<tr>
<td>14</td>
<td>TransportationUnit</td>
<td>transportation unit</td>
<td>The unit with which the good or item will be transported.</td>
</tr>
<tr>
<td>15</td>
<td>Truck</td>
<td>truck, lorry, road train</td>
<td>A truck or lorry is a motor vehicle designed to transport cargo.</td>
</tr>
<tr>
<td>16</td>
<td>Container</td>
<td>container, compartment</td>
<td>A container that can be transported via ship, train or truck.</td>
</tr>
<tr>
<td>17</td>
<td>RequestedDates</td>
<td>requested dates</td>
<td>Requested dates define a time period from when to when a freight or cargo can be picked up.</td>
</tr>
<tr>
<td>18</td>
<td>AcceptableDates</td>
<td>acceptable dates</td>
<td>Acceptable dates define a time period from when (min) to when (max) a freight or cargo can be picked up.</td>
</tr>
<tr>
<td>19</td>
<td>Note</td>
<td>note</td>
<td>Within a shipment request you can enter any number of language dependent notes. Notes are</td>
</tr>
</tbody>
</table>
categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print and communicate.

20  Dossier  dossier  A dossier allows you to enter document references, information on required documents, and file attachments (e.g. scanned documents for shipment request) and make them accessible to processors.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>Range ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>hasItem-Data</td>
<td>item data, good data, freight data</td>
<td>Comprises detailed definitions of the goods being transported.</td>
<td>[1,1]</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>hasBusinessPartner</td>
<td>business partner</td>
<td>Comprises at least three business partners, such as shipper, consignee, and the customer.</td>
<td>[3,-1]</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>hasResource</td>
<td>resource</td>
<td>You can either use resources here to defined resources that are provided on the customer side (e.g. customer requests goods to be collected already in packaged form in a container) or you can make a note of special resource requirements from the customer here.</td>
<td>[0,-1]</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>hasNote</td>
<td>note</td>
<td>Within a shipment request you can enter any number of language dependent notes. Notes are categorized user-defined texts (e.g. category shipping note) that can be used both to pass on information in the processing chain and to print and communicate.</td>
<td>[0,1]</td>
<td>1</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>hasDossier</td>
<td>dossier</td>
<td>A dossier allows you to enter document references, information on required documents, and file attachments (e.g. scanned documents for shipment request) and make them accessible to processors.</td>
<td>[0,1]</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>6</td>
<td>hasAddress</td>
<td>address</td>
<td>A business partner does</td>
<td>[1,1]</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
always has an address.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Labels</th>
<th>Description</th>
<th>Card.</th>
<th>Domain ID</th>
<th>DataType</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>hasAddress</td>
<td>address, address location</td>
<td>A location has an address</td>
<td>[1,1]</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>hasRequestedDates</td>
<td>requested dates</td>
<td>Requested dates define a time period from when to when a freight or cargo can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>hasAcceptableDates</td>
<td>acceptable dates</td>
<td>Acceptable dates define a time period from when (min) to when (max) a freight or cargo can be picked up.</td>
<td>[1,1]</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>10</td>
<td>hasGrossWeight</td>
<td>gross weight, good weight, item weight, weight</td>
<td>The weight of the item or good that has to be transported.</td>
<td>[0,1]</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>hasGrossSize</td>
<td>gross size, item size, good size, size</td>
<td>This property includes the size of the item or good that will be transported.</td>
<td>[0,1]</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>hasSourceLocation</td>
<td>source location, customer location, consignor location, addressee location</td>
<td>This is the location from where the good or item will be shipped.</td>
<td>[1,1]</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>13</td>
<td>hasPickupPeriod</td>
<td>pickup period, pickup time, pickup time period, pickup date</td>
<td>The period in which the item or good can be picked up.</td>
<td>[1,1]</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>14</td>
<td>hasDestinationLocation</td>
<td>destination location, consignee location, addressee location</td>
<td>This is the location where the good or item will be shipped to.</td>
<td>[1,1]</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>hasDeliveryPeriod</td>
<td>delivery period, delivery time, delivery time period, delivery date</td>
<td>The period in which the item or good can be delivered.</td>
<td>[1,1]</td>
<td>7</td>
<td>6</td>
</tr>
</tbody>
</table>

Table M.6 ObjectProperty
<table>
<thead>
<tr>
<th>No.</th>
<th>Property</th>
<th>Description</th>
<th>Range</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>hasStreet</td>
<td>The name of the street of the address.</td>
<td>[1,1]</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>hasNumber</td>
<td>The number of the house in the street.</td>
<td>[1,1]</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>hasCity</td>
<td>The city or the town of the address.</td>
<td>[1,1]</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>hasZIP</td>
<td>The zip code of the address.</td>
<td>[1,1]</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>hasCountry</td>
<td>The country of the address.</td>
<td>[1,1]</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>hasDate-From</td>
<td>The date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>17</td>
</tr>
<tr>
<td>9</td>
<td>hasDate-To</td>
<td>The date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>17</td>
</tr>
<tr>
<td>10</td>
<td>hasEarliestDate</td>
<td>The earliest date from which the freight can be picked up.</td>
<td>[1,1]</td>
<td>18</td>
</tr>
<tr>
<td>11</td>
<td>hasLatest-Date</td>
<td>The latest date to which the freight can be picked up.</td>
<td>[1,1]</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>hasWeight-Value</td>
<td>Comprises the value of the good weight or item weight respectively.</td>
<td>[1,1]</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td>hasWeightUnit</td>
<td>Comprises the weight unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>8</td>
</tr>
<tr>
<td>14</td>
<td>hasBreadth</td>
<td>The breadth of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
</tr>
<tr>
<td>15</td>
<td>hasLength</td>
<td>The length of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
</tr>
<tr>
<td>16</td>
<td>hasHeight</td>
<td>The height of the item or good.</td>
<td>[1,1]</td>
<td>9</td>
</tr>
<tr>
<td>17</td>
<td>hasLengthUnit</td>
<td>Comprises the length unit of the good or item that has to be transported.</td>
<td>[1,1]</td>
<td>9</td>
</tr>
</tbody>
</table>

*Table M.7 DataProperty*
### Appendix N

<table>
<thead>
<tr>
<th>Noun</th>
<th>Synset (Description)</th>
</tr>
</thead>
</table>
| **transport** | - conveyance, transport (something that serves as a means of transportation)  
- transport (an exchange of molecules (and their kinetic energy and momentum) across the boundary between adjacent layers of a fluid or across cell membranes)  
- transportation, shipping, transport (the commercial enterprise of moving goods and materials)  
- ecstasy, rapture, transport, exaltation, raptus (a state of being carried away by overwhelming emotion) ‘listening to sweet music in a perfect rapture’. Charles Dickens  
- tape drive, tape transport, transport (a mechanism that transports magnetic tape across the read/write heads of a tape playback/recorder)  
- transportation, transport, transfer, transference, conveyance (the act of moving something from one location to another) |
| **order** | - order (often plural) a command given by a superior (e.g., a military or law enforcement officer) that must be obeyed)  
- order, order of magnitude (a degree in a continuum of size or quantity)  
- order (established customary state (especially of society))  
- ordering, order, ordination (logical or comprehensible arrangement of separate elements)  
- orderliness, order (a condition of regular or proper arrangement)  
- decree, edict, fiat, order, rescript (a legally binding command or decision entered on the court record (as if issued by a court or judge))  
- order, purchase order (a commercial document used to request someone to supply something in return for payment and providing specifications and quantities)  
- club, social club, society, guild, gild, lodge, order (a formal association of people with similar interests)  
- order, rules of order, parliamentary law, parliamentary procedure (a body of rules followed by an assembly)  
- Holy Order, Order (usually plural) the status or rank or office of a Christian clergyman in an ecclesiastical hierarchy)  
- order, monastic order (a group of person living under a religious rule)  
- order (biology) taxonomic group containing one or more families)  
- order (a request for something to be made, supplied, or served)  
- order (architecture) one of original three styles of Greek architecture distinguished by the type of column and entablature used or a style developed from the original three by the Romans)  
- order, ordering (the act of putting things in a sequential arrangement) |
| **source** | - beginning, origin, root, rootage, source (the place where something begins, where it springs into being)  
- source (a document (or organization) from which information is obtained)  
- source, seed, germ (anything that provides inspiration for later work)  
- source (a facility where something is available)  
- informant, source (a person who supplies information)  
- generator, source, author (someone who originates or causes or initiates something)  
- source (technology) a process by which energy or a substance enters a system)  
- reservoir, source (anything (a person or animal or plant or substance) in which an infectious agent normally lives and multiplies)  
- reference, source (a publication (or a passage from a publication) that is referred to) |
| **address** | - address, computer address, reference (computer science) the code that identifies where a piece of information is stored)  
- address (the place where a person or organization can be found or communicated with)  
- address, speech (the act of delivering a formal spoken communication to an audience)  
- address (the manner of speaking to another individual)  
- address (a sign in front of a house or business carrying the conventional form by which its location is described)  
- address, destination, name and address (written directions for finding some location; written on letters or packages that are to be delivered to that location)  
- address (the stance assumed by a golfer in preparation for hitting a golf ball)  
- savoir-faire, address (social skill) |
| **destination** | - finish, destination, goal (the place designated as the end (as of a race or journey))  
- destination, terminus (the ultimate goal for which something is done)  
- address, destination, name and address (written directions for finding some location; written |
on letters or packages that are to be delivered to that location)

- pickup, pickup truck (a light truck with an open body and low sides and a tailboard)
- pickup (a warrant to take someone into custody)
- pickup, pick-me-up (anything with restorative powers)
- pickup (a casual acquaintance; often made in hope of sexual relationships)
- pickup, getaway (the attribute of being capable of rapid acceleration)
- tone arm, pickup, pickup arm (mechanical device consisting of a light balanced arm that carries the cartridge)
- cartridge, pickup (an electro-acoustic transducer that is the part of the arm of a record player that holds the needle and that is removable)
- pickup (the act or process of picking up or collecting from various places)
- pickup (the act of taking aboard passengers or freight)

- time, time, clip (an instance or single occasion for some event)
- time (a period of time considered as a resource under your control and sufficient to accomplish something)
- time (an indefinite period (usually marked by specific attributes or activities))
- time (a suitable moment)
- time (the continuum of experience in which events pass from the future through the present to the past)
- time (a person's experience on a particular occasion)
- clock time, time (a reading of a point in time as given by a clock)
- fourth dimension, time (the fourth coordinate that is required (along with three spatial dimensions) to specify a physical event)
- meter, metre, time (rhythm as given by division into parts of equal duration)
- prison term, sentence, time (the period of time a prisoner is imprisoned)

- good, good (benefit)
- good, goodness (moral excellence or admirableness)
- good, goodness (that which is pleasing or valuable or useful)
- commodity, trade good, good (articles of commerce)

- truck, motortruck (an automotive vehicle suitable for hauling)
- hand truck, truck (a handcart that has a frame with two low wheels and a ledge at the bottom and handles at the top; used to move crates or other heavy objects)

- cargo, lading, freight, load, loading, payload, shipment, consignment (goods carried by a large vehicle)
- cargo, lading, freight, load, loading, payload, shipment, consignment (goods carried by a large vehicle)
- dispatch, despatch, shipment (the act of sending off something)

- customer, customer, client (someone who pays for goods or services)
- consignee (the person to whom merchandise is delivered over)

- period, period of time, period (an amount of time)
- period (the interval taken to complete one cycle of a regularly repeating phenomenon)
- period (ice hockey) one of three divisions into which play is divided in hockey games)
- period, geological period (a unit of geological time during which a system of rocks formed)
- period (the end or completion of something)
- menstruation, menses, menstruum, catamenia, period, flow (the monthly discharge of blood from the uterus of nonpregnant women from puberty to menopause)
- period, point, full stop, stop, full point (a punctuation mark (.) placed at the end of a declarative sentence to indicate a full stop or after abbreviations)

- item, point (a distinct part that can be specified separately in a group of things that could be enumerated on a list)
- detail, particular, item (a small part that can be considered separately from the whole)
- item (a whole individual unit; especially when included in a list or collection)
- detail, item, point (an isolated fact that is considered separately from the whole)
- token, item (an individual instance of a type of symbol)

Table N.1 Synset example for some logistics
Curriculum vitae

Personal details
Name: Rolf Kluge
Address: Holbeinstr. 14, 04229 Leipzig, Germany
Date of birth: 15th of October 1980
Place of birth: Leipzig, Germany
Nationality: Federal Republic of Germany
Parents: Dr. Gert Kluge, Diploma and Doctor of Chemistry, District Manager at UCB Germany (pharmaceutical company)
Christine Kluge, née Blaha, Teacher, Primary school teacher
Language: German, English (business fluent), French (3 years schooling)
Hobbies: 12 years judo as competitive sport, Roadbike, Football

Career details
Sep 1987 to Jun 1992  94. Oberschule Leipzig (primary school)
Sep 1992 to Jul 1999  Sportgymnasium Leipzig
                     (secondary school with a specialty on sport)
                     finished with Abitur (roughly equivalent to A levels)
                     final grade: 1,8 (good)
Jul 1999 to Apr 2000  Military service at the German Federal Armed Forces
Oct 1999 to Apr 2001  Distance learning at SGD
                     finished with SGD-Diploma as C/C++ programmer,
                     final grade 1,0 (very good)
Jul 2000 to Apr 2002  Employment at the Software-Company
                     SMB GmbH Leipzig (www.smb-tec.com)
                     field of activity: java programming, XML-technologies
Oct 2000 to Oct 2006  Study of Information Systems University of Leipzig
                     specialization: „Enterprise Application Integration“ and „Controlling“, finished with Diploma of Information
                     Systems, final grade: 1,9 (good)
Aug 2002 to Sept 2006  Executive director at Kluge & Schmidt GbR
                     (partnership under the civil code),
field of activity: development and distribution of internet shop solutions

Oct 2003 to Sep 2005 Employee as a working student at the Microsoft Deutschland GmbH (Microsoft Germany)

2004 and 2005 Attendee at the Microsoft student technology competition Imagine Cup
May 2004 – 2nd best Germany, project SmartShopping
May 2005 – 1st best Germany, project SmartRunner
Aug 2005 – 4th-best worldwide in Yokohama/Japan

Jan 2006 to Jun 2006 Diploma thesis at the DaimlerChrysler AG,
Title: “User Authentication within an interoperable Environment of Java 2 Enterprise Edition and Microsoft Windows .NET, regarding to the IT-Infrastructure at DaimlerChrysler AG”, 90 Pages, English, 90 p., final grade: 1,3 (very good)

Oct 2006 to Sept 2011 Research associate and PhD-Student at the University of Leipzig, Information Systems Institute, Faculty of Economics, specialization: service oriented architecture, business service engineering, semantics, ontologies (scope of application: logistics and service for mobile devices)

Since Dec 2008 PhD-Student at the Macquarie University Sydney Faculty of Science, Department of Computing

Since Oct 2011 Co-founder and chief technical officer at Smartrunner GmbH

................................................
(Signature)
Selbstständigkeitserklärung

Hiermit erkläre ich, die vorliegende Dissertation selbstständig und ohne unzuläs-
sige fremde Hilfe angefertigt zu haben. Ich habe keine anderen als die angeführten
Quellen und Hilfsmittel benutzt und sämtliche Textstellen, die wörtlich oder sinn-
gemäß aus veröffentlichten oder unveröffentlichten Schriften entnommen wurden,
und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich
gemacht. Ebenfalls sind alle von anderen Personen bereitgestellten Materialien
oder erbrachten Dienstleistungen als solche gekennzeichnet.

…………………
(Ort, Datum)

……………………………………………
(Unterschrift)