An Enterprise Ontology based Approach to Service Specification

Gde Hardjosumarto
An Enterprise Ontology based Approach to Service Specification

THESIS

submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Gde Hardjosumarto
born in Amsterdam, the Netherlands
An Enterprise Ontology based Approach to Service Specification

Author: Gde Hardjosumarto
Student id: 1149962
Email: g.c.hardjosumarto@student.tudelft.nl

Abstract

Businesses operate in ever-changing business environments. Information and Communication Technology (ICT) systems must constantly adapt in order to meet changing business requirements. Yet, adaptation of ICT systems in order to cope with these changes is difficult, because ICT systems in an organization strongly cohere. Service-oriented architecture (SOA) enables ICT systems to cope with fast changing business requirements by providing ICT systems’ functionalities as services to other ICT systems. Because of this, service specification takes an important role in the realization of SOA. It provides a description of the functionality that a service offers and how the service can be used. It was found, however, that current frameworks for specifying services insufficiently focus on business people.

This thesis proposes a new framework for specifying services in SOA, which deals with the issue above. This framework is different from other current frameworks existed as it is based on a well-founded theory, which is the concept of 'enterprise ontology'.

Thesis Committee:

Chair: Prof. dr. ir. Jan Dietz, Faculty EEMCS, Delft University of Technology
University supervisor: Dr. dipl-ing Antonia Albani, Faculty EEMCS, Delft University of Technology
Company supervisor: Ir. Linda Terlouw, Ordina ICT B.V.
Committee Member: Dr. Joseph Barjis, Faculty TPM, Delft University of Technology
This document has been produced in conjunction with the Master Thesis project as part of the Master program Information Architecture at Delft University of Technology from March until October in 2008.

The Master Thesis project could not have been completed without the efforts and support of a number of people.

First, I would like to thank my supervisors for their guidance and great assistance during the project. I am very thankful for the support and help provided by the university supervisor, Antonia Albani, and the company supervisor, Linda Terlouw, during our meetings wherein borders appeared not to be any obstacle for productive discussions. In particular, I am Linda Terlouw grateful for giving me the opportunity to participate in an interesting research regarding an innovative paradigm in the field of business and ICT and to carry out the project in an inspiring environment as Ordina ICT B.V., which is one of the most innovative and progressive companies in the area of service-oriented architecture in the Netherlands.

Second, I would like to express my gratitude for the advices and help I received from people at Ordina ICT B.V. Their expertise and knowledge helped me to incorporate valuable insights from practice into a project which strongly focused on theory and science.

Third, I would like to express my appreciation to the two experts at the Dutch governmental organization for giving me the opportunity to conduct a case study regarding the use of SOA in their organization. I am thankful for their time and willingness to share their insight and knowledge regarding service specification in practice.

Last, but certainly not least, I would like to thank my family and girlfriend for their continuous support, care and understanding during the last months which were fully dominated by the Master Thesis project.

Gde Hardjosumarto
Delft, the Netherlands
October 20, 2008
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>iii</td>
</tr>
<tr>
<td>Contents</td>
<td>v</td>
</tr>
<tr>
<td>List of Figures</td>
<td>ix</td>
</tr>
<tr>
<td>List of Tables</td>
<td>xi</td>
</tr>
</tbody>
</table>

## I Preliminaries

### 1 Introduction

1.1 Background .................................. 3
1.2 Service Oriented Architecture ............ 3
1.3 Service ...................................... 4
1.4 Service Specification ...................... 5
1.5 Current Issues on Service Specification .. 6
1.6 Research Goal ................................ 6
1.7 Research Approach .......................... 7
1.8 Report Structure ............................ 9

### 2 Enterprise Ontology

2.1 Why Enterprise Ontology? .................. 11
2.2 The Operation Axiom ....................... 12
2.3 The Transaction Axiom ..................... 13
2.4 The Distinction Axiom ..................... 16
2.5 The Organization Theorem .................. 17

### 3 Related Work

3.1 The UDDI Specification Framework .......... 23
3.2 Business Component Specification .......... 24
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3</td>
<td>The Faceted Specification Approach</td>
<td>26</td>
</tr>
<tr>
<td>II</td>
<td>Towards a Service Specification Framework</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>Service in Enterprise Ontology</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>The Concept of ‘Service’</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Business Service in Enterprise Ontology</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>ICT Service in Enterprise Ontology</td>
<td>41</td>
</tr>
<tr>
<td>4.4</td>
<td>Composite Service and Service Component</td>
<td>44</td>
</tr>
<tr>
<td>4.5</td>
<td>Service Provider and Service Consumer</td>
<td>51</td>
</tr>
<tr>
<td>4.6</td>
<td>Evaluation</td>
<td>53</td>
</tr>
<tr>
<td>5</td>
<td>Service Specification in Enterprise Ontology</td>
<td>57</td>
</tr>
<tr>
<td>5.1</td>
<td>The Concept of ‘Service Specification’</td>
<td>57</td>
</tr>
<tr>
<td>5.2</td>
<td>Conditions</td>
<td>62</td>
</tr>
<tr>
<td>5.3</td>
<td>Business Service Specification</td>
<td>66</td>
</tr>
<tr>
<td>5.4</td>
<td>ICT Service Specification</td>
<td>68</td>
</tr>
<tr>
<td>5.5</td>
<td>Evaluation</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>Service Quality</td>
<td>75</td>
</tr>
<tr>
<td>6.1</td>
<td>The Concept of ‘Quality’</td>
<td>75</td>
</tr>
<tr>
<td>6.2</td>
<td>ICT Services and the Concept of ‘Quality’</td>
<td>76</td>
</tr>
<tr>
<td>6.3</td>
<td>Business Services and the Concept of ‘Quality’</td>
<td>78</td>
</tr>
<tr>
<td>6.4</td>
<td>Methods for Identifying Quality Aspects</td>
<td>81</td>
</tr>
<tr>
<td>6.5</td>
<td>Identification of Quality Aspects</td>
<td>84</td>
</tr>
<tr>
<td>6.6</td>
<td>Evaluation</td>
<td>87</td>
</tr>
<tr>
<td>III</td>
<td>The Generic Service Specification Framework</td>
<td>91</td>
</tr>
<tr>
<td>7</td>
<td>The Generic Service Specification Framework</td>
<td>93</td>
</tr>
<tr>
<td>7.1</td>
<td>Service Provider Information</td>
<td>93</td>
</tr>
<tr>
<td>7.2</td>
<td>Service Contract Information</td>
<td>95</td>
</tr>
<tr>
<td>7.3</td>
<td>Service Function Information</td>
<td>96</td>
</tr>
<tr>
<td>7.4</td>
<td>Service Usage Information</td>
<td>98</td>
</tr>
<tr>
<td>7.5</td>
<td>Terminology</td>
<td>99</td>
</tr>
<tr>
<td>8</td>
<td>Case Study</td>
<td>101</td>
</tr>
<tr>
<td>8.1</td>
<td>Research Questions</td>
<td>101</td>
</tr>
<tr>
<td>8.2</td>
<td>A Case Study as Research Method</td>
<td>102</td>
</tr>
<tr>
<td>8.3</td>
<td>Case Study Approach</td>
<td>102</td>
</tr>
<tr>
<td>8.4</td>
<td>Service Specification in the Organization</td>
<td>103</td>
</tr>
<tr>
<td>8.5</td>
<td>Verification of the Generic Service Specification Framework</td>
<td>106</td>
</tr>
<tr>
<td>8.6</td>
<td>Evaluation</td>
<td>109</td>
</tr>
</tbody>
</table>
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV Conclusions</td>
<td>111</td>
</tr>
<tr>
<td>9 Contributions</td>
<td>113</td>
</tr>
<tr>
<td>10 Conclusion</td>
<td>115</td>
</tr>
<tr>
<td>11 Future Work</td>
<td>119</td>
</tr>
<tr>
<td>Abbreviations</td>
<td>121</td>
</tr>
<tr>
<td>Bibliography</td>
<td>123</td>
</tr>
<tr>
<td>A Identification of Quality Aspects Using the Extended ISO-model</td>
<td>129</td>
</tr>
<tr>
<td>A.1 Functionality</td>
<td>129</td>
</tr>
<tr>
<td>A.2 Reliability</td>
<td>136</td>
</tr>
<tr>
<td>A.3 Usability</td>
<td>141</td>
</tr>
<tr>
<td>A.4 Efficiency</td>
<td>141</td>
</tr>
<tr>
<td>A.5 Maintainability</td>
<td>143</td>
</tr>
<tr>
<td>A.6 Portability</td>
<td>148</td>
</tr>
</tbody>
</table>
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Basic representation of SOA</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Research phases</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>Representation of the operation axiom [18]</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>The basic pattern of a transaction [18]</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>The standard pattern of a transaction [18]</td>
<td>14</td>
</tr>
<tr>
<td>2.4</td>
<td>Cancellation pattern of a request [18]</td>
<td>15</td>
</tr>
<tr>
<td>2.5</td>
<td>Cancellation pattern of a promise [18]</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>Cancellation pattern of a statement [18]</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>Cancellation pattern of an acceptance [18]</td>
<td>16</td>
</tr>
<tr>
<td>2.8</td>
<td>Representation of the distinction axiom [18]</td>
<td>17</td>
</tr>
<tr>
<td>2.9</td>
<td>The process of performing a coordination act [18]</td>
<td>18</td>
</tr>
<tr>
<td>2.10</td>
<td>Representation of the organization theorem [18]</td>
<td>19</td>
</tr>
<tr>
<td>2.11</td>
<td>The layered integration of an organization [18]</td>
<td>20</td>
</tr>
<tr>
<td>2.12</td>
<td>Organizations and ICT systems [18]</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>UDDI data types</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Specification levels of business components [4]</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>The faceted specification approach [60]</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Service according to enterprise ontology</td>
<td>37</td>
</tr>
<tr>
<td>4.2</td>
<td>A composite service in an enclosed transaction</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>The structure of a self-activation transaction</td>
<td>40</td>
</tr>
<tr>
<td>4.4</td>
<td>Decomposition of a composite service on the basis of production acts</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Decomposition of a composite service on the basis of single acts (1)</td>
<td>47</td>
</tr>
<tr>
<td>4.6</td>
<td>Decomposition of a composite service on the basis of single acts (2)</td>
<td>48</td>
</tr>
<tr>
<td>4.7</td>
<td>A composite service on the basis of delegation</td>
<td>50</td>
</tr>
<tr>
<td>4.8</td>
<td>The service providing party regarding ICT services and delegation in business services</td>
<td>52</td>
</tr>
<tr>
<td>4.9</td>
<td>The service providing party regarding business services</td>
<td>52</td>
</tr>
</tbody>
</table>
4.10 The service consuming party regarding ICT services and delegation in business services ........................................ 53
4.11 The service consuming party regarding business services ........................................ 54

5.1 Black-box view on a providing service component ........................................ 58
5.2 The focus of service usage information ........................................ 61
5.3 The focus of service function information ........................................ 63
5.4 The service specification framework ........................................ 72

6.1 Salmela’s view on ‘quality’ [47] ........................................ 77
6.2 Scope of service specification regarding ‘quality’ ........................................ 79
6.3 An adapted version of Salmela’s view on ‘quality’ ........................................ 81
6.4 The Extended ISO model ........................................ 82
6.5 The service specification framework ........................................ 88

7.1 The Generic Service Specification Framework ........................................ 94
7.2 Service contract information ........................................ 95
7.3 Service function information ........................................ 97

8.1 Aspects of service specification in the organization ........................................ 104
8.2 Service specification framework in the organization ........................................ 106
List of Tables

4.1 Types of services ......................................................... 54
5.1 Types of function conditions ........................................... 66
6.1 Quality aspects in ICT and business services ....................... 84
6.2 Determinants of business quality and quality aspects ............. 85
6.3 Areas of concern and quality aspects ................................ 86
Part I

Preliminaries
Chapter 1

Introduction

1.1 Background

Companies operate in environments which are characterized by changes. Mergers, acquisitions, and the use of new technologies are examples of drivers for change in business environments [19]. These drivers enable more opportunities to meet customer requirements. A company is almost obliged to cope with this situation due to competition with other companies. This causes changing business requirements in order to meet long-term business goals such as stable growth of the company. Companies have to deal with changes. To meet constantly changing business requirements Information and Communication Technology (ICT) systems must be in constant evolution [19].

Current trends in business environments require companies to change business requirements more frequently than ever, because customers are more demanding and require better and cheaper products. Alongside this development, legislations such as the Sarbanes-Oxley Act [52] oblige companies to comply with stern demands on quality and reliability of the business administration [62]. In addition, the Internet and mobile phones create the opportunity to offer products and services via new channels. Under the force of these circumstances, companies should renew their products and services more frequently with the result that more new ICT systems are introduced in the company and legacy ICT systems should adapt more often. However, it has been proven that changing the ICT infrastructure is difficult, because ICT systems in an organization strongly cohere [62]. This coherence resulted from the fact that ICT systems are often specifically tailored to the needs of an organization [62]. ICT is therefore often regarded as an obstacle to meet changes in business [61]. It is clear that there is a need for making the ICT systems more flexible with minimal changes in the ICT infrastructure.

1.2 Service Oriented Architecture

Service-oriented architecture (SOA) is a modern paradigm, which has been introduced in order to respond to fast changing business requirements and to cope with ICT issues discussed in the previous section. ICT systems provide their functionalities as services, which
1.3 Service

In the SOA domain, service is defined as an exposed piece of functionality, which is self-contained, autonomous, platform independent and can dynamically be located, invoked and can be distributed over a network. These services are accessible to other ICT systems by offering them through standard interfaces. In that way, the implementation of services is no point of interest to the users of the service. By encapsulating legacy systems in a service layer for SOA, companies can put new life into their legacy systems instead of introducing new ICT systems into the company.

Companies can realize automated support of their business processes by means of SOA. A business process is an ordered collection of business activities. Such an activity can be supported using ICT systems, which can be encapsulated in a service layer as stated above. Hence, automated support of business processes comprises a controlled sequence of service calls. Such a sequence is called an orchestration [54]. A change in the business process can now simply be heard by changing the orchestration, calling new services or by removing certain service calls from the orchestration. In that way, ICT systems are better able to adapt to fast changing business requirements.

SOA involves three particular parties. A service provider publishes his service to the service registry. A service registry contains a repository of services published. It can be regarded as the well-known Yellow Pages containing descriptions of distinct services, which are known as service specifications. A service consumer can browse through the service registry in order to find the service that matches his criteria. When the service consumer locates the service desired in the registry, it can access and use this service. If the service consumer and his data comply with the rules of the service provider, a binding can be established. The service consumer can pass his data to the service in order to be processed. The service returns the value resulted to the service consumer.

At the moment, SOA is clearly gaining popularity in business in response to changing business requirements. In 2007, SOA was used, to some extent, in more than 50 percent of large, new applications and business processes designed in 2007 [3]. Gartner even expects that by 2010 more than 80 percent of the companies will use some aspect of SOA [3].
(re-)combined [41]. ‘Self-contained’ means that a service maintains its own state. It does not depend on the state of other services. ‘Autonomous’ refers to the fact that the operation of the service is perceived as opaque by external components [41]. ‘Platform independent’ indicates that a service consumer from any communication device using any computational platform, operating system and any programming language can use the service [41]. The service can communicate with other services through their interfaces without concerning how these services are implemented. It does not care whether these services are realized using new ICT systems or legacy systems adapted. What implies that a service does not concern whether a service is local or remote, is the fact that a service can dynamically be located, invoked and (re-)combined. In other words, services are loosely coupled. The interfaces of these services are invocable and implementation details are hidden to service consumers. They are saved in a service registry as part of the service description.

Although one might suggests that the concept of ‘service’ and SOA are inextricably bound up with each other, this is not the case. ‘Service’ is a much broader concept as the following example shows:

A customer is interested in credit loan. The customer visits the bank in order to obtain more information. The bank employee at the service desk clarifies the different types of credit loans that are offered by the bank.

This example shows that services have not to be tied to any kind of ICT system. It is therefore important to have a general view on the concept of ‘service’ without considering its implementation and its application domain. This has already been stated more than twenty years ago by Vissers and Logrippo [64]. Over the course of time, several researchers have proposed their view on ‘service’. An interesting definition of service has been given by Elfatatry, who stated that service is any act or performance that one party can offer to another that is essentially intangible and does not result in the ownership of anything, where its production may or may not be tied to a physical product [19]. In our opinion, this definition gives a clear general view on service.

### 1.4 Service Specification

Service specification is an essential part of the realization of SOA. Service specifications contain a complete, unequivocal and precise description of the external view of a service component and describe the conditions under which the service is provided [58]. Service providers use service specifications for advertising the service capabilities in the service registry. Service consumers use service specifications in order to find a certain service, to judge whether they are interested in a certain service and to know how to use a certain service. This denotes that service specification consists of two parts; a brief description of what the service offers and information about how to use the service [67]. We call the first part *service function information* and the second part *service usage information*. Besides this important role of service specifications, they are even more a sine qua non for SOA realization since they might be the only available support for a service consumers for judging the usefulness of the service [58]. This has been recognized over the course of time and several
specification standards have been developed in order to specify each conceivable service aspect. However, the morbid growth of specification standards made service specification even more complicated. It is not realistic to expect that service providers will support all of the specification standards existed [66].

1.5 Current Issues on Service Specification

An important issue on service specification is the lack of support to business people in service specification [54]. Business people have to be involved in service specification, because they need to validate whether a certain service really fulfil their needs for business process support [54]. Let us consider, for instance, the use of Web services, which is the most popular type of services available today in SOA [41]. The Web Service Description Language (WSDL) [15] only enables to specify technical service information of Web services. Service information with respect to the business can be exposed using Universal Description, Discovery and Integration (UDDI) [1]. However, UDDI is a very technical oriented way for specifying services [54].

Current approaches on service specification are based on developments concerning SOA in practice. A good example of such a specification approach is the Faceted Specification approach from the Service Centric System Engineering (SeCSE) project [60]. This specification approach distinguishes several facet types representing service properties to be specified on the basis of existing specification techniques in SOA. Each facet type has been developed as a result of the needs of the project partners with respect to service properties to be specified. Issues on determining the service aspects on SOA developments in practice concern the fact that there is often no clear rationale behind the meaning of service aspects and the decision of including these service aspects in service specification. This signifies the importance of using a well-founded theory for service specification.

1.6 Research Goal

The previous sections imply the need of a decent framework for service specification. What is meant by a framework for service specification, is an outline of the service aspects to be specified for service selection and service usage. The framework has to encourage effective and efficient specification of services which is understandable to all types of stakeholders. What we mean by a 'stakeholder' is any individual or group of individuals that make use of service specifications in his or in their work. It goes without saying that service consumers can be counted as one of the stakeholders, since they are the ones that use services. The framework has to be applicable to all services and contribute to the realization of SOA in business. Contrary to service specification frameworks existed, we approach service specification from a scientific point of view using a robust theoretical foundation. This theoretical foundation is obtained from enterprise ontology. In all, the research goal can be formulated as follow:
Develop a service specification framework using a robust theoretical foundation which outlines the service properties that should be part of every service specification in order to support service selection and usage by stakeholders.

In order to achieve the research goal, a number of research questions can be discerned:

1. How can enterprise ontology provide ground for specifying services in SOA?
   a) What is enterprise ontology?
   b) How is enterprise ontology related to the concept of ‘service’?
   c) How is enterprise ontology related to the concept of ‘service specification’?

2. How does the service specification framework support stakeholders in service selection and service usage?
   a) What types of stakeholders make use of service specifications according to enterprise ontology?
   b) What service aspects are included in the service specification framework using enterprise ontology?
   c) How does the service specification framework actually support stakeholders in practice?

1.7 Research Approach

In order to provide a solid ground for a successful achievement of the research goal, the research had been divided into phases. We believed that a phased approach contributed to high-quality research results, since each phase could provide clear and concrete results, which could be examined and reviewed before the start of the next phase. This ensured useful and quality information as input for each next phase. Figure 1.2 gives a schematic representation of the phases.

The formulation of the research goal, as indicated by phase A, was the starting point of the research. It sets out the directions for the research process. Therefore, we took care over the formulation of the research goal.

Phase B comprised the study of enterprise ontology. It was of great importance to have a clear and full understanding of enterprise ontology. A successful achievement of the research goal stood or fell with the degree of understanding enterprise ontology. Phase B could therefore be regarded as the fundamental part of the research.

Having a clear and full understanding of enterprise ontology laid the foundation of further research on the relation between enterprise ontology and service and service specification in phase C and D. The aim of these phases was to have a clear view on the concept of ‘service’ and the role of service specification according to enterprise ontology. By finishing these phases, we gained a clear picture of the actual foundation enterprise ontology could provide for the development of the framework. The results of the phases indicated
1.7 Research Approach

Figure 1.2: Research phases
whether using enterprise ontology as a theoretical foundation was actually the right choice. Therefore we had to take further research on other theories and methodologies into account during the research as presented by phase E.

The development of the service specification framework in phase F should actually not be regarded as a complete separate phase from the previous phases. The results of phase C, D, and E provided valuable insights into the design of the framework. They all constituted the framework as the result of the research.

In order to verify the usefulness of the framework developed, the framework had to be applied in practice. In that way, the strengths and shortcomings of the framework could be exposed. The results of phase G indicated to what extent the framework developed actually supports service consumers in service selection and service usage.

Phase H concerned writing the thesis report. The findings of each phase were incorporated into the report in order to have a detailed insight into the process of coming to the framework developed.

1.8 Report Structure

The report consists of four parts. Here below we give a brief description of the contents of each chapter.

Part I comprises the preliminaries that give an overall picture of the topic of discussion. It introduces all aspects that form the context of the project. Chapter 2 gives an extensive overview of the theory that lays the foundation of the service specification framework to be developed. Such an extensive overview is needed in order to have a clear understanding of the theory and each of its elements. Next to this, it is important to have a clear view on current notions on and approaches to service specification. Therefore, we briefly discuss a number of service specification frameworks that are currently well-known in the SOA domain in Chapter 3.

The theme of Part II is the identification of service aspects that should be part of every service specification. Chapter 4 studies in great detail how the concept of 'service' can be described in terms of enterprise ontology. After this has been done, we are able to identify the role of service specification and its content on the basis of enterprise ontology in Chapter 5. As these chapters will make clear, enterprise ontology only concentrates on the functional aspects of service. Therefore, we propose an approach for identifying service quality aspects in Chapter 6.

Part III concerns the presentation and verification of the service specification framework. Chapter 7 gives a detailed overview of the service specification framework developed. In Chapter 8, we focus on the case study, which was conducted at a large Dutch governmental organization in order to verify the usefulness of the framework in practice.

Part IV constitutes the closing part of the thesis. In Chapter 9, we enumerate the main contributions of the research to a service specification framework, which is based on a well-founded theory. Chapter 10 deals with the conclusion of the research wherein the research questions and the research goal given in the introduction are reviewed. Chapter 11 provides a list of useful suggestions for future work.
Chapter 2

Enterprise Ontology

This chapter gives an overview of the concept of ‘enterprise ontology’ and the theory that underlies ‘enterprise ontology’. Before we discuss the theory in more detail, we first explain why we selected the concept of ‘enterprise ontology’ and its theory as the basis for the framework in Section 2.1. The theory of enterprise ontology comprises several axioms and what is termed the organization theorem. We will discuss the relevant elements of the theory in the next sections. Section 2.2 deals with the operation axiom. In Section 2.3, we look at the transaction axiom. Section 2.4 focuses on the distinction axiom. The organization theorem is discussed in Section 2.5

2.1 Why Enterprise Ontology?

In the introduction, we explained the need of a theory on which the specification framework to be develop can rely. It should aid us in solving the current issues on service specification. This implies that the theory should be well-founded, which may be proved from the acceptance of this theory in science and the practical experience of using this theory. In addition, the theory should provide the opportunity to incorporate service information, which supports business people, in service specification. Because of this, it is interesting to mention the relation between services and human activities in a company. As can be derived from Section 1.2, services are orchestrated in a particular order to support a certain business process. Since a business process consists of an ordered collection of human activities, it can be stated that a service supports a particular activity in a company. Clearly, services and human activities are closely linked to each other. This known fact might aid in finding a way for specifying services comprehensibly to business people.

*Enterprise ontology* deals, simply said, with the essence of an enterprise. It concerns the construction and operation of an enterprise independent from its implementation. In that way, the enterprise’s construction and operation can be described in a coherent, comprehensive, consistent and concise manner [18]. The essence of the operation of an enterprise is described in terms of actor roles and their activities without mentioning how they are fulfilled. The realization and implementation of the enterprise consist of the fulfillment of these actor roles. Actor roles are fulfilled by human beings and these human beings perform
their activities to each other within the enterprise or to the enterprise’s environment. ICT systems are regarded as a supporting tool for human beings in performing their activities.

Enterprise ontology gives the opportunity to make a clear distinction between the activities in an enterprise and the realization and implementation of these activities using human beings or ICT systems. It enables to find out the relation between human beings and ICT systems whereby we can come upon a way for exposing service aspects at a non-technical oriented way.

Enterprise ontology is a theory, which is widely accepted in the scientific world. Many articles have been published regarding this theory in journals and scientific papers. Some of these articles are listed at the website of the International CIAO! Network (www.ciao.tudelft.nl) and at the website of the Design and Engineering Methodology for Organizations (DEMO) Knowledge Center (www.demo.nl). Besides the acceptance in science, enterprise ontology relies on fifteen years of practical experience using the DEMO methodology [18]. In all, enterprise ontology is the well-founded theory needed for the development of the specification framework.

2.2 The Operation Axiom

The operation axiom states that the operation of an enterprise comprises of activities and actor roles, which are atomic elements of authority and responsibility, fulfilled by human beings [18]. It signifies the performance of two kinds of activities by actor roles. Production acts (P-acts) are acts that deliver goods or services from the enterprise to its environment [18]. The delivery of goods is realized by means of material acts such as manufacturing and transporting, while the delivery of services are realized by means of immaterial acts such as deciding and judging. The performance of P-acts results in what are termed production facts (P-facts) [18]. Let us clarify this in the context of a pizzeria. A P-fact would be, for instance, ‘the costs of pizza order O have been paid’. The variable O refers to an instance of pizza order. The creation of such a P-fact by a corresponding P-act is known as a transition in the production world (P-world) [18]. The state of a P-world is formed by the set of P-facts that have been created. This concept is backed up by the diagram given in Figure 2.1.

![Figure 2.1: Representation of the operation axiom [18]](image-url)

Coordination acts (C-acts) are performed by one actor and directed to another actor [18]. They coordinate the execution of production acts using a set of intentions of which request, promise, statement and acceptance are the basic intentions in enterprise ontology [18]. A C-act is defined using one of those intentions followed by the P-fact such as for example, the
being requested of the P-fact 'the costs of pizza order O have been paid'. As with P-acts, the performance of C-acts results in facts, which are known as \textit{coordination facts} (C-facts) [18]. The creation of C-facts by C-acts is called a transition in the \textit{coordination world} (C-world) [18].

\subsection{2.3 The Transaction Axiom}

The \textit{transaction axiom} states how actor roles and the different acts, defined by the operation axiom, are related together [18]. C-acts are performed in a certain pattern in order to achieve a particular result, which is a P-fact. This pattern of C-acts is called a \textit{transaction} [18].

\begin{figure}[h]
\begin{center}
\includegraphics[width=0.5\textwidth]{figure2_2.png}
\end{center}
\caption{The basic pattern of a transaction [18]}
\end{figure}

Consider Figure 2.2 which illustrates a \textit{basic pattern} of a transaction. A transaction involves two actor roles, the initiator and the executor. Let us also again consider the P-fact of the previous section in order to clarify the actor roles and acts involved in a transaction. The P-fact 'the costs of pizza order O have been paid' is the result of the transaction. The initiator is the pizzeria who wants the customer to pay his pizza. This implies that the customer is the executor of the transaction, since he is the one who can create the P-fact concerned. The transaction is initiated by a request of the pizzeria for paying the pizza order. By acknowledging the request, the customer promises to pay the pizza order (i.e. the promise that the P-fact is created). The transaction part discussed up to now is known as the \textit{order} phase (O-phase) of the transaction. Now we enter into the \textit{execution} phase (E-phase) of the transaction. After the customer has promised to pay for the pizza order, the customer actually pays. This holds the performance of a P-act resulting in the P-fact 'the costs of pizza order O have been paid'. Then, the customer states that the costs of the pizza order have been paid, which means that the customer has handed over the amount of money for the payment to the pizzeria. The pizzeria accepts the fact that the costs of the pizza order have been paid and completes the transaction. This is known as the \textit{result} phase (R-phase).

The basic pattern of a transaction suggests that the initiator and the executor always agree on each other during a transaction. For instance, in the basic pattern the state act is
always followed by the accept act. However, one may note that in practice this is actually inconceivable. Further, it is also a utopia to assume that the executor will always promise as respond to a request from the initiator. Therefore, we should extend the basic pattern of a transaction in order to cope with those scenarios. This is called the *standard pattern* of a transaction [18].

![Figure 2.3: The standard pattern of a transaction [18]](image)

Disagreements in transactions are indicated with the *decline* and *reject* act and result in what are termed *discussion states* [18]. We call states resulted from acts included in the basic pattern *normal states* [18]. A discussion state indicates that the initiator and the executor of the transaction are in discussion in what way the transaction can successfully be completed. This discussion may eventually lead to another attempt to complete the transaction successfully. On the other hand, the transaction might be quitted by the initiator or stopped by the executor.

Recall the pizzeria example. As can be seen in Figure 2.3, the customer may decline the request for paying his pizza instead of promising. He could decline the request, because in his opinion the pizza as part of the order was delivered too cold. This implies a discussion state in the transaction. The pizzeria could try to convince the customer that the pizza is at its tastiest at that temperature. If the customer is convinced, the customer will finally promise to pay the pizza order, which will lead to the actual payment. Else, the pizzeria could take back his request for paying the pizza order i.e. he quits the transaction. The pizzeria might reject the payment that has been done by the customer, which is presented by the state act, instead of accepting it. One might think, for instance, of the fact that the pizzeria thinks that the customer has used false money for paying the pizza order. In that case, the customer and the pizzeria are in a discussion state. The customer can try to convince the pizzeria of
the fact that he did not pay using false money. If he succeeds, the pizzeria and the customer agree with each other and the pizzeria actually accepts the payment. If the pizzeria cannot be convinced, the transaction is stopped.

It is quite common in practice that either the initiator or the executor of a transaction wants to revoke an act, which urges to introduce the possibility to cancel any C-act in the basic pattern at any time [18]. This is known as the cancellation pattern [18]. As a respond to the performance of a cancel act, an allow or a refuse act can be performed indicating whether the cancellation is actually realized. We discuss the cancellation patterns of a request, promise, state and acceptance here below using the pizzeria example.

![Figure 2.4: Cancellation pattern of a request][18]

The pizzeria could recall the request for paying the pizza, since the pizzeria suddenly remembers that pizzas are free of charge at that moment. The performance of the cancel act implies that the pizzeria and the customer end up in the discussion state ‘cancelled’. The customer can allow the cancellation by performing an allow act. This results in the performance of the quit act whereby the transaction is ended. However, the customer can also refuse the cancellation, which would be very unlikely in this situation. In that case, the state ‘refused’ is the terminal state for the cancellation, which means that the C-fact ‘requested’ remains the case [18]. From this state, the customer can decline the request or promise to pay for the pizza.

![Figure 2.5: Cancellation pattern of a promise][18]

Let us now consider the cancellation of a promise. The customer can find out that he has no money in his wallet for paying the pizza order whereby he has to cancel the promise to pay for the pizza order. By allowing the cancellation, the decline act is performed by the customer, which holds that the pizzeria and the customer end up in the discussion state.
2.4 The Distinction Axiom

The distinction axiom states that three distinct human abilities play a role in the performance of C-acts and P-acts [18]. This is illustrated in Figure 2.8.

On the coordination side, the forma ability deals with the form aspects of communication and information [18]. Information should be expressed in a particular language or code scheme that both the initiator and the executor of a transaction understand. In other words,
2.5 The Organization Theorem

The organization theorem states that the organization of an enterprise is a heterogeneous system that is constituted as the layered integration of three homogeneous systems: the B-organization (from Business), the I-organization (from Intellect), and the D-organization (from Document) [18]. These homogeneous systems are also known as the aspect organizations of the enterprise. In order to clarify the organization theorem, the concept of ‘system’...
and the aspect organizations are explained in more detail first.

Two different notions on ‘system’ can be distinguished, which enable to view a system from two different perspectives. The teleological notion of a system approaches a system from the functional perspective. It defines the system as a black-box model that concerns the external behavior and functions of the system. The ontological notion of system views a system from a constructional perspective. This is also known as the white-box model of the system. According to this notion of system, something is a system if it has the following properties [18]:

1. Composition: a set of elements of a certain category
2. Environment: a set of elements of the same category, where the composition and environment are disjoint
3. Production: the elements in the composition produce things
4. Structure: a set of influence bonds among the elements in the composition, and between them and the elements in the environment

As mentioned above, each aspect organization is considered to be a system on its own. This means that it deals with elements within its composition and elements from its environment as well. As the enterprise itself, each of the three aspect organizations is a social system, which states that the organization is composed of human beings as actors performing acts in order to produce things to the environment. The aspect organizations differ at the type of things produced.
The B-organization concerns the essence of the enterprise [18]. It focuses on the products and services that the enterprise delivers to the environment. This production of the B-organization is ontological, which means that it deals with the core business of the enterprise [18]. Actors and acts within the B-organization directly contribute to the enterprise’s goals and functions. One might think, for instance, of sales persons as actors in the B-organization representing the enterprise. The I-organization embraces the content aspects of information and knowledge in the enterprise [18]. Actors bring changes to information and knowledge by acts concerned such as reasoning, reproducing and computing. In that way, new information and knowledge can be created in the enterprise. A certain competence is needed from the actors for performing that kind of acts and this clarifies the focus on ‘intellect’ in this organization. The type of production by the I-organization is known as infological. Accountants are typical actors in the I-organization producing infological things. The D-organization concerns the documentation of information in the enterprise and deals with the form of information [18]. This implies acts such as storing, transmitting or destroying information that are performed in the D-organization. The production of the D-organization is therefore known as datalogical. Actors in the D-organization are for instance archivists.

The layered integration of these aspect organizations is based on the distinction of the functional and constructional perspectives on the organizations. Each underlying layer supports construction of the layer above by means of its function. The construction of an organization consists of the elements in the organization and the influence bonds among these elements. As stated above, these elements are human beings working on a certain task on their own or together. In that way, these elements realize the functions of the organization concerned. Let us consider, for instance, the D-organization supporting the I-organization in order to clarify this. As can be seen in Figure 2.11, the construction of the I-organization is supported by the function of the D-organization. An accountant, which is an actor in the I-organization, is in need of the documents containing financial figures of 2006 in order to compare them with the financial figures of 2007. An archivist can provide the accountant...
2.5 The Organization Theorem

The organization theorem denotes the use of ICT-systems in the enterprise as the implementation of the organization of an enterprise. Consider Figure 2.12 which illustrates the different organizations and ICT systems. ICT systems are distinguished between hardware and software. Software is used in order to support the different aspect organizations in the enterprise or to take over the acts performed by the actors in these aspect organizations [18]. Software can therefore be classified in the same way as organizations, which implies the document containing the financial figures of 2006. This is a typical function realized by the D-organization.

Figure 2.11: The layered integration of an organization [18]

Figure 2.12: Organizations and ICT systems [18]
the following types of software: B-application, I-application and D-application [18]. Hardware is considered to be the lowest layer supporting the D-application. As D-organizations, D-applications concern the storage, transmission or destruction of information [18]. One might think of email systems which enable to transmit digital documents. I-applications use existing information and knowledge in order to obtain new information and knowledge by computation or calculation [18]. Accounting systems are examples of I-applications. B-applications support decision-making within the enterprise [18]. This type of applications is for instance workflow systems.

As Figure 2.12 suggests, each underlying ICT system supports the ICT system above. An I-application can run using several D-applications. A B-application is supported by means of several I-applications.
Chapter 3

Related Work

Service specification is still in its infancy in SOA. In order to give a picture of the approaches on service specification that currently exist, this chapter discusses the UDDI specification framework [9], business component specification [4] and the faceted specification approach [60].

3.1 The UDDI Specification Framework

*Universal Description Discovery Integration* (UDDI) is a global, platform-independent registry which enables service providers to publish their services and service consumers to discover services [9]. In order to realize this, UDDI provides a standardized specification framework for describing services, which defines three types of service information:

- White pages
  White pages describe general information of the company that provides the service such as name, description and contact information.

- Yellow pages
  Yellow pages describe the application field of the service in terms of standard taxonomies. In that way, services can be located by industry, category, or geographical location using standard industrial categorizations.

- Green pages
  Green pages contain technical information about the service and include references to the WSDL document of the service

These types of service information are stored and exposed using the Extensible Markup Language (XML) [13] in the form of four data types which is shown in Figure 3.1.

A *businessEntity* structure represents the business as service provider and holds information with respect to the business such as name, description and contact information. As Figure 3.1 illustrates, a businessEntity can be linked to multiple *businessService* structures, which indicates that a service provider can offer more than one logical grouping of services. A businessService structure exposes the name, description and classification information
of a particular logical grouping of services without mentioning any technical information about these services [1]. It can hold several bindingTemplate structures representing a single service. A bindingTemplate structure describes such a single service in more detail and includes technical information how to bind and interact with the service. Each bindingTemplate structure refers to one or more tModels. A tModel structure is used for defining the technical specification of a service and can be based on information defined in the WSDL document of the service.

3.2 Business Component Specification

Business components are reusable, self-contained and marketable software artifacts that provide services through well-defined interfaces to the business domain [4]. As mentioned in Section 1.3, a service in SOA offers its functionalities through a standard interface as business components do. Since the implementation details of a service are irrelevant in SOA, this specification framework is also suitable for specifying services in SOA [54]. Business component specification denotes a number of specification levels that distinguish the different viewpoints on service from ICT and business. Each level includes a software contract, which is defined as obligation to which a service provider and a service consumer agree [16]. Here below, an overview is given of the different specification levels as described in [4].

- **Interface level**
  The interface level defines service arrangements in terms of public attributes, data types, variables, constants and exceptions. It exposes these elements offered, but also mentions the services that the service component needs in order to function properly.

- **Behavioural level**
  The behavioural level defines how services behave in terms of invariants and pre- and postconditions. These constraints are applied to the service elements defined in the interface level. The service provider and the service consumer must comply with these constraints in order to execute the services.
• Coordination level
  The coordination level indicates the order wherein services are able to occur. Statements on the coordination level can refer in the same manner to services offered by the service component concerned as they can refer to services required from other components. This holds that services within the same service component as well as services of different service components are mentioned in the specification.

• Quality level
  The quality level concerns non-functional characteristics of the service component. Examples of non-functional characteristics include response time, reliability and availability.

• Terminology level
  The terminology level comprises the definitions and terms that are used in the distinct levels of the service specification. It can be viewed as a glossary of every concept that is needed for understanding the service specification [23]. Besides the defi-
initions, relationships between entities defined are described, for example sequence relationships in which the predecessor and/or successor of entities are stated. The terminology level enables shared understanding with regard to definitions and terms used. Multiple interpretations of definitions and terms are avoided in this way.

- **Task level**
  The task level indicates which business tasks are supported by the service component. Business tasks can be decomposed into subtasks. An index lists the business tasks together with their subtasks. What is suggested to be used as a specification method, is a language which is clear to end-users and close to natural language.

- **Marketing level**
  The marketing level concerns business, organizational and technical characteristics of the service component which is of interest from a business and organizational perspective. Name, versioning, domain and contact person are examples of such characteristics. Service information at the marketing level can be seen as minimal information for potential service consumers in order to decide if they are interested to use the service.

### 3.3 The Faceted Specification Approach

The Service Centric System Engineering (SecSE) project developed a service specification framework in order to bring together, and bring order to, specifications that are expressed in different schemes or languages that address similar properties of a service [66]. According to this project group, there is currently a large variety of specification techniques available, which are often competing. In addition, it is not realistic to expect that service providers will support all of the technologies proposed [66]. The framework developed is known as the faceted specification approach. Service specifications consist of several facets each referring to one or more service properties. More specific, a facet is a projection over one or more service properties that provides a partial description of a service and that covers particular aspects of a service [60]. Figure 3.3 illustrates the specification structure using facets.

As can be seen, a service specification contains the service name, its unique ID and one or more facets. Facets are, in principle, optional to be included in service specification. In that way, service providers are not forced to describe certain properties of their service, which are probably even not of interest for that particular service. For instance, if there is no test data available for testing the service, the service provider can exclude this facet from the service specification. However, service specifications should at least contain the facets Description, Signature and Management in order to provide sufficient valuable service information. Alongside these facets, there are currently seven other facet types for describing service properties. Each facet type has been developed as a result of the needs of the project partners with respect to service properties to be specified. Yet, one might expect that new specification techniques will evolve over the course of time. Therefore, it is quite possible that new facet types are developed in the future. The current facet types are listed below:
Related Work

3.3 The Faceted Specification Approach

- **Description**
  The Description facet provides a high-level description of the service. It covers the most essential properties of the service and is used to support early service discovery. In other words, it enables service consumers to decide if they are interested to use the service.

- **Signature**
  The Signature facet provides a detailed description of the service operations and binding information. It is comparable to the service information that is available in a WSDL document.

- **Commerce**
  The Commerce facet concerns commercial information related to the service such as cost of using the service and Service Level Agreements (SLA’s)

- **Testing**
  The Testing facet provides the information needed for testing the service. Besides test data, it also describes how service consumers can test the service.
3.3 The Faceted Specification Approach

- **QoS**
  The QoS facet concerns the Quality-of-Service (QoS) of the service. It comprises the non-functional properties of the service.

- **Operational Semantics**
  The Operational Semantics facet provides additional information with respect to information defined in the Signature facet. It enables service consumers to understand and evaluate the information defined in the Signature facet using semantics.

- **Exception**
  The Exception facet deals with the failure behaviour of a service in terms of pre- and postconditions.

- **Management**
  The Management facet covers management information related to the service needed for service administration. This includes information such as service versioning.

- **P2P**
  The P2P facet deals with services that have been developed as P2P services. It provides a service description and states how the P2P service can be discovered and invoked.

- **Business Models**
  The Business Models facet captures information relating to the service provider’s business models that may be of interested to the service consumer.
Part II

Towards a Service Specification Framework
Chapter 4

Service in Enterprise Ontology

The previous chapters discussed the concept of ‘service’, service specification and the theory of enterprise ontology in detail. As part of enterprise ontology, the organization theorem makes a clear distinction between the realization of the organization of an enterprise and the implementation of the organization of an enterprise. If we consider Section 1.3 again, it conveys the impression that the same kind of distinction can be made with services. In this chapter, we further study the relations between the concept of ‘service’ and enterprise ontology in detail. In Section 4.1 we discuss to what extent the concept of ‘service’ can be defined on the basis of enterprise ontology. Section 4.2 concentrates on the type of services, which are based on the realization of the organization. Section 4.3 deals with the type of services that adheres to the implementation of the organization. In Section 4.4 we discuss the notion of a composite service and a service component in SOA related to enterprise ontology. Section 4.5 studies the notion of ‘service provider’ and ‘service consumer’ in more detail. Section 4.6 concludes by giving a brief review of the findings in this chapter.

4.1 The Concept of ‘Service’

In the introduction, we exposed the general view on a service using the service definition of Elfatatry. Elfatatry stated that a service is any act or performance that one party can offer to another that is essentially intangible and does not result in the ownership of anything, where its production may or may not be tied to a physical product [19]. Note that ‘act’ as defined by Elfatatry does not refer to ‘act’ in enterprise ontology. Obviously, a service concerns two parties; one party that is capable and authorized to offer a particular act and another party that makes use of that act. These parties can be regarded as the service provider and the service consumer, respectively. It further seems that the offer of an act, as defined by Elfatatry, from one party to another might be regarded as a transaction in enterprise ontology (see Section 2.3). The executor of a transaction is authorized to produce something on request of the initiator of the transaction.

Evidently, a service and a transaction have some aspects in common. Let us therefore consider a transaction in enterprise ontology in more detail. Section 2.3 clarified the basic, standard and cancellation patterns of a transaction and the different acts the patterns are
4.1 The Concept of ‘Service’

Service in Enterprise Ontology

In order to uncover a possible relation between a service and a transaction in a clear manner, we first concentrate on the basic and standard pattern of a transaction. Afterwards, we deal with the cancellation patterns.

4.1.1 Service and the Basic and Standard Pattern of a Transaction

The request act denotes the initiation of a transaction. If we compare this to ‘service’, we can identify this as a service request from the service consumer. The request act is followed by the promise act from the executor of the transaction. Regarding a service, this holds that the service provider agrees with the request and is able to execute the service. However, there are a number of factors that could prevent the service provider from performing the promise act. For example, the service provider may negatively respond to the request if the service consumer has no permission to execute the service concerned. Another factor is, for instance, when the input data provided by the service consumer does not comply with the prerequisites of service execution. As can be seen, a negative respond of the service provider refers to the decline act and a positive respond concerns the promise act. A negative respond can result in quitting the conversation between the service provider and the service consumer or can even still result in a positive respond if the service consumer can convince the service provider to agree with the request. In other words, the service consumer performs a quit act or the service provider still performs a promise act. The P-act comprises the production during a transaction. In terms of services, this refers to the actual service execution. The state act holds that the production has been accomplished and is presented to the initiator of the transaction. If we relate this to service, this means that the result of the service execution is available to the service consumer. The accept act is clearly an act performed by the initiator who can accept the P-fact produced by the executor. In that case, the executor and the initiator agree on the P-fact resulted from the P-act. As a consequence, the P-fact comes into existence and the transaction is completed. Regarding services, this holds that a service consumer may accept the result from the service execution. As a result, the conversation between the service consumer and the service provider is successfully ended. The reject act makes it mandatory for the executor to reconsider the P-fact produced. Consequently, the executor performs the state act again so that the initiator and executor agree with each other and the initiator finally performs the accept act. On the other hand, the executor can end the transaction by performing the stop act. Regarding a service, the performance of a reject act indicates that another service call is performed by the service consumer next to the service request. As a consequence, the service provider can convince the service consumer to finally accept the service result. In terms of enterprise ontology, an accept act is finally performed whereby the P-fact comes into existence and the transaction is completed. Regarding services, the conversation between the service consumer and the service is successfully ended. If the initiator and the executor cannot find any agreements regarding the P-fact, the executor will perform the stop act, which means that the P-fact does not come into existence. In terms of services, this holds that the service will undo the service result and will end the conversation.

The above uncovered a clear relation between a transaction and a service. The main focus of a transaction is, however, the communication between two actors on the basis of
4.1 The Concept of ‘Service’

C-acts in order to perform a P-act. It is yet questionable whether service also includes all of these C-acts. A request act is part of a transaction, however, requesting a service is not part of a service itself. A service request should rather be referred to the invocation of the service. On the other hand, a promise act and a decline act can be regarded as part of a service, since they are acts of the service itself as respond to the service request. A quit act as respond to a decline act should not be considered to be a part of a service, since it is a reaction of the service consumer on the declination. The P-act of a transaction clearly refers to the actual service execution. The state act further means that the service provider states that the result of the execution is known and available to the service consumer, which can also be regarded as a part of a service. The acceptance or rejection of this result is not part of a service. Acceptance and rejection should be regarded as judgements on the service result made by the service consumer. Regarding enterprise ontology, the transaction is completed by performing an accept act. This can be referred to the end of the conversation between the service provider and the service consumer, which holds that they are disconnected. The performance of a reject act results in another performance of the state act or in the performance of a stop act. As stated above, a reject act is another service call performed by the service consumer. Although the general view on a service does not state whether multiple service calls are allowed, there are several discussions in literature whether multiple service calls concerning one particular service is possible in SOA. The fact that another service call is performed implies that a service is stateful, which however conflicts with the pursuit of stateless services in SOA [22]. Statelessness is a topic that has been much discussed in practice and literature and is regarded as a critical requirement for loose coupling of services [50]. However, multiple service calls require that the service should retain state information in order to cope with following service calls. Srinivasan and Treadwell proposed an approach for dealing with this [50]. According to them, a service must hand over sufficient state information after each intermediate step to the service consumer so that any other service is able to continue the transaction if the transaction is interrupted [50]. If we relate this approach to our research, we can describe this approach as follows:

- At the end of a state act, the service must hand back to the consumer sufficient state information to enable any qualified service to identify and continue the transaction if the consumer performs a reject act.

- The consumer must hand the state information to whichever service it selects when performing a reject act so that the service can process the next step of the transaction.

- The selected service must be able to handle the state information supplied by the consumer, regardless of whether it performed the earlier acts itself.

The above implies that the service should hand over sufficient information about its state so that it is not needed to retain the information itself. In addition, the information should be adequate for the service consumer for performing the reject act as a service call. The above further suggests that we should distinguish between two service types; one that is called by the performance of a request act and one that is called by the performance of a reject act. Recall the service definition of Elfatatry given in Section 1.3. Besides the fact that Elfatatry
defines service as any act or performance that one party can offer to another. Elfatatry also clearly states that 'service' concerns production. Further, the definition of service in SOA describes service as an exposed piece of functionality as stated in Section 1.3. In addition, in [18] can be seen that functionality refers to production. It is therefore not suitable to actually define the service type called by the performance of a reject act as a service, since this service type only comprises of a state act and a stop act, which are coordination acts. Therefore, we rather speak of two service parts instead of two service types. The first service part is called by the performance of a request act and includes the production act, which is the actual service execution. This service part should therefore be regarded as a compulsory part of a service. The second service part should rather be seen as an optional part, since it is not necessary to enable the service consumer to perform a reject act for correct service execution. It should be noted that regarding services in SOA a reject act will basically result in a stop act and not in a state act. As described in Section 2.2, a reject act can result in a state act followed by an accept act when the service consumer is convinced to agree with the service result. An automated service cannot 'convince' the service consumer to finally accept the result. By performing a stop act the result as P-fact does not come into existence and the conversation between the service and the service consumer is ended. The performance of a stop act should therefore be regarded as a rollback, which means that the service result is undone.

4.1.2 Service and the Cancellation Patterns in a Transaction

As described in Section 2.3, a cancellation pattern refers to the revocation of a C-act in the basic pattern of a transaction. Let us consider the cancellation of a request. Besides the promise act, the request act can be followed by a cancel act of the initiator of the transaction. The executor can allow or refuse the cancellation of the request. The performance of an allow act enables the initiator to quit the transaction by performing the quit act. When the executor performs a refuse act, the executor should then decide whether he declines the request or promises to perform the execute act. Regarding services, the performance of a cancel act indicates a service call performed by the service consumer. This implies that there is a third type of service call next to the request and reject act as service call. The allow and refuse act can be regarded as a service at first sight, since they are acts performed by the service provider as respond to the cancel act. However, as explained in Section 4.1.1, a service should always include a production act. Since neither the allow act nor the refuse act are a production act, these acts performed by the service provider cannot be viewed as a service on their own. These acts should rather be regarded as an optional part of the service such as the performance of a stop act as respond to the reject act as service call. As the performance of the reject act as a service call, the service consumer needs sufficient state information for performing a cancel act as service call. When it is possible to perform a cancel act as service call, the service provider can perform the allow act, which results in ending of the conversation between the service consumer and the service provider by the performance of the quit act by the service consumer. When the refuse act is performed, the conversation between the service consumer and the service proceeds. It depends on whether the service provider positively or negatively responds to the service request in what way the
The promise act can be followed by a cancel act as well and gives the initiator the option to allow or refuse the cancellation. The performance of an allow act results in the performance of the decline act by the executor. On the other hand, the performance of a refuse act means that the initiator and the executor remain in the ‘promise’ state from the execution act is performed. In terms of services, the cancellation of a promise implies that the service provider wants to cancel the agreement regarding the service request. It is, however, questionable whether it is possible to generally state that a service provider is able to cancel an act which it performed just before. This would signify that a service provider can reconsider a decision. Regarding services in SOA, this is impossible, since ICT systems are unable to reconsider decisions. In other words, a cancel act cannot be performed by the service provider subsequent to the performance of a promise act. As a consequence, a cancel act performed by a service provider cannot be regarded as part of a service.

The cancellation pattern of a statement comprises the performance of a cancel act by the executor after he performed the state act. As a respond, the initiator can perform the allow act, which means that the P-act is performed again. However, the performance of a refuse act gives the initiator the opportunity to perform a reject or an accept act. If we consider this cancellation pattern regarding services, it can be noted that this pattern holds that the service provider should reconsider his performance of the P-act. As explained in the previous paragraph, it is impossible to state in general that service providers are able to reconsider the acts they performed. Therefore, this performance of the cancel act cannot be viewed as part of a service as well.

The accept act can also be followed by a cancel act of the initiator. Allowing the cancellation enables the initiator to perform the reject act. On the other hand, the refusal of the cancellation by the executor has no further consequences, since the P-fact remains accepted. In terms of services, the cancellation of an acceptance means that the service consumer still wants to reject the service result after the service conversation has been ended. The performance of the cancel act can be regarded as a service call by the service consumer. As with the cancellation of a request, the allow and refuse act performed by the service provider cannot be viewed as a service on their own, since neither of them are production acts. Therefore, we should view the performance of the allow and refuse act as an optional part of a service. The service consumer is only able to perform the cancel act as a service call if the service provider has hand over sufficient state information to the service consumer for performing such a service call. If this is the case, the service provider can perform the allow act, which enables the service consumer to perform a reject act. Yet, the service might perform the refuse act, which means that this service call has no further consequences for neither the service provider nor the service consumer, since the service result remains accepted.

### 4.1.3 The Definition of the Concept of ‘Service’

The above showed that a service should be defined as a part of a transaction rather than a whole transaction. This actually corresponds to the service definition of Elfatatry concerning only the acts performed by the service provider and not the ones performed by the service consumer. A service consumer is able to perform three types of service calls, which
are the request act, the reject act and the cancel act. This would imply that we should deal with three types of services. However, since a service has to consist of a production act, we cannot regard acts as a service on their own that are performed after a reject or cancel act as service call. Therefore, these acts are defined as the optional parts of a service. A service should minimally comprise the acts performed after a request act as service call. This pattern of acts is called the compulsory part of a service. Figure 4.1 illustrates our notion on service on the basis of enterprise ontology. In order to obtain a clear overview, ‘service provider’ and ‘service consumer’ are both represented using two boxes. The blue-colored part represents the compulsory service part that provides a particular functionality to the service consumer. The green-colored part stands for the optional service parts. In addition, we can define service as follows:

A service is a universal pattern of acts performed by one party (party A) for another party (party B). We distinguish between a compulsory part and optional parts of the service. In the compulsory part, the acts are performed as follows:

1. Party A performs a promise act or a decline act as respond to the request act of party B.
2. If party A performed a promise act, it executes the production act, which results in a production fact.
3. After the production act has been executed, party A performs a state act stating that the execution has been performed.

The acts in one optional part are performed as follows:

1. Party A performs a stop act as respond to the reject act of party B.

The acts in the other optional part are performed as follows:

1. Party A performs an allow or a refuse act as respond to the cancel act of party B.

It should be noted that the pattern of acts can be tacitly performed. Regarding services in SOA, the performance of a promise act, for instance, which indicates that the production act will be performed, is basically not exposed. During this research, we further focus on the compulsory part of the service, since the optional parts do not need to be considered in a service. By this, we can concentrate on the main composition of the service in a clear and comprehensible way. We deal with the optional parts if it is needed to explicitly concern a reject act or a cancel act as service call.

The definition of ‘service’ as a part of a transaction offers the opportunity to further detail the service definition. Besides the structure of a transaction as a sequence of acts, enterprise ontology clarifies the different types of transactions that exist. Dietz denotes three composition types of transactions. According to Dietz, every transaction is either enclosed in some other transaction, is a transaction initiated by an actor from the environment of the enterprise, or is a self-activation transaction [18]. Consider Figure 4.2 below. The figure provides a representation of a transaction which is enclosed into another transaction. As
4.1 The Concept of ‘Service’

Figure 4.1: Service according to enterprise ontology
4.1 The Concept of ‘Service’

Service in Enterprise Ontology

can be seen, transaction T1 is initiated by an environmental actor. When the executor of T1 promises, he does two things; he initiates the execution of the P-act and he requests for T2.

It is possible to translate the structure of an enclosed transaction in terms of services. After the service promises the service consumer to create the P-fact, it calls underlying services for creating the production facts needed to accomplish its own P-act. What can be identified here, is the notion of a composite service. In the SOA domain, a composite service is regarded as a service whose implementation calls other services. It involves assembling existing services that access and combine information and functions from possibly multiple service providers [40]. On the basis of enterprise ontology, we can define a composite service as follows:

A composite service is a service from which the creation of the production fact depends on the creation of production facts from underlying services. After the composite service promises the service consumer to create the production fact, it requests the underlying services. If the composite service has accepted the production fact from the underlying services, it creates its own production fact and presents this to the service consumer.

Another type of service structure concerns services that do not call any other underlying services for service execution. They are known as atomic services in the SOA domain. On the basis of enterprise ontology, we define an atomic service as follows:

An atomic service is a service from which the production fact is independently created from other services and is not composed of underlying services.

We will take a closer look on composite services in Section 4.4.

A self-activation transaction is a transaction which is initiated and executed by the same actor role and it is regarded as the generic solution for defining periodic activities in the enterprise [18]. The structure of a self-activation transaction is depicted in Figure 4.3. The performance of a request act in a self-activation transaction results in two acts. First, the promise act is performed as any other regular transaction. Second, a new request is performed, with a requested occurrence time of the P-fact [18]. For instance, if this periodic activity occurs every week, the requested occurrence time of the P-fact is one week from this moment. Let us consider, for instance, a grocery store which weekly controls the stock of the products supplied by the supplier. In that case, one transaction is what is termed the stock control transaction and the other transaction is the product supply transaction. By performing the stock control transaction, the product supply transaction is requested if the stock of products is not sufficient. Furthermore, a new request for the stock control is performed for the next week.

Regarding services, self-activation holds that the service consumer and service provider are regarded as one particular role. The actor who fulfills the role of service consumer also plays the role of service provider. Although there is no specific definition known for this type of services in the SOA domain, it is possible to transform the concept of self-activation in terms of services. If we regard services in SOA, we can consider a stock control service that is invoked every week in order to control the number of products in stock. The service
can invoke another service to order products if there are not enough in stock. In all, we can define a self-activation service as follows:

A **self-activation service** is a service from which the role of service consumer and service provider is fulfilled by the same actor.

### 4.2 Business Service in Enterprise Ontology

Section 2.5 stated that the realization of an organization of an enterprise is the layered integration of the B-organization, I-organization and D-organization. Since each aspect organization is considered to be a social system on its own, each aspect organization consists of actors that enter into and comply with commitments to each other regarding the production acts within the aspect organization concerned [18]. However, as part of the layered integration of an organization of an enterprise, they differ on the things produced, which was mentioned in Section 2.4 and 2.5. The distinction of the aspect organizations in the enterprise implies three distinct actors in the enterprise, B-, I- and D-actors that are the executors of three types of transactions distinguished in the same manner (i.e. B-, I- and D-transactions as explained in Section 2.4). In order to fulfill the role of a B-, I- or D-actor, actors should exert the three basic human abilities; performa, informa and forma [18]. We discussed these
abilities in Section 2.4. In all, a transaction in an aspect organization involves actors, who exert the human abilities, fulfilling the role of initiator or executor performing acts. As Dietz stated in [18], we consider human beings to be the only possible actors, who master the three human abilities. We call a service that is part of a transaction performed by actors in the aspect organizations a **business service**.

If we consider Section 2.4, services as part of D-transactions deal with datalogical productions such as the transmission, storage, copy and destruction of information without considering the content of the information. We call this type of business services **datalogical business services**. Since D-actors are the executors of D-transactions, they can be defined as the providers of datalogical business services. D-actors can provide these services to each other, which implies that D-actors are also the consumers of datalogical business services. In addition, we can also state that I-actors are the consumers of datalogical business services, which can be derived from the layered integration of an organization depicted in Figure 2.11. As was discussed in Section 2.5, the construction of the I-organization makes use of the functionalities of the D-organization. Business services as part of I-transactions concern infological productions such as reasoning, computing and deriving information, which we call **infological business services**. Logically, I-actors are the providers and consumers of infological business services. B-actors should also be identified as consumers, since the function of the I-organization in the form of infological business services supports the construction of the B-organization. Business services as part of B-transactions deal with ontological productions such as making judgements and decisions. We call this type of busi-
ness services ontological business services. The B-actors are the providers and consumers of this type of business services. The environment of the enterprise should also be regarded the consumer of ontological business services.

Dietz pointed out in [18] that ontological transactions are realized by means of infological transactions. The execution of infological transactions results in the reproduction or the derivation of knowledge that is needed to realize ontological transactions [18]. Regarding services, it can be stated that ontological business services are always constituted of at least one infological business service. In other words, before the execution of an ontological business service, the infological business services that constitute the ontological business service should be invoked first. Hence, an ontological business service can be defined as a composite service composed of infological business services. Dietz also stated in [18] that infological transactions are realized by means of datalogical transactions. Datalogical transactions provide for the storage, retrieval and transmission of documents containing the information that is needed in order to realize infological transactions [18]. This implies that infological business services always involve one or more datalogical business services, which indicates that an infological business service is a composite service constituted of datalogical business services.

4.3 ICT Service in Enterprise Ontology

As the distinction between the realization and implementation of the organization of an enterprise, we can speak of services realized by means of human acts and services implemented using Information and Communication Technology (ICT) systems. In the SOA domain, service is considered to be an automated service implemented using ICT systems. As stated in Section 2.5, there are three types of ICT systems each supporting or (partly) taking over the acts performed by the actors in the aspect organizations. In order to research to what extent ICT can support or take over business services in the organization, we have to consider the C-acts and P-acts as part of a service that are implemented or supported using ICT systems. We should therefore take a closer look on the difference between human beings and ICT in performing C-acts and P-acts in B-, I- and D-transactions.

The performance of C-acts is based on the performa, informa and forma exchange between actors. In performing C-acts, an actor has to master the performa, informa and forma ability defined in the distinction axiom in order to enable forma, informa and performa exchange. According to Dietz, ICT systems only control the informa and forma ability and are unable to enter into or comply with commitments contrary to human beings [18]. This holds that ICT systems cannot completely take over C-acts performed by humans. Due to their lack of the performa ability, ICT systems cannot be held responsible for the C-acts they perform. This implies that human beings are always the ones who are held responsible for the C-acts performed by ICT systems [18]. Since C-acts are performed in each transaction type, the fact that ICT systems cannot completely take over C-acts performed by humans holds for all transactions wherein ICT systems are involved. In terms of services, this means that an ICT system can never fully take over the performance of a business service. It only supports the performance of a business service.
Let us now consider the P-acts performed by ICT systems. Datalogical business services comprise datalogical P-acts such as transmission, storage or destruction of information. If we recall the distinction axiom, it can be seen that performing datalogical P-acts requires the forma ability. Contrary to C-acts, ICT systems are thus able to fully take over the datalogical P-acts performed by human beings to perform datalogical P-acts. We call services concerning datalogical P-acts provided by ICT systems datalogical ICT services. Terlouw distinguishes between datalogical ICT services that concern transportation and recording of information [54]. Recording of information comprises, for instance, registration, retrieval, alteration, and removal services as types of datalogical ICT services [54]. On the other hand, we could consider, for instance, distribution services as datalogical ICT services that concern transportation of information [54]. An email service is a typical example of a distribution service.

If we regard the organization theorem depicted in Figure 2.12, it can be noted that we can refer datalogical ICT services to the function of the D-application. Logically, one would denote the D-application as the service provider as we also defined the D-organization as the service provider of datalogical business services. However, it is questionable whether it is feasible for ICT systems to perform the role of a service provider. In Section 4.1, we regarded a service provider as the executor of a transaction. Dietz states that the fulfillment of an actor role requires all human abilities for dealing with coordination and production [18]. ICT systems, however, are only able to master the forma and informa human abilities as stated earlier. Therefore, ICT systems can never fully take over human beings in the fulfillment of actor roles. There should always be a human being responsible for the acts performed by an ICT system [18]. Hence, the D-application and the human being, who is held responsible for the ICT system and its service, fulfill the service provider role. In general, it can be stated that the fulfillment of the service provider role concerning ICT services is performed by the ICT system and the human being who is held responsible for the ICT system and the ICT service. This distinction is also made in the SOA domain, where the ICT system provides its service through a standard interface and a certain person or company is responsible for the service provided. In SOA, such an ICT system is known as the service component [49][48][41]. We further use this term in order to denote a B-, I- or D-application performing a particular ICT service.

Figure 2.12 exposes that the function of the D-application supports the construction of the D-organization composed of D-actors. This underpins the fact that a datalogical ICT service supports a datalogical business service. In addition, this means that the D-organization should be defined as the service consumer of datalogical ICT services. Besides the support of the construction of the D-organization, the function of the D-application supports the construction of the I-application as well, which can be seen in Figure 2.12. This implies that the I-application is the service consumer of datalogical ICT services. Since we defined a service consumer as the initiator of a transaction, it is an actor role like a service provider. Therefore, we should regard the I-application and the human being, who is held responsible for the I-application, fulfilling the service consumer role of a datalogical ICT service. Note that the I-application is the service component. Next to this, ICT systems as D-applications can also provide datalogical ICT services to each other and can consume these services.
Infological business services deal with infological P-acts such as reproducing and computing. Since ICT systems master the informa ability, they are able to perform infological P-acts. We call services concerning infological P-acts provided by ICT systems *infological ICT services*. Infological ICT services are, for instance, *calculation services*, *validation services*, *selection services*, and *matching services* [54]. If the infological ICT service supports human actors in realizing the infological business service, I-actors in the I-organization can be defined as service consumers. Next to this, as functions of I-application, infological ICT services support the construction of the B-application, which holds that B-applications and the human beings responsible for their acts performed should also be regarded as service consumers. Logically, I-applications and the human beings responsible for their acts performed are also the consumer of infological ICT services. The provider of an infological ICT service is composed of the I-application realizing the service and the human being responsible for the infological ICT service. As with datalogical ICT services, it is impossible to hold the ICT-system responsible for its actions since its ability is limited to the forma and informa ability.

Ontological business services produce ontological things by making decisions or judgments. Unlike infological and datalogical business services, it is actually impossible to implement ontological business services using ICT systems. ICT systems are unable to perform ontological P-acts due to their lack of the performa ability. Although ICT systems such as Business Process Management systems are often regarded as systems that make real decisions and judgements, this is actually not the case. In fact, such systems just calculate or compute new information on which B-actors base their decisions and judgements [18]. ICT systems such as Business Process Management systems are thus regular I-applications. However, in order to distinguish I-applications that support the B-organization from the pure I-applications that support the I-organization, I-applications that support the B-organization are defined as B-applications [18]. We call ICT systems that provide services that directly support ontological business services *ontological ICT services*. Unlike datalogical and infological ICT services, ontological ICT services cannot be divided into categories since they are specific to a certain industry [54]. B-actors should be regarded as service consumers of ontological ICT services, since ontological ICT services support them in performing ontological business services. B-applications and human beings responsible for the ontological ICT service can together fulfill the service provider role as well as the service consumer role.

As can be seen, it is also possible to define three types of ICT services on the basis of the organization theorem. All in all, if we take the definition of ‘service’ in SOA (see Section 1.3) and the above into account, we can define an ICT service more specifically on the basis of the ICT service definition of Terlouw [54]:

An *ICT service* is a service offered by a service provider to service consumers that conforms to the following properties:

1. it is accessible through an interface;
2. its implementation is hidden to service consumers;
3. it is autonomous;
it supports a business service;
5. it is either ontological, infological, or datalogical.

The first three properties in the definition are taken from the service definition given in Section 1.3 and were clarified in that section. The other two properties were derived from the theory of enterprise ontology and were explained above. Note that the fourth property differs from the original definition of Terlouw, who stated that an ICT service supports a transaction.

### 4.4 Composite Service and Service Component

The previous sections enable us to discuss the concept of a composite service in more detail. In Section 4.1, we stated that the concept of a composite service in SOA is related to the structure of an enclosed transaction in enterprise ontology. Because of this, we defined a composite service as a service from which the creation of its P-fact depends on other P-fact creations. Section 4.3 clarified that regarding ICT services a service provider consists of the ICT system as the service component and the human being who is held responsible for the acts performed by the service component. The service component partly performs the C-acts and fully performs the P-act that compose the service, however, the responsibility for this performance relies on a human being.

By defining a composite service as a service from which the creation of its P-fact depends on other P-fact creations, we regard the composition of the service purely on the basis of the information gathered as result of the execution of the underlying services i.e. the output of the underlying services. In other words, the composition of the service is based on the production acts of the underlying service components. On the basis of our definition of a composite service, we can model composite services as Figure 4.4. Figure 4.4 illustrates the composite service ‘complete flight booking’ which is built up from the atomic services ‘deliver data’ and ‘create flight ticket’. Acts with the same color belong to the same service component. As can been seen, the composition of a service on the basis of the production act, which the service component performs, holds that each component is assigned to exactly one service provider role. The other way around, a service provider role only consists of exactly one service component. This is due to the fact that the performance of a production act should be accompanied with the coordination acts that belong to the same service.

Another approach can also be made on composite service. In the business components domain (see Section 3.2), Albani and Dietz researched the identification of business components on the basis of enterprise ontology [5]. In order to determine the service composition, it does not only concern the production of the component, but also the relationships between the C-acts in performing the P-act [5]. Relationships between acts are illustrated with a black arrow, which can be seen in Figure 4.4. The relationship between single acts expresses how often an act is performed within a transaction and how close two acts are related to each other [5]. Albani and Dietz distinguish between standard, optional and main relationships, which express the cardinality constraints [5]. An optional relationship is represented by the cardinality range 0..k and it holds that the following act is not needed to be
Figure 4.4: Decomposition of a composite service on the basis of production acts

performed [5]. This is presented in Figure 4.4 using the link between the promise act of the service ’complete flight booking’ and request act of the service ’deliver data’. A main relationship can be expressed using the cardinality range \(1..n\) and it states that the following act is performed at least once [5]. The link between the promise act of the service ’complete flight booking’ and the request act of the service ’create flight ticket’ indicates a main relationship. A standard relationship indicates that the following act is exactly performed once. This is usually not expressed in a model as the other relationship types.

As stated above, a service component is assigned to exactly one service provider role. The other way around, a service provider role only consists of exactly one service component. With this known fact in mind, a service component can be regarded as a system according to the system definition of Dietz given in Section 2.5. Consequently, it can be said that a service component is built up from a set of elements. The elements of the service component perform the acts as part of the service and interact with the service consumer if necessary. Together, the performance of these acts brings about the service i.e. the function of the service component, which is determined by its production. What can be stated is that
the relationships between single acts refer to the structure of the service component as a system i.e. the set of influence bonds among the elements in the composition, and between them and the elements in the environment. The performance of these single acts can be regarded as interactions through these influence bonds. This holds that Albani and Dietz determine the composition of a service on the basis of the strength of the influence bonds between single acts.

Let us examine what is meant by a composite service if we follow Albani’s and Dietz his view on the composition of a service and what this means for the current notion on ‘service component’. Consider Figure 4.5. The figure gives an illustration of the same composite and atomic services, however, the composition is different. We do not consider the promise act, P-act and state act as a whole anymore, but we regard each C-act and P-act separately from each other. The composite service is composed of two service components. As can be seen, the distinction between the service components does not rely on the actor roles they fulfill. The dividing line between the two components is located between the promise act and P-act of the service ‘create flight ticket’. As a consequence, the service provider role for the service ‘complete flight booking’ and ‘create flight ticket’ is fulfilled by two service components. While the composite service of Figure 4.4 consists of two atomic services, it is questionable whether the two components of Figure 4.5 can independently operate from each other. If we consider the red component, we can see that the services ‘complete flight booking’ and ‘create flight ticket’ are partly tied to the red component and that the service ‘deliver data’ is fully tied to the red component. The question that arises now is whether we can speak of a service that the red component offers. On the one hand, it can be said that the red component offers the service ‘deliver data’ since it performs the pattern of promise act, P-act and state act of that service. Yet, we cannot speak of service offering in this case. The red component receives as input a request for the service ‘complete flight booking’. As output, it does not present the result of the service execution of ‘complete flight booking’, but it confirms that it executes service ‘create flight ticket’, since it performs the promise act of the service ‘create flight ticket’ as last. Hence, the red component cannot independently offer a service to service consumers.

Besides the situation described above, we can also regard a component which only partly ties to a particular service; the component only performs one specific act of the service or it performs two of the three acts that form the service. Consider Figure 4.6. The blue component can only perform the P-act and state act of the service ‘create flight ticket’. If this component is independently offered, it needs a promise act of the service ‘product construction’ as input. Yet, it is illogical to assume that the service consumer would perform the promise act of the service that he requests. Hence, the blue component cannot be regarded as a component that provides a service on its own.

The above implies that we should reconsider our notion on ‘service component’ given in the previous section. If we adhere to Albani’s and Dietz his view on the composition of a service, we cannot regard a service component as a system anymore, since the service component is then not necessarily tied to a production. Therefore we make a clear distinction between a service component and a service component part. We should redefine a service component as follows:
A service component is a system whose elements in the composition interact together to perform the promise act, decline act, P-act and state act of a particular service in order to deliver this particular service requested by the service consumer. It should meet the following conditions:

1. The first act performed is the promise act.
2. The decline act is performed if it disagrees with the request of the service consumer.
3. The last act performed is the state act.
4. The first and the last act should correspond to the same service.

Note that ‘composition’ refers to the composition property as part of the system definition of Dietz given in Section 2.5. Next to the notion of a service component, we have seen that a component is able to perform just a part of the service whereby it cannot fully perform the service, which can defined as the service component part. We can define such a service component part as follows:
A service component part is a set of elements of some category that interact together and does not meet at least one of the conditions of a service component.

Note that we do not speak of a system, since a system produce things according to the system definition given in Section 2.5. A service component part does not need to produce things by performing the P-act.

So far, this section studied the notion of composite services regarding ICT services as one can note. Here, we regarded service components as the actual executors of the acts that compose the services. They do not fully fulfill a particular actor role. For instance, if we regard Figure 4.4, the service component (partly) performs the C-acts and P-act as part of actor role A02 and the human being that is responsible for the acts of the service component completes the fulfillment of A02. Since we approached the concept of ‘service’ from a general perspective on the basis of the service definition of Elfataty and enterprise ontology up to now, it would be interesting to study whether the findings above can also be related to business services. As can be derived from [18], transactions generally concern

Figure 4.6: Decomposition of a composite service on the basis of single acts (2)
two actors that fulfill the role of initiator and executor. It considers the fulfillment of the acts of an initiator or executor as a whole and does not particularly concern which actor is performing a specific C-act or P-act as part of the initiator or executor role. Regarding the role of an executor, an executor is distinguished on the basis of the P-act that it performs. Hence, we can also speak of (de)composition of the service component on the basis of P-acts regarding composite business services.

However, in order to conclude whether we can relate (de)composition of a composite service on the basis of single acts to business services, we should first consider the notion of delegation according to enterprise ontology. Delegation is described as follows: 'The human being who is authorized for performing an act, may delegate authority to someone else. If a human being A delegates authority to human being B, human being A remains (also) responsible for the acts of human being B' [18]. Consider Figure 4.7 which illustrates delegation in a restaurant. The restaurant is managed by the owner and in order to support him in the restaurant he has taken several persons into employment. As the manager and the owner of the restaurant, he is authorized and responsible for providing services to the customer and realizing these services. However, he can transfer this authority to his employees. This implies that his employees may provide and realize the services in the restaurant and that they are responsible for the acts they perform. However, the manager still takes final responsibility for the services provided as the delegator. As can be noted, we can make the same distinction in business services as in ICT services regarding the fulfillment of the service consumer or service provider role; the one that actually performs the C-acts and P-act of the service and the one who takes the final responsibility of the acts performed. However, the employees are also responsible for their acts to a certain extent contrary to ICT systems that cannot be held responsible for their acts.

As can be seen in Figure 4.7, the completion of the service 'complete dinner order' concerns several underlying services. Under each C-act is the one denoted who performs that particular C-act. For instance, the order taker is the one who receives the service request from the customer. In enterprise ontology, the order taker is called a functionary [18]. In the figure, we assigned each functionary to a particular color. According to Dietz, the one who performs the promise act in a transaction is authorized to be the executor of that transaction [18]. We adhere to this notion as can be seen in the figure. The figure exposes three different functionaries which composes the service 'complete dinner order'. As can be seen, one functionary is not necessarily assigned to one particular service provider role. He may perform just one particular act. The service 'complete dinner order' can also be decomposed on the basis of single acts.

In all, we can distinguish two ways of service composition in business services as in ICT services. The previous two paragraphs arouse our interest whether we can also speak of the term ‘service component’ in business services. In that case, we should investigate if we can speak of systems that perform the C-acts and the P-act composing the service. The restaurant in Figure 4.7 can be regarded as an enterprise. It can therefore be seen as a collection of socially linked human beings [18]. In the restaurant, these human beings are the employees that take the order of customers, serve the dinners, etc. The restaurant can therefore be defined as a social system. If we regard the service ‘complete dinner order’, it is delivered by a collection of socially linked human beings and thus by a social system. In
general, each C-act can be assigned to one particular functionary. Since a service consists of several C-acts, it is always possible to have several functionaries that deliver a service. Hence, we can also use the term ‘service component’ in business services. The distinction with ICT services relies on the type of system, which is social regarding business services.

Now, the question arises whether it is also feasible to refer the term ‘service component part’ to business services. In that case, it should always be possible to have several functionaries as a set of elements performing a particular C-act or P-act. However, enterprise ontology states that a particular C-act or P-act is performed by one specific functionary. Therefore, we cannot use the term ‘service component part’ for business services. Yet, as will be explained in the next chapter, we do not further focus on service component parts during this research.

Let us again concentrate on the concept of a composite service. In conclusion we can say that in order to define what a composite service is, whether it is an ICT service or a business service, we should regard the productions from which it depends on and its composition from several components. Therefore we should redefine a composite service as follows:

A composite service is a service whose production is tied to a composition of several service components. Its production depends on the productions of those service components.
As a consequence, we should also redefine an atomic service as follows:

An atomic service is a service whose production is tied to exactly one service component. Its production does not depend on the productions of other service components. Furthermore, it is not composed of underlying services.

### 4.5 Service Provider and Service Consumer

In Section 4.1, we argued that 'service consumer' and 'service provider' can be regarded as the initiator and executor of a transaction in enterprise ontology, respectively. However, what urges us to reconsider our notion on 'service provider' is the distinction between a service component and the one who is held responsible for the acts performed by the service component, which may hold for ICT services as well as business services.

In the SOA domain, the one who is held responsible for the acts performed by the service component is generally called the service provider [14]. This does not match with our notion on 'service provider'. Since the term 'service provider' is already widespread used in practice for defining the one who is held responsible, we redefine our notion on 'service provider'. From now on, we call the service provider as defined in Section 4.1 the service providing party. The service providing party consists of the service component and the human being(s) that is responsible for the service performed by the service component. To be perfectly clear, the service component as part of the service providing party is defined as the providing service component. It actually performs the service. We call the human being or these human beings who are responsible for the service component the service provider. This is illustrated in Figure 4.8. In order to be clear, we show how this distinction relates to the executor of a transaction in enterprise ontology. The arrow signifies the mapping to the service domain.

The distinction in the service providing party shown in Figure 4.8 holds for ICT services and for business services in case of delegation. Yet, regarding other cases in business services, the one, who is held responsible for the service performance, is also the one, who actually performs the service. In other words, the service provider also does the actual performance of the service. This is shown in Figure 4.9. Consequently, the service provider is the providing service component as well.

Likewise, we can redefine our notion on 'service consumer' given in Section 4.1 as well, since we can also make a distinction between the service component and the human being(s), who are held responsible for the service component. Consequently, the service consumer as defined in Section 4.1 is known as the service consuming party from now on. We call the service component as part of the service consuming party the consuming service component. The human being(s), who are held responsible for the consuming service component, are defined as the service consumer. This is shown in Figure 4.10.

The distinction in the service consuming party depicted in Figure 4.10 holds for ICT services and business services in case of delegation. In terms of other cases in business services, the one, who is held responsible for the performance of the service component, is also the one who actually consumes the service. In other words, the service consumer also
4.5 Service Provider and Service Consumer

Service in Enterprise Ontology

Figure 4.8: The service providing party regarding ICT services and delegation in business services

Figure 4.9: The service providing party regarding business services
does the actual consumption of the service. Figure 4.11 depicts this concept. As a result, the service consumer is the consuming service component as well.

### 4.6 Evaluation

This chapter studied in what way the concept of ‘service’ could be uncovered using enterprise ontology. The general view on the concept of ‘service’ of Elfatatry enabled us to study which elements in enterprise ontology could be related to the service definition of Elfatatry. The fact that service concerns ‘any act or performance that one party can offer to another’ according to Elfatatry made it clear that we had to study the relation between a service and a transaction in more detail. As a result, it was found that a service could be regarded as a part of a transaction. This known fact gave us the opportunity to clarify composite and atomic services in SOA using enterprise ontology.

What could be obtained by means of the organization theorem was a clear picture of the different types of services that exist in an enterprise. The main distinction between services relies on the fact whether human beings or ICT systems provide the service. Services provided by human beings are known as business services, whereas services provided by ICT systems are called ICT services. Although these services are realized in a different way, they are strongly related to each other. ICT systems support human beings in their acts by partly taking over these acts. Likewise, ICT services support human beings in performing business services. A further breakdown in service types could be made on the basis of the production types. Consequently, we speak of six different service types that exist:
4.6 Evaluation

Service in Enterprise Ontology

A service component in SOA is regarded as the ICT system that provides a particular ICT service. According to enterprise ontology, the role of a service provider cannot be fulfilled by ICT systems. ICT systems only master the forma and informa ability and lack the performa ability, which means they cannot be held responsible for the acts they perform. Only human beings take responsibility for the acts they performed. Therefore, regarding ICT services, the fulfillment of the service provider role always consists of the ICT system and the one who is responsible for the ICT system. Logically, this also holds for the fulfillment of the service consumer role. This clear distinction made using enterprise ontology aroused our interest in finding out whether the concept of 'service component' is applicable to business services as well. The question to be answered was if it was possible to regard the service component as a human being or a set of interacting human beings having some
kind of leader who takes the final responsibility for the acts performed. Using the concept of ‘delegation’, we could confirm using enterprise ontology that the concept of 'service component' is also applicable to business services.

The above obliged us to redefine our notion on ‘service provider’ and ‘service consumer’. It urged us to make a clear distinction between the actor role, the one that actually performs the acts belonged to that role and the one that takes the final responsibility for these acts. This is because of the widespread use in practice of the term ‘service provider’ for defining the one who takes the final responsibility for the performance of the service. Therefore, we redefined the service provider as the ‘service providing party’ in order to adhere to the term ‘service provider’ in practice. The one that actually performs the acts as a service is now known as ‘providing service component’. In the same way, we redefined the service consumer as ‘the service consuming party’ consisting of ‘service consumer’ and ‘consuming service component’.

The most important insights we gained in this chapter are the fact that the concept of ‘service’ and all other aspects related can be applied to ICT systems as well as human beings. In addition, this chapter provided us insights into the types of stakeholders. On the basis of enterprise ontology, we can regard service consuming parties as stakeholders that need service specifications in their work. Service consumers and consuming service components represent more specific types of stakeholders. The next chapter elaborates on these insights wherein we study the relation between ‘service specification’ and enterprise ontology.
Chapter 5

Service Specification in Enterprise Ontology

The previous chapter identified the service elements on the basis of enterprise ontology. Now we are able to discuss service specification on the basis of enterprise ontology in more detail. Section 5.1 focuses on the service specification aspects that can be identified on the basis of enterprise ontology. In Section 5.2, we take a closer look at the role of conditions in service specification. Section 5.3 concentrates on the specification of business services. Afterwards, Section 5.4 deals with the ICT services. The chapter ends with Section 5.5 giving a brief review on the findings in this chapter.

5.1 The Concept of ‘Service Specification’

As mentioned in Section 1.4, service specification is a description of the external view of a service component and the service that it provides. It does not concern the internal details of the service component. In other words, it is the black-box model of the providing service component that stakeholders see. What is meant by the black-box model is the notion of specification in terms of the output variables of the system and their relationships with the input variables of the system [18]. Consequently, service specification concerns the function perspective of a providing service component. Figure 5.1 depicts the black-box view on the providing service component. As can be seen, the consuming service component can hand over information as input to be processed by the providing service component when performing the service [55]. The service result is the output of this process and can be handed over to the consuming service component [55].

The fact that service specification does not concern the internal details of a providing service component indicates that stakeholders do not notice whether they deal with a composite service or an atomic service. This implies that there is no distinction between specifying composite services and atomic services. Stakeholders do not know about the number of underlying service components or component parts composing the providing service component. From the stakeholder’s perspective, it is just one service component that provides the service.
5.1 The Concept of ‘Service Specification’  

Service Specification in Enterprise Ontology

Figure 5.1: Black-box view on a providing service component

We also stated in Section 1.4 that service specification consists of service function information and service usage information. It can be argued that these two parts of the service specification affect the type of information to be included in service specification. Regarding service function information, one would like to have general information of the service which is sufficient for finding the service and judging on whether it matches certain criteria. Service usage information would include more specific service information, which clarifies how to approach and use the service. Here below, we discuss the difference between these two parts of service specification in more detail and study how they are related to enterprise ontology.

5.1.1 Service Usage Information

What is meant by service usage information as part of the service specification is that it exposes information to stakeholders on how to approach the providing service component and make use of the service. Service usage information can be regarded as a tool for realizing successful communication between the consuming service component and the providing service component in offering the service. Regarding enterprise ontology, communication comprises the performance of C-acts in a transaction. Therefore, we should study the role of service specification in successfully performing C-acts in a transaction.

A successful performance of C-acts strongly relies on the performa, informa and forma ability. As illustrated in Figure 2.9, the formative exchange lays the foundation of a successful C-act performance. This holds that service specification should be presented in a form (e.g. a language or code scheme) that enables significational (or syntactic) understanding between the initiator and the executor. Significational understanding is a prerequisite for informative exchange. This holds that service specification should define the content aspects in such a way that the initiator and the executor share the same thoughts. In other words, the meaning of the content aspects should be clear for enabling sharing of thoughts. This is known as intellectual (or semantic) understanding, which is a condition for performative exchange. As stated in Section 2.4, performative exchange concerns social understanding in terms of exposing and evoking commitment. Unlike the forma and informa ability, it is
not quite clear how service specification enables social understanding. In fact, we think that this is not an aim of service specification. Service specification does not and cannot concern the commitment of the initiator or the executor. It rather supports social understanding by realizing syntactic and semantic understanding.

Let us now study the distinct C-acts in more detail. First, we consider the performance of the C-acts before the actual service execution. The main concern is thus how service specification can contribute in such way that the request act and promise act can successfully be performed with regard to the input information for the service. Let us first consider a request act, the service invocation. A request act implies that a service consuming party as a stakeholder (i) is in need of something (ii) contacts the party that can fulfill this need (iii) asks this party to fulfill this need. This shows that there are three aspects wherein service specification could fulfill a role here. The first aspect states that it should be clear for the service consuming party himself what kind of service he needs. Before he searches and requests a service, he should clearly define his service criteria. Service specification cannot contribute in this aspect, since the first aspect concerns the service consuming party who should define his needs. The second aspect implies that service specification should expose information on how to approach and connect to the providing service component in order to make use of the service and to send a service request (i.e. to perform a successful request act). This may hold for instance a set of connection rules or the location of the providing service component. It should be clear that the consuming service component should be connected to the providing service component before input information can actually be handed over. The third aspect denotes the necessity of an appropriate way to formulate the request in such a way that the providing service component understands. Here, service specification can assist the service consuming party by clarifying the language used and the terminology for enabling syntactic and semantic understanding. Note that this holds for each C-act and not particularly only for a request act.

The role of service specification during a promise act can be somehow vague at first sight, since a promise act is performed by the providing service component itself. Yet, service specification can be used in order to let a service consuming party know when the providing service component can perform a successful promise act. A providing service component will only promise if it agrees with the service request. In order to notify this to the service consuming party, the conditions for an agreement should be exposed through the specification. When the service consuming party has met these conditions, the providing service component performs the promise act. If not, it performs the decline act. In Section 2.3, we stated that the performance of the decline act results in a discussion state. This means that the consuming service component and the providing service component are in discussion how they can solve their disagreement. What role can service specification fulfill in this case? Recall the fact that the service providing party consists of the providing service component and the service provider in case of ICT services and delegation in business services. If the providing service component is an ICT system, it is impossible to discuss with it due to its lack of the performa ability. In this situation, the service consuming party should contact the service provider to solve the disagreement. When the providing service component is a human being or a set of human beings, the service consuming party can discuss with them how to solve their disagreement. If these human beings are not
fully responsible for the service, the service consuming party should be able to contact the
service provider. This signifies the need of contact information of the service provider to be
included in service usage information.

The state act comprises the statement that the actual service execution has been done.
This holds that the service result is available to the service consuming party. Let us con-
cern the C-acts performed by the consuming service component as respond to the state act.
Regarding enterprise ontology, the accept act should be performed in order to complete the
transaction. This act will only be performed if the service consuming party agrees with the
service result, the output that the providing service component processed from the input in-
formation. Before using the service, the service consuming party will probably have based
his choice, for instance, on the output that the providing service component delivers. It goes
without saying that the result from the service execution should also satisfy a number of
conditions. For instance, a service that calculates the area of a square cannot return a nega-
tive value as a result from the service execution. In this example, the condition could be for
instance that the service does not return a value below zero. These conditions should also
be exposed using the service specification. If the result has met these conditions, the service
was correctly executed whereby the service consuming party will accept the result by the
performance of the accept act. Nonetheless, it should be taken into account that the service
consuming party does not agree with the service result provided for any reason whatsoever,
while the result has met the conditions. As mentioned in Section 4.1, the performance of a
reject act should be viewed as another service call of the consuming service component in
order to undo the service result. The state information, which is needed as input data for the
performance of a reject act, should be provided by the providing service component along
with the performance of the state act. This holds that the consuming service component
does not need to concern about the input data necessary for the performance of a reject act.
The service provider is responsible for an appropriate delivery of state information by the
providing service component. Service specification should denote whether the providing
service component is able to provide this information and whether it gives the opportunity
to perform a reject act. This also holds for the performance of a cancel act as a service call.

What should also be taken into account is that the providing service component does
not correctly execute the service i.e. it can fail to return the service result according to
the conditions defined in the service specification. In that case, it is likely that the service
consuming party will not accept the result, but will reject it. Consequently, the consuming
service component performs a reject act. As a result, the consuming service component
and the providing service component end up in a discussion state. The service specification
should therefore denote how the providing service component deals with the fact that it
did not provide the result intended, but raises a fault. Conditions should be defined in
the service specification concerning faults that might raise if the result intended cannot
be provided for some reason, even though the conditions have been met by the service
consuming party [7]. These conditions should clarify which fault has raised and explain
the service consuming party how to handle this fault. If the service provider also actually
performs the service, the service consuming party can discuss with him concerning the
output of the service, which is not the service result intended. However, in case of delegation
in business services or regarding ICT services, the service consuming party should be able
to contact the service provider when the consuming service component and the providing service component end up in a discussion state. The service consuming party can notify the service provider concerning the fault and discuss with the service provider how to deal with the fault. This discussion may lead to the fact that the service consuming party still accepts the result or that the service is stopped.

![Focus of service usage information](image)

**Figure 5.2: The focus of service usage information**

### 5.1.2 Service Function Information

Service function information as part of the service specification should describe what the providing service component offers. It should expose the service properties needed for choosing the right service by a potential service consuming party. In order to determine the content of service function information, it is important to emphasize the distinction between service usage information and service function information. Service usage information concerns the connection of the consuming service component with the providing service component and concerns service usage when the connection has been established. The emphasis falls on the communication between the consuming service component and the providing service component and therefore service usage information concentrates on the performance of the C-acts. In that case, it is important to expose the manners for communication with the providing service component in terms of languages or code schemes. For instance, regarding ICT services, the Web Description Script Language (WSDL) [15] could be used for communication. Hence, information on communication between the consuming service component and the providing service component strongly depends on the human beings and
ICT systems that perform the service. On the other hand, service function information does not concern the communication between the consuming service component and the providing service component. By focusing on what the providing service component offers, the emphasis of service function information falls on the production of the providing service component. In other words, the function of the providing service component is determined by the P-act of the service. For instance, if the providing service component can create the P-fact ‘the costs of pizza order O have been paid’, this holds that the P-act as part of the service that the providing service component offers can be regarded as the payment of the costs of pizza order O. Besides the fact that the function of the providing service component can be referred to the P-act of the service, it can also be described in terms of the output variables and their relationship with the input variables of the providing service component as we stated above.

In addition, Section 5.1.1 exposed the important role of defining conditions in order to enable agreements between the consuming service component and the providing service component during connection. Regarding service usage information, conditions are established in order to successfully perform a certain C-act and a certain sequence of C-acts. In other words, these conditions focus on successful communication between the consuming service component and the providing service component and successful exchange of data and messages. Regarding service function information, we should concern conditions that involve the P-act of a service. In that case, these conditions do not state how messages and data should be transferred to each other, but what messages and data should be exchanged and presented to each other for successful service execution. This implies that we should distinguish between different types of conditions. We will discuss the different conditions in more detail in the next section.

All in all, service function information concerns the production of the providing service component. It exposes information on how successful performance of the P-act can be realized. In Figure 5.3, we illustrate the focus of service function information.

5.2 Conditions

As stated in the previous section, we should discuss the different types of conditions in more detail. The previous section exposed that we should clearly discern conditions on what messages or data are exchanged between the consuming service component and the providing service component and how these messages and data are exchanged in order to enable agreements between both parties. We first focus on the conditions that should be specified in service usage information, which concerns how messages and data are handed over to each other. Then, we discuss the conditions that are part of service function information.

5.2.1 Conditions in Service Usage Information

In order to enable message and data exchange between the consuming service component and the providing service component, these components should be able to communicate with each other successfully. As discussed in Section 5.1.1, successful communication
Figure 5.3: The focus of service function information

relies on the performa, informa and forma ability and consequently on syntactic, semantic and social understanding between the parties involved. In other words, there should not only be talk of understanding between the consuming service component and providing service component, but also between the service consumer and the service provider.

In order to establish agreements between the service consuming party and the service providing party concerning the performance of C-acts, the messages and data exchanged should be fully understood by both parties. Syntactic understanding between the service consuming party and the service providing party holds that the messages and the data should be presented in a form on which both parties agree. In order to enable semantic understanding, the service consuming party and the service providing party should agree on the content aspects of messages and data. As explained in Section 5.1.1, service specification does not enable social understanding in terms of exposing and evoking commitment, but supports social understanding by realizing syntactic and semantic understanding. This implies that there are no conditions or any other restrictions that can be defined concerning exposing and evoking commitment by the service provider and the service consumer.

The above shows that the communication between the consuming service component and the providing service component should somehow be regulated in order to enable syntactic and semantic understanding between both parties by complying with the forma and informa conditions. In the ICT domain, protocols are used for this purpose. In its simplest form, a protocol can be defined as the rules governing the syntax, semantics and synchronization of communication [34]. If we look more specifically at ICT services, protocols such as Internet Protocol (IP) [43], Transmission Control Protocol (TCP) [44], Hypertext
Transfer Protocol (HTTP) [24] and Simple Object Access Protocol (SOAP) [65] take an important role in regulating the syntax and semantics of communication and the data and messages that are exchanged. Most protocols also deal with errors in terms of detection and correction. One might think, for instance, of TCP which makes use of checksum in order to detect errors during communication [12]. In literature such as in [51] and [41], this concept of 'protocol' is often termed as 'communication protocol'.

Although the concept of 'protocol' as described above is mainly related to the ICT domain, it is not surprising that rules governing the syntax, semantics and synchronization of communication are also of great significance regarding business services. What the concept of 'protocol' actually holds is to define a common way for ICT systems for communication and data exchange. Regarding business services, we can note the concept of 'protocol' as a way for sharing thoughts by speaking, listening, writing or reading using a common language. All in all, successful communication between the consuming service component and the providing service component is established by using rules in the form of protocols rather than by defining conditions. Violating protocols will make communication more difficult, if not, completely impossible [53].

5.2.2 Conditions in Service Function Information

Conditions in service function information should define what kind of messages and data are exchanged between the consuming service component and the providing service component. We call these conditions function conditions. Before we further discuss the type of function conditions in detail, it should be noted that any data is logically somehow created as a result of a certain production act. We could consider, for instance, a request form that should be filled in as input data and should be handed over in order to make use of a service. If this request form has not been filled in, it is impossible to make use of the service. Somehow the condition 'the request form F has been filled in' should be true in order to execute the service. Note that the variable F refers to an instance of 'request form' (see Section 2.2). Logically, the filling in of the request form can be seen as a production act from which the condition 'the request form F has been filled in' can be mentioned as the result of that production i.e. as the production fact. The example conveys the impression that we may consider P-facts as function conditions that define what kind of data and messages should be exchanged between the consuming service component and the providing service component. Yet, P-facts do not necessarily concern the data or messages exchanged as the following example shows. A service might, for instance, remove all yesterday’s data in a certain department. The removal of yesterday’s data can be regarded as a production act with ‘yesterday’s data has been removed in department D’ as the production fact. Yet, another service might have as prerequisite that yesterday’s data in the department should be removed. In that case, the P-fact ‘yesterday’s data has been removed in department D’ should be viewed as a condition even though it does not state anything about the messages and input data that is exchanged between the consuming service component and the providing service component. This underpins the fact that we may generally state that function conditions are P-facts.

So far, we considered a P-fact as the production result of a P-act. Such a P-fact is
known as a production factum [18]. Yet, production facts are not necessarily the result of a production act. Such a production fact is called a production statum [18]. Let us consider, for instance, the statum ‘The minimum number of drinks is 0’. This P-fact does not denote any result of a P-act. Yet, it is possible to use it as a condition. One might think, for instance, of a service that calculates the monthly turnover of a restaurant. The turnover is calculated by multiplying the number of drinks and dinners sold with their individual prices and adding these values together. In that case, the statum ‘The minimum number of drinks sold is 0’ is suitable to be used as a condition for this particular service. Clearly, we can regard production stata as conditions as well. All in all, it can be stated that function conditions comprise the set of production facta that have been created and the set of production stata that exist. The sets of created facta and existed stata are known as the state of some world [18]. Consequently, the function conditions define the required state of the P-world (see Section 2.2) prior to service execution and subsequent to service execution.

Let us now discuss the different types of function conditions in more detail starting with conditions that should be met by the service consuming party prior to service execution. We call these conditions preconditions. Preconditions are the P-facts that have to be in existence before service execution. According to the previous paragraph, we should distinguish between preconditions defined as production facta and production stata. Since there are three types of P-acts (datalogical, infological and ontological), we can discern three types of production facta. Because of this, we will speak of datalogical preconditions, infological preconditions and ontological preconditions. Datalogical preconditions are P-facts resulted from datalogical P-acts that have to be in existence in order to execute the service. Infological preconditions are regarded as P-facts resulted from infological productions that have to be in existence prior to service execution. An infological precondition could be, for instance, that the P-fact ‘insurance policy premium P has been calculated’ should have come into existence before the service ‘bind insurance policy’ is executed. Ontological preconditions are the P-facts that have to be in existence before service execution and that result from ontological productions. We could consider, for instance, the P-fact ‘policy quotation Q has been created’ as an ontological precondition, which should be in existence before the service ‘bind insurance policy’ can be executed. Production stata that are used as preconditions are called statum preconditions. The previous paragraph gave an example of such a statum precondition.

Postconditions in service function information denote the P-facts that should exist subsequent to service execution. It goes without saying that we should discern three types of postconditions in the same manner as with preconditions. A datalogical postcondition defines a P-fact resulted from a datalogical production that has to be in existence subsequent to service execution. Let us consider, for instance, a service that completes a flight booking. In that case, a datalogical postcondition could be, for instance, ‘Flight tickets of flight booking B have been printed out’. An infological postcondition is a P-fact resulted from an infological production that should exist after service execution. Likewise, an ontological postcondition is a P-fact that has to be in existence subsequent to service execution and that resulted from an ontological production. Statum postconditions are the P-facts that have to be in existence after service execution, but did not result from a production act.

The above implies that an overlap between preconditions and postconditions might ex-
For instance, if we again regard the turnover service of the second paragraph, the precondition ‘The minimum number of drinks sold is 0’ might also be viewed as a post-condition which should be met subsequent to the service execution. Conditions, which are preconditions and postconditions as well and should be complied with throughout the service execution, are known as invariants.

As stated in Section 5.1.1, the service consuming party should be able to cope with faults that might raise even though the invariants and preconditions have been met prior to service execution. A fault is defined as a feature of a system that precludes it from operating according to its specification [6]. Regarding services, this holds that the service executed by the providing service component did not provide the result as defined in the specification. In other words, the service result did not meet the postconditions or invariants that are specified in the specification. Fault conditions should state what type of fault has raised. An additional message should accompany a fault condition in order to explain the service consuming party how to handle this fault.

All in all, several types of function conditions can be specified in order to define the input and output data and to enable successful performance of the P-act of a service. Now the question arises how extensive the set of function conditions should be. Regarding the SOA domain, services should be autonomous and loosely coupled. As a consequence, the service performance should be restricted to conditions as least as possible. In other words, a minimal set of function conditions is preferable. This holds that it is not needed to describe the complete state of the production world. Only those production facts that clearly affect the service execution or the input or output variables should be regarded as function conditions. Table 5.1 provides an overview of the distinct types of function conditions.

### 5.3 Business Service Specification

The previous sections discussed which service properties should be incorporated in service specification. Let us now study in more detail what this actually holds for specifying business services. Since we only concern business services in this section, we do not need to explicitly distinguish between the service component and the one who takes the final responsibility for the service component in this section.
In order to perform a service request, the service consuming party needs to know where it can find the service providing party. Regarding business services, the physical location of the service providing party may be denoted in the specification. We could consider, for instance, a service called ‘retrieve financial reports’ which is performed by an archivist. Regarding this service, the location of the archivist in the organization could be, for instance, the second floor room 2.110 of building A. Yet, there are other ways to indicate the location of the service providing party. It is not needed to particularly denote the physical location of the service providing party. One might think, for instance, of phone numbers or email addresses to be used in order to contact the service providing party.

The performance of a promise act depends on the fact whether the service consuming party and the service providing party agree with each other. In order to come upon this agreement, the service consuming party should meet several conditions, which were discussed in Section 5.2. Service usage information describes the protocols that are used for communication. Regarding business services, protocols are ways for sharing thoughts by speaking, listening, writing or reading using a common language. Protocols might be, for instance, the use of specific languages for communication such as English or Arabic. Yet, protocols could also be, for example, manners which depend on the domain of the business service. Rules on how communication is proceeded depend on whether these rules concern, for example, a bar tender in a cafe or a lawyer at a business meeting. As can be seen, the type of protocols depends on the human beings that provide the service. Service function information denotes the preconditions and invariants that have to be in existence prior to service execution in terms of P-facts. P-facts resulted from P-acts have been created by the execution of other services. It does not matter whether these P-facts came into existence as a result of business services or ICT services. Regarding the service ‘retrieve financial reports’, a precondition could be, for instance, that the service consuming party should have been registered at the archive department. In terms of P-facts, the precondition can be defined as ‘Person P has been registered’. The service ‘retrieve financial reports’ does not need to know whether the precondition has been created using a business service or an ICT service. The performance of a promise act should only be regulated using the invariants and preconditions and should not concern how these invariants and preconditions came into existence. When at least one of the conditions is violated, the service providing party performs a decline act. The service consuming party can discuss with the service providing party to still enable an agreement. In case of delegation, there is a clear distinction between the service provider and the providing service component as parts of the service providing party, where the actual service performance is done by the providing service component. In that case, the service consuming party should contact the service provider as the one who takes the final responsibility for providing the service.

The actual service execution is followed by the statement of the service providing party that the execution has been performed. As prior to service execution, the service consuming party should hold to the protocols in order to receive the service result successfully. The service result itself should meet the postconditions defined in the specification. A postcondition could be, for instance, that the financial reports are written in English. If the service result did not meet the postconditions (e.g. the reports are written in German), the service consuming party will reject the service result. In other words, the service consuming party
rej ects the service result due to a fault of the service execution. As mentioned in Section 5.2.2, fault conditions can clarify the fault and how to deal with it. Yet, regarding business services, it is more likely that the service consuming party and the service providing party enter into a discussion in order to solve the fault. In case of delegation, the service consuming party can contact the service provider in order to enter into discussion.

What we discussed so far holds for each type of business service. However, what relies on the terminology used, is the difference between the specifications of the distinct types of business services. As stated in Section 2.5 regarding the organization theorem, each underlying layer supports construction of the layer above by means of its function. This function should be described in terms of the operation of the layer that is supported by that function [18]. Regarding services, this holds that services should be described in terms of the operation of the service consuming parties. Since the I-organization and the D-organization are the service consuming parties of datalogical business services (see Section 4.2), specification of datalogical business services should be described in terms of the operation of the I-organization and the D-organization. If we take a look at the terminology used for specifying infological business services, we can argue that these specifications should be described in terms of the operation of the B-organization and the I-organization. This is due to the fact that the B-organization and the I-organization are the service consuming parties of infological business services. Since the B-organization is the most upperlying aspect organization, ontological business services are offered to the environment of the enterprise as well, besides within the B-organization itself. This implies that ontological business services should be specified in terms of the operation of the B-organization and the environment of the enterprise.

5.4 ICT Service Specification

As mentioned in Section 4.3, ICT systems only control the informa and forma ability and are unable to enter into or comply with commitments contrary to human beings. The question arises to what extent this difference between human beings and ICT systems affects the specification. If we again consider Section 5.1.1, we mentioned that service specification only plays an important part in the formative and informative exchange between actors. Therefore, it can be stated that service specification fulfills the same role in ICT services as in business services. Let us now discuss ICT service specification in more detail.

The performance of a request act requires knowledge about the location of the providing service component. Unlike business services, we do not consider the providing service component’s physical location regarding ICT services. In SOA, an automated service should be requested by the consuming service component irrespective of its location [40]. Basically, the location of the providing service component is defined as an address in the form of a Uniform Resource Locator (URL).

In order to enable an agreement between the consuming service component and the providing service component for performing the promise act, the consuming service component should comply with the protocols, invariants and preconditions defined in the specification. Regarding ICT services, several protocols have been proposed in SOA such as
These protocols are specifically used for communication between ICT systems. Regulating communication between the consuming service component and the providing service component strongly depends on whether it concerns a business service or an ICT service. Invariants and preconditions indicate the required state of the production world in order to execute the service. As with business services, the performance of a promise depends on the P-facts that denote the invariants and preconditions and does not consider how these P-facts have been created. As a consequence, it does not matter whether the invariants and preconditions came into existence using business services or ICT services. A decline act is performed if the protocols, invariants or preconditions have been violated. This signifies a disagreement and consequently the consuming service component and the providing service component end up in a discussion state. Regarding ICT services, it is impossible to negotiate or discuss with ICT systems as providing service components. Hence, the performance of a decline act should be avoided as much as possible, which implies that the protocols, invariants and preconditions should be defined as clear as possible. Therefore, formal languages are required for defining protocols, invariants and preconditions regarding ICT services in order to avoid ambiguity in its expression to ensure consistency. A language is formal, when syntax and semantics of the language are precisely and unequivocally defined.

The actual service execution results in the performance of the state act whereby the service result is presented to the consuming service component. Postconditions should be defined in order to determine the conditions to which the service result should comply with. As with invariants and preconditions, it is important to define postconditions in such a way that ambiguity in the expression is avoided and consistency is ensured, since it is impossible to discuss with ICT systems. It is therefore of great significance to use formal languages for defining postconditions. Unclear postconditions could lead to misinterpretations by the service consuming party whereby he could expect different service results. Fault conditions should also clarify faults as clear as possible in order to restrict making contact with the service provider as much as possible.

The above discussions hold for each type of ICT service. The difference regarding the specification of each type of ICT service is based on the terminology that is used for specifying the service and thus on the types of consuming service components of the service. Contrary to business services, ICT services may have human beings as well as ICT systems as consuming service components. In order to obtain a clear view on the terminology used, we first focus on one particular type of ICT service, which is ‘datalogical ICT services’. The I-application can be regarded as the consuming service component of datalogical ICT services. Consequently, specification of datalogical ICT services should be described in terms of the operation of the I-application. In that case, datalogical ICT services do not deal with human interventions since they are invoked by an infological ICT service. The question presents itself how service specification can be described in such a way that the ICT system as an I-application is able to select the datalogical ICT service that matches its criteria. Before we deal with this question, we should first recall what we stated about service specification in Section 5.1.1. We indicated that service specification can be used for realizing formative and informative exchange between the consuming service component and the providing service component i.e. it enables syntactic and semantic understanding.
In this situation, the service specification should indeed be syntactically and semantically understood by the ICT system as a consuming service component. This holds that service specification should be machine-readable and machine-interpretable. Even though one can find much literature on defining machine-readable and machine-interpretable information such as in [32], it is still questionable whether this is feasible for large enterprises since it takes a lot of effort [54]. Therefore, we should consider the other aspect we stated about service specification; support of performative exchange and social understanding. This aspect implies that service specification should support the service consumer as the one who is responsible for the consuming service component. Hence, service specification should be described using the terminology of the service consumer rather than the ICT system as the consuming service component. We could consider, for instance, a service consumer who selects the appropriate datalogical ICT services in order to realize a particular infological ICT service. As stated in Section 4.3, the D-application should also be regarded as consuming service components, since ICT systems as D-applications can provide datalogical ICT services to each other. This would hold that specification of datalogical ICT services should be described in terms of the operation of the D-application. Yet, it should be noted that the same discussion on specification concerning the I-application as consuming service component also holds for the D-application as consuming service component. The D-organization is the service consuming party of datalogical ICT services, which means that it is the service consumer as well as the consuming service component. This is because of the fact that datalogical ICT services can support and take over the human acts performed at a datalogical business service. This holds that the specification of datalogical ICT services should be described in terms of the operation of the D-organization.

The above gave a clear view on the terminology used regarding datalogical ICT services. In terms of infological ICT services, we should consider the B-application, I-application and the I-organization as the service consuming parties. Regarding the B-application and I-application, the specification should be described using the terminology used by the service consumers, who are responsible for the B-application and I-application as consuming service components. The I-organization as the service consuming party holds that the specification of infological ICT services should be built up in terms of the operation of the I-organization. Ontological business services have the B-application, the B-organization and the environment as the service consuming parties. In order to come upon an understandable service specification for the B-application, the specification should be described using the terminology used by the service consumer, who is responsible for the ICT system as a B-application. Furthermore, the specification should also be constructed on the basis of the terminology used by the B-organization. Next to this, the terminology in the specification depends on the environment of the enterprise as well.

5.5 Evaluation

The previous sections showed a different glance on service specification than current approaches on service specification. By studying the role of service specification using enterprise ontology, we were able to identify the aspects that should be described in the service
As mentioned in Section 1.4, service specification consists of service usage information and service function information. This chapter explained in a clear way what is actually meant by service usage information and service function information. Enterprise ontology enabled us to uncover the fundamental differences between these two parts of service specification. Service usage information contains the information needed to communicate with the providing service component successfully. In terms of enterprise ontology, service usage information concerns the successful performance of the C-acts of a service. As stated in Section 1.4, service function information describes what the providing service component offers. As can be seen in this chapter, this holds the P-act of the service and the input and output variables. Service function information contains the information needed to perform the P-act of the service successfully.

The fundamental difference above also clearly comes forward in the conditions that should be specified in order to come upon successful communication and service execution. Conditions in service usage information should rather be regarded as protocols. Protocols give the opportunity to enable syntactic and semantic understanding between the consuming service component and the providing service component. Conditions in service function information indicate which activities have to be performed prior to service execution and the conditions to which input and output data should comply with. These conditions have been created themselves using other P-acts or always exist. Together they form the required state of the production world prior to and subsequent to service execution.

The previous chapter discussed the concept of 'service component', which implied that a clear distinction should be made on the actual performance of the C-acts and P-act composing the service and the one who takes the final responsibility for this performance. This resulted in the distinction between the providing service component and the service provider in the fulfillment of 'service providing'. This distinction has a clear effect on the composition of a service specification. Since providing service components are just partly responsible or even not responsible at all for the acts they perform, they cannot be held fully responsible in case of faults or other failures. In addition, regarding ICT services, it is impossible to enter into a discussion when the consuming service component and the providing service component end up in a discussion state resulting from the performance of a decline or reject act. Therefore, information about the service provider should be part of every service specification. We call this service provider information. This information basically consists of two parts; a description of the service provider and contact information in order to contact the service provider.

Another topic of discussion in this chapter was the terminology used in service specification. It was found that terminology was the decisive factor in the differences in service specification regarding the different types of services. The terminology used in the specification is based on the operation of the service consumer and the consuming service component if the component consists of human beings. Since each type of service has a different set of service consumer parties, each type of service should use a different terminology for specification. Since a clear terminology used in the specification enables semantic understanding, it is of great significance to have a separate section in a service specification which describes the terms used and their definition in order to avoid ambiguity in interpretation of
5.5 Evaluation Service Specification in Enterprise Ontology

terms.

As can be seen, service specification consists of four essential parts, which are service provider information, terminology, service function information and service usage information. From now on, we call these parts areas of concern, since they all specifically have separate concerns regarding a service. On the basis of the findings in this chapter, we can already set up the specification framework as shown in Figure 5.4

As can be seen in the figure, ‘service function information’ and ‘service usage information’ are linked with ‘terminology’. The arrows signify that terms used in ‘service function information’ and ‘service usage information’ should comply with the descriptions of these terms given in ‘terminology’.

If we compare our approach to other service specification approaches such as the ones described in Chapter 3, it can be seen that there are more important service elements to be specified than just the ones identified using enterprise ontology. One can note that these elements are the quality aspects of a service. As stated in Section 1.4, service specification is used to judge the usefulness of the service. Logically, usefulness is based on the extent to which a service can fulfill a stakeholder’s needs. These needs do not only concern the function of the service, but also its quality [63]. This has also been acknowledged by Tosic et al., who stated that quality aspects are essential to specify, since they help to predict and
prevent problems concerning services and can help in selection between services with the same or similar functionality [56]. Enterprise ontology did not enable us to identify quality aspects of a service that should be specified. This also arouses our interest whether there are other stakeholders than service consuming parties. We will discuss this in the next chapter.
Chapter 6

Service Quality

This chapter discusses the quality aspects of a service that should be specified. Even though enterprise ontology did not mention any quality aspects, it is of great importance to include these aspects in service specification as stated in several literature such as [58] [56] [36].

Section 6.1 deals with the fact what the concept of ‘quality’ actually implies. Section 6.2 and Section 6.3 look at the concept of ‘quality’ related to ICT services and business services. Hereafter, we investigate in Section 6.4 how quality aspects of services can be identified fitting into the concept of ‘quality’ defined in the first section. Section 6.5 studies the quality aspects identified in greater detail and discusses whether these aspects can be related to ICT or business services. The chapter ends with Section 6.6 giving a brief review on the findings in this chapter.

6.1 The Concept of ‘Quality’

According to the ISO/IEC 8402 standard, ‘quality’ can be defined as the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs [38]. As can be derived from [58], there are different types of stakeholders regarding service specifications in the SOA domain, which implies different perspectives on services. Business people and technical people, for example, have distinct views on services. Each of type of stakeholder has its own stated and implied needs with respect to using services. As a consequence, each stakeholder differently interprets the concept of ‘quality’ regarding services. The differences between stakeholders also come clearly forward in our research so far considering the distinct types of service consuming parties we identified in the previous chapters.

Recall Section 4.2, Section 4.3 and Section 5.1. In Section 4.2 and Section 4.3, we clarified that different types of service consuming parties exist regarding services. Next to this, Section 5.1 stated that services should be described in terms of the operation of the service consuming party, which implies different terminologies to be used for specifying services. All in all, it is thus of great concern to expose the quality aspects of services which are of interest to each specific type of service consuming party. The quality aspects of interest might differ between the distinct types of service consuming parties and thus between dif-
6.2 ICT Services and the Concept of ‘Quality’

In literature, much has been written on quality of services in SOA [36] [40]. Although several quality aspects have been proposed in literature in order to determine the quality of a service, no literature has been found, which concerned with the different types of service consuming parties varying from business to technical, despite a thorough study. Literature does not clarify what quality aspects are important to be included in service specification for each type of service consuming party. Therefore, it is interesting to consider the view of Salmela on ‘quality’ [47]. Salmela states that there is a clear relation between the quality of ICT systems and the business quality of ICT systems. Business quality is defined as the net value of an ICT system for the organization of an enterprise, which is affected by both the cost of planning, developing, maintaining and using the system, and by the benefits achieved through systems use [47]. Salmela elaborated on the Software Library Evolution (SOLE) quality model of the IS Quality group [21] [20]. This theoretical model aims at a division of quality classes consistent with the different decision makers decisions made during the software life-cycle [21].

Even though Salmela does not focus on ICT services, it deals with ICT systems and their functions. As mentioned in Section 4.4, ICT systems and their functions can be regarded as providing service components and the ICT services they provide. Next to this, Salmela’s model focuses attention on the decision makers regarding the development and use ICT systems. In terms of services, we can consider such a decision maker as a type of stakeholder. As a consequence, this model enables us to take the distinct perspectives of stakeholders into account. This makes Salmela’s model suitable to relate ICT services to the concept of ‘quality’.

Figure 6.1 gives an overview of Salmela’s model. As can be seen, business quality is determined by IS work quality, IS user quality and business integration quality. IS work quality comprises the quality aspects of an ICT system, which aim at ensuring efficient delivery and maintenance of ICT systems according to user requests [47]. This includes aspects that concern the development and maintenance of ICT systems such as testability and maintainability. IS user quality implies the quality of ICT systems from the perspective of users [47]. In other words, it refers to what the ICT system does for the users and how well it performs its assignments [8]. Salmela states that users of ICT systems can be found on a management level or operation level of the organization [21]. This known fact is quite interesting, since we only identified stakeholders from an operation level as service consuming parties using enterprise ontology. This point will be elaborated on later. Business integration quality views ICT systems from a business perspective and involves strategic,
tactical and operational issues [47]. Strategic and tactical issues concerns the relation be-
tween investments in the ICT systems that should be made and the business benefits [47].
Operational issues deal with issues regarding the beneficial uses of ICT systems. They con-
cern with the extent to which the capability of the ICT system can meet the requirements
from users. As Salmela states, business integration can be regarded as a process of match-
ing IS work quality and user quality with business requirements [47]. Business integration
quality can therefore be described as the extent to which IS work and user quality matches
with business requirements.

Let us now study how Salmela’s view on ‘quality’ relates to ‘service’ and ‘service spec-
ification’. It can be noted that IS user quality covers the quality aspects of ICT services
from a stakeholder’s perspective. Logically, we should include quality aspects that adhere
to IS user quality in ICT service specification. As mentioned above, users can be found
on a management and operation level. It should be noted that stakeholders have only been
identified on an operation level so far, since enterprise ontology only focuses on the opera-
tion of an enterprise. ‘Management’ should be regarded as the governing of the operation
of the enterprise. In other words, management implies the regulation of the operation of
enterprise. In other words, management implies the regulation of the operation of
enterprise. Regarding services, stakeholders from a management level are not the users
of the service, which we defined using enterprise ontology. Yet, they do need to validate
whether the service fits into the operation of the enterprise. In other words, they judge on
the service whether it fits into the needs of the organization. As mentioned in Section 1.4,
judging the service is one of the reasons for using service specification. This implies the
distinction between stakeholders on a management level and an operation level which we
should concern with regarding service specification.

IS work quality embraces quality aspects that concern the development and mainte-
nance of ICT systems. Regarding ICT services, we should consider those quality aspects,
since the effort needed to make modifications in ICT services for maintenance by the service providing party may affect the usage of ICT services for service consuming parties as stakeholders. Business integration quality can be regarded as the investments made in an ICT service related to the benefits achieved with this ICT service. In other words, it concerns with the costs for using the ICT service and to what extent it supports service consuming parties in performing business services. It relates IS work quality and IS user quality to business requirements. As can be seen, business integration quality can be of great significance in selecting the ICT service which matches the criteria of service consuming parties and stakeholders at a management level and which meets business requirements. It should therefore be included in ICT service specification.

What is actually more complicated is defining the relation between business quality and ‘service’ and ‘service specification’. Contrary to business integration quality, IS work quality and IS user quality, it is questionable whether business quality is of any concern to stakeholders. According to Andersson and von Hellens, business quality is the concern of senior management and management at the departmental levels [8]. Regarding the chief executive officer (CEO), the key quality factor is utility measured in terms of costs and benefits, a satisfactory level of quality is achieved when the benefits, expressed in monetary terms, exceed the costs [8]. Logically, such managers do not concern with service selection and service usage. Therefore, we do not need to consider business quality in ICT service specification.

In all, there are distinct quality dimensions, which represent the distinct views on ‘quality’ that distinct types of stakeholders can have. It was found that ‘quality’ does not only concern service consuming parties, but also stakeholders on a management level, who select the services and judge whether they can meet the business requirements. Figure 6.2 depicts the scope of service specification.

6.3 Business Services and the Concept of ‘Quality’

If we consider ‘quality’ regarding business services, we should regard the quality of the providing service component and the business service that it provides. Despite a thorough study, we could not find any literature which specifically concern ‘quality’ regarding business services. Yet, literature was discovered in the area of management strategies on quality improvement such as Total Quality Management (TQM) [35], which could possibly be used for investigating business services and ‘quality’. It was found that these strategies are not suitable since they only focus on meeting the requirements of the customers in the environment of the enterprise. Yet, there is one management strategy on quality improvement, which stands out from the other management strategies. This management strategy is known as Six Sigma [46]. Contrary to other management strategies on quality improvement, Six Sigma strongly focuses on improving ‘quality’ by improving business processes [33]. As mentioned in Section 1.2, a business process is an ordered collection of business activities. Such a business activity can be viewed as a business service, whereby it was worth to study Six Sigma in more detail and to investigate whether we could relate Six Sigma to business services in a certain way. Six Sigma defines ‘quality’ as any production element that sup-
ports the customer’s vision of the product [42]. As also stated by Kumar et al., Six Sigma is a management strategy which provides a roadmap to continuously improve business processes to eliminate defects in products, processes and service [59]. Note that ‘service’ in Six Sigma is unrelated to our notion on ‘service’. Six Sigma enables to make its notion on ‘quality’ measurable using a set of metrics, which can quantify defects in business processes. To be perfectly clear, Six Sigma defines a defect as the production elements that interfere with the customer’s vision [42]. We could consider, for instance, the metric ‘defects per million opportunities’, which measures the number of defects in a process in terms of million opportunities. Despite the fact that Six Sigma enables to define the quality of processes, it was found that Six Sigma only focuses on meeting the requirements of the customer outside the enterprise. There were no opportunities to concern the requirements and expectations of the employees in performing business services as part of business processes. Therefore, Six Sigma does not fit into our concept of ‘quality’.

Another opportunity to study business services and the concept of ‘quality’, is to investigate whether we can incorporate Salmela’s view on quality to business services. As can be seen from the previous section, Salmela’s view consists of three types of quality affecting business quality and focuses on ICT systems and their functions. Regarding ICT services, we stated that such an ICT system can be regarded as a providing service component, which provides an ICT service. In order to incorporate business services to Salmela’s view on ‘quality’, we have to study whether Salmela’s view is also applicable to providing service components if providing service components are composed of human beings.

Let us first focus on IS work quality. If we refer IS work quality to providing service components, we should define IS work quality as the quality aspects of a providing service
6.3 Business Services and the Concept of ‘Quality’

Service Quality

component, that concern the development and maintenance of providing service components. Now the question arises whether we can speak of ‘development’ and ‘maintenance’ of a providing service component if the providing service component consists of human beings. What can be regarded as ‘development’ is composing the providing service component for providing the service. In order to clarify this, we consider the example given in Section 4.4 regarding the service ‘complete dinner order’ in a restaurant. Here, the providing service component is composed of the functionaries ‘order taker’, ‘dinner maker’ and ‘dinner serving’. In order to provide the service ‘complete dinner order’, human beings should fulfill the functionary roles appropriately. ‘Development’ can be referred to the engagement of human beings for fulfilling the functionary roles. Maintenance of a providing service component can be regarded as giving training to the human beings composing the providing service component so that they can fulfill their role as well as possible. Such a training can be, for instance, gaining knowledge about all types of wine that the restaurant serves. As can be noted, we can relate IS work quality to business services. In order to cover ICT services and business services as well, we rather speak of service component work quality from now on instead of IS work quality.

Regarding providing service components, IS user quality should be interpreted as the quality of providing service components from the perspective of stakeholders. It concerns the quality aspects that denote the quality of the function of the providing service component and the quality of the communication with the providing service component. Regarding business services, one might think, for instance, of a group of consultants who are hired to participate in a project of a particular company. These consultants can be regarded as a providing service component carrying out the project as a service. From the perspective of the company, the quality of the providing service component and the service provided is determined, for instance, by the time needed to complete the project. For instance, a group of consultants, who finishes a project within two months, are considered to provide a better service quality compared to another group of consultants, who finishes the same project with the same result but after six months. As can be seen, we can relate IS user quality to business services. In order to refer to both ICT services and business services, IS user quality is called service component user quality.

In terms of providing service components, business integration quality can be regarded as the extent to which service component work and user quality matches with business requirements. It can be viewed as the costs for using the service provided by the providing service component and the benefits achieved using the service. Here, we can again consider the example of a group of consultants as the providing service component. Basically, hiring consultants involves considerable costs. However, in return, you get a lot of knowledge and specialism which aid in finishing the project successfully. One might think, for instance, of a company with less ICT knowledge that hires ICT consultants in order to participate in an ICT project of the company. In that case, the benefit achieved is the ICT knowledge the company gains for finishing the project. It is likely then that making use of the ICT consultants will lead to a better project result within less amount of time. Hence, business integration quality can be related to business services.

In all, it is possible to relate Salmela’s view on ‘quality’ to business services. Yet, we use other terms in order to fully cover the concept of ICT services and business services.
6.4 Methods for Identifying Quality Aspects

The previous sections clarified the distinct quality dimensions focusing on distinct types of stakeholders. Now we should investigate which methods exist for identifying quality aspects that belong to these quality dimensions.

Quality aspects of services are often discussed in literature concerning SOA such as in [36] and [39]. However, researchers have often distinct views on the concept of ‘quality’ and the quality aspects whereby there is no unambiguous notion on ‘quality’ and its aspects. Turowski et al. proposed a procedure for defining and specifying quality criteria, which copes with the different views on the concept of ‘quality’ [4]. One of the steps in the procedure concerns defining the concept of ‘quality’ using a model or framework which can make the concept of ‘quality’ operational. What is meant by ‘operational’ in this case, is that ‘quality’ should be measurable. Since we already formed our view on the concept of ‘quality’, we should find a model or framework which fits into our view and which enables to make our concept of ‘quality’ operational. Unfortunately, there are almost no models or frameworks known that could make the concept of ‘quality’ regarding services operational. In most cases, researchers enumerate the quality aspects that are important from their own perspective for making the concept of ‘quality’ operational such as in [36]. A well-known framework in SOA for making ‘quality’ operational is the Web Service Level Agreements framework [31]. The WSLA framework has been developed in order to specify and monitor
Service Level Agreements (SLA) for Web Services [31]. A Service Level Agreement is an agreement regarding the guarantees of a service and it defines mutual understandings and expectations of a service between the service provider and service consumers [29]. These guarantees focus on the service quality. However, as stated by Trienekens et al., a Service Level Agreement is often a technical document regarding concepts and terminology that can only be understood by a small group of technology oriented specialists [57]. Since we concern varied types of stakeholders, we should find another method for identifying quality aspects.

The Extended ISO-model is another well-known framework for making the concept of ‘quality’ operational [63]. It makes the concept of ‘quality’ operational by providing a set of quality aspects which can be measured and verified. The Extended ISO-model does not specifically focus on services, but it concerns quality aspects of ICT systems in general. The Extended ISO-model defines six main quality characteristics, which can be divided into several quality subcharacteristics. This is shown in Figure 6.4. The Extended ISO-model is based on the ISO 9126-model [37], which has been developed by the International Organization for Standardization (ISO). As the name already suggests, the Extended ISO-model is an extension of the ISO 9126-model. It includes a number of additional subcharacteristics compared to the ISO 9126-model. Each subcharacteristic can be measured on the basis of a set of indicators. By these measurements, it can be stated to what extent the quality subcharacteristic is realized regarding a particular ICT system. For instance, in order to indicate to what extent the characteristic ‘security’ is realized, one could measure the data damage ratio [63].

![Figure 6.4: The Extended ISO model](image)

Although the Extended ISO-model is focused on quality of ICT systems, it can be found suitable for identifying quality aspects of ICT services, since ICT systems can be regarded as providing service components. In addition, the Extended ISO-model is also appropriate for identifying quality aspects of business services. First, unlike the WSLA framework, the descriptions of the Extended ISO-model are independent of technology [63]. In other words,
each quality characteristic or subcharacteristic does not require any reference to technology and technological standards for measurement. References to hardware platforms, development environments and types of software are not parts of the Extended ISO-model [63]. This enables us to relate quality characteristics to ICT systems as providing service components and human beings as providing service components as well. Furthermore, it gives the opportunity to make quality characteristics understandable to all types of service consuming parties and to stakeholders at a management level. Second, although the Extended ISO-model provides a wide-variety of quality characteristics, it does not oblige to describe all of these quality subcharacteristics in the specification of a particular software product. In fact, the model clearly states that only the quality characteristics that are relevant to the software product should be specified [63]. By this, the model gives us a free hand to select the quality characteristics that fit into the domain of ICT or business services.

Regarding Salmela's view on quality, the Extended ISO-model enables to measure IS work quality, IS user quality and the operational part of business integration quality. In order to completely identify all quality aspects regarding the scope of service specification as depicted in Figure 6.3, we should investigate in what way benefits achieved through service usage and costs for service usage as part of business integration quality can be described in service specification. Regarding the benefits of service usage, it is questionable whether it is possible to specify this. In this situation, ‘benefit’ is of subjective nature and depends on the needs of the stakeholder. It is therefore difficult to specify. Yet, the degree of benefit is something which could be of overriding importance in selecting a service. In order to anticipate on the various needs of distinct stakeholders, the service providing party should define different service levels. A service level is defined as the quality level of a service. The quality level is often determined by one or several quality aspects, which can be taken from the set of quality aspects that have been quantified. Literature shows that service levels are generally based on the service’s performance and availability [45] [30]. For example, a service could be delivered with two different service levels; one service level denoting 24/7 availability of the service and another service level representing service availability of one hour per day. Regarding the consultants example, one service level may hold that the consultants only work on the project for two days a week. Another service level could indicate that the consultants work on the project for five days a week.

Let us now discuss how costs of using the service should be described in service specification. What should be clear from the service specification for stakeholders is how service usage is charged. Gisolfi proposes several models for charging service usage [25]. The Transactional model refers to the pay-per-click or fee-for-use model [25]. It holds that service usage is paid per time the service is used. The Membership or Subscription model refers to the model that pertains to an established user account with specific terms for usage [25]. Such a membership or subscription holds that periodic access to a service can be obtained. Charges are per period, which might be weekly, monthly or annual. Different types of memberships could be offered on the basis of the number of times the service will be used. The Lease or License model refers to the model that is based on charges by volume of transactions [25]. In other words, the charges are based on the number of times the service is used.

Summarizing, the identification of quality aspects of services strongly adheres to find-
6.5 Identification of Quality Aspects

This section discusses which quality (sub)characteristics of the Extended ISO-model fit in IS work quality, IS user quality or business integration quality and can be regarded as quality aspects for ICT services or business services. Prior to this determination, a thorough study was carried out to find out which quality characteristics should be regarded as quality aspects for ICT services and business services. A full description of this study can be found in Appendix A. As a result from this study, we identified several quality subcharacteristics which can be regarded as quality aspects for ICT services or business services. Table 6.1 gives an overview of the quality aspects identified.

<table>
<thead>
<tr>
<th>Quality Characteristic</th>
<th>ICT Service</th>
<th>Business Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>suitability, accuracy, interoperability, compliance, security</td>
<td>suitability, accuracy, interoperability, compliance, security</td>
</tr>
<tr>
<td>Reliability</td>
<td>maturity, fault tolerance, recoverability, availability</td>
<td>maturity, fault tolerance, recoverability, availability</td>
</tr>
<tr>
<td>Usability</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Efficiency</td>
<td>time behaviour</td>
<td>time behaviour</td>
</tr>
<tr>
<td>Maintainability</td>
<td>changeability, stability, testability, reusability</td>
<td>changeability, stability, testability, reusability</td>
</tr>
<tr>
<td>Portability</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6.1: Quality aspects in ICT and business services

As can be seen, each quality subcharacteristic, which we defined as a quality aspect for ICT services, can be related to business services as well. This seems odd since ICT systems and human beings are dissimilar. However, the interpretation of these quality aspects depends on whether it concerns an ICT service or a business service. Let us clarify this using the quality aspect ‘stability’ as an example. ‘Stability’ implies the risk of new faults that arises after the providing service component has been modified i.e. after maintenance of the providing service component. Regarding ICT services, ‘maintenance’ holds the change of the internal details in order to improve the performance of the providing service component. This implies changes in the programming code of which the providing service component is composed. Regarding business services, a providing service component consists of human beings. Modifications in the providing service component are, for instance, the replacement of one of the human beings since that human being does not correctly perform the act for which he is responsible. Let us consider the restaurant example of Section 4.4 again, where the providing service component consists of human beings fulfilling the functionary roles.
‘order taker’, ‘dinner maker’ and ‘dinner serving’. The human being, who fulfills the role of ‘order taker’, could improperly perform and make faults during his work whereby he gets dismissed. The restaurant manager can then engage another human being to fulfill the role of ‘order taker’. Even though the new employee does not make the same faults as the former employee, he could also improperly perform by making another kind of mistake in his performance. Hence, this can be regarded as a new fault that arises after the providing service component has been modified. As can be seen, ‘stability’ can be regarded as a quality aspect for ICT services as well as business services, yet it is differently interpreted.

<table>
<thead>
<tr>
<th>Determinants</th>
<th>Quality Aspects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business integration quality</td>
<td>suitability</td>
</tr>
<tr>
<td>IS work quality</td>
<td>recoverability, changeability, stability, testability, reusability</td>
</tr>
<tr>
<td>IS user quality</td>
<td>accuracy, interoperability, compliance, security, maturity, fault tolerance, availability, time behaviour</td>
</tr>
</tbody>
</table>

Table 6.2: Determinants of business quality and quality aspects

Now the question arises to what quality dimensions these quality aspects can be related. Table 6.2 provides an overview of this. As can be seen, most of the quality aspects concentrate on IS user quality. This is quite logical, since we mainly concern the usage of the service rather than the development or maintenance of it. Nevertheless, we identified quality subcharacteristics of ‘maintainability’ as the quality aspects, which are related to IS work quality. These aspects are able to affect the service quality in such a way that it influences the stakeholder’s view on the quality of the service. Let us clarify this using the quality aspect ‘stability’ mentioned above. As can be seen, ‘stability’ focuses on modifications of the providing service component and thus the maintenance of the providing service component. Although this concerns the internal details of the providing service component, the effect of modifications can be noticeable to stakeholders in terms of new faults that might arise as the definition of ‘stability’ indicates. What also catches the eye is the fact that one quality aspect could be referred to business integration quality. As stated earlier in this chapter, business integration quality also partly concerns the operation level of the service.

As mentioned by Van Zeist et al., quality aspects can be incorporated into the functional specification [63]. This urges us to consider in what way quality aspects can be merged into the framework structure that we defined so far. Currently, we defined service provider information, terminology, service function information and service usage information as the areas of concern in the service specification framework. Quality aspects should be incorporated in service function information or service usage information. Since service function information concerns the production act of the service and the production world related to the service, it is logical to state that quality aspects that concern the quality of the production act of the service should be included as part of service function information. Likewise, what are included in service usage information are the quality aspects that indicate the quality of the communication between the consuming service component and the providing service component. On the basis of the research on the quality characteristics, we are able to out-
6.5 Identification of Quality Aspects

line the quality aspects that either belong to service function information or to service usage information in the table below.

<table>
<thead>
<tr>
<th>Category</th>
<th>Quality Aspects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service function info.</td>
<td>suitability, accuracy, maturity, fault tolerance,</td>
</tr>
<tr>
<td></td>
<td>recoverability, time behaviour, changeability, stability, testability, reusability</td>
</tr>
<tr>
<td>Service usage info.</td>
<td>interoperability, compliance, security, maturity,</td>
</tr>
<tr>
<td></td>
<td>fault tolerance, recoverability, time behaviour, changeability, stability, testability, availability</td>
</tr>
</tbody>
</table>

Table 6.3: Areas of concern and quality aspects

One can perceive some overlap between quality aspects in service function information and service usage information. Quality aspects that are included in service function information as well as service usage information have one particular thing in common, which is that they all concern faults which can occur regarding a service. These faults may relate to the actual service execution and to the communication between the consuming service component and the providing service component as well. We can again consider the quality aspect ‘stability’ in order to clarify this. Faults may arise concerning the communication between the consuming service component and the providing service component. For instance, the service request could not be received by the providing service component due to faults regarding the connection between the consuming service component and the providing service component. Hence, ‘stability’ should be mentioned in service usage information. Yet, faults can also raise regarding the actual service execution, the P-act of the service. The providing service component could perform the P-act of the service incorrectly due to faults. As a consequence, ‘stability’ should also be included in service function information.

Let us now consider how service levels should be determined. As mentioned in Section 6.4, a service level can be defined on the basis of the set of quality aspects that have been quantified for the service. Now it can be noted that these quality aspects are the ones that the service providing party has defined in service function information and service usage information. From these quality aspects, the service providing party can select the quality aspects with which he wants to distinguish different service levels. We could consider, for instance, a service providing party who denotes the quality of a particular service by including ‘suitability’, ‘accuracy’ and ‘fault tolerance’ in service function information and ‘stability’, ‘testability’ and ‘availability’ in service usage information. The service providing party wants to provide different service levels to anticipate on the various needs of stakeholders. In order to do this, he can pick one of the quality aspects defined in service function information or service usage information on which he wants to base distinct service levels. He could for instance decide to define different service levels on the basis of ‘availability’.

86
6.6 Evaluation

This chapter exposed the concept of ‘quality’ and how it can be related to ICT services and business services. In general, the concept of ‘quality’ can be defined as the characteristics of an entity that bear on its ability to satisfy stated and implied needs [38]. Regarding services, we indicated that each stakeholder has his own stated and implied needs. Therefore, we defined the concept of ‘quality’ regarding services as the characteristics of the providing service component and the service that it provides that bear on the ability to satisfy the stated and implied needs of each type of stakeholder.

Salmela’s view on quality enabled us to focus on the providing service component, the service that it provides and on the distinct types of stakeholders. Although Salmela’s model concerns quality of ICT systems, it was possible to apply it to ICT services, since ICT systems are actually providing service components. With this known fact in mind, we studied whether Salmela’s model was applicable as well if the providing service component has been composed of human beings. It was found that Salmela’s actually can be related to business services. This urges us to present an adapted version of Salmela’s view on ‘quality’ focusing on providing service components.

In order to make the concept of ‘quality’ operational, we had to find a method with which we could identify measurable quality aspects for the different quality dimensions of Salmela’s model. For this purpose, we used the Extended ISO-model. Due to its extensive set of quality aspects and technology independent approach, the Extended ISO-model was suitable to identify quality aspects for ICT services and business services. A thorough study resulted in the identification of quality aspects, which were able to denote the quality of an ICT service or a business service. Each quality aspect could be referred to IS work quality, IS user quality or business integration quality. In this way, it became clear to what types of stakeholders these quality aspects are of interest. The next step was to incorporate the quality aspects in the current structure of the service specification framework as depicted in Figure 5.4. Each quality aspect could be included in service function information or service usage information.

A service level is defined using the quality aspects in service function information or service usage information. These service levels can be offered to stakeholders for distinct costs. These costs can be charged in several ways such as a membership or on the basis of lease. We call the combination of a service level, the costs belonged and the ways for charging service usage a contract option. A service providing party can provide different contract options. As can be noted, contract options form another area of concern in service specification. Therefore, we should include the possibly contract options as one particular area of concern in the service specification framework, which we call service contract information.

Besides the identification of quality aspects for ICT services and business services, we were able to distinguish between stakeholders at an operation level and stakeholders at a management level. Stakeholders at an operation level are the service consuming parties we identified using enterprise ontology in the previous chapters. Stakeholders at a management level judge on whether a service meets the business requirements. We call these stakeholders management stakeholders.
6.6 Evaluation Service Quality

On the basis of the findings above, we can complete the structure of the service specification framework. This is shown in Figure 6.5. Regarding service consuming parties, service function information and service usage information is of great interest. These areas of concern describe what kind of functionality the service offers and how one can make use of the service. Regarding management stakeholders, service contract information and service function information is of interest. On the basis of service function information, management stakeholders are able to obtain a clear view of the functionality that the service offers and whether this functionality could meet certain business requirements. Service contract information provides valuable information regarding the investments that should be made for using the service.

As can be seen, service usage information, service function information and service contract information have their own specific set of stakeholders. Due to this distinction,
these areas of concern have their own specific terminology. Therefore, each of these areas of concern has its own section within ‘terminology’.
Part III

The Generic Service Specification Framework
Chapter 7

The Generic Service Specification Framework

This chapter gives an overview of the service specification framework developed as a result of our thorough study in the previous parts of the thesis. Throughout the research, we focused on the specification of services performed by ICT systems (ICT services) and services performed by human beings (business services). This resulted in a service specification framework, which supports the specification of ICT services and business services. Therefore, we call the service specification framework developed the Generic Service Specification Framework, which is illustrated in Figure 7.1.

Each section focuses on a particular area of concern of the Generic Service Specification Framework. Section 7.1 discusses service provider information, which contains the information needed about the one that takes the final responsibility for the service. Section 7.2 deals with service contract information, which contains service information in terms of costs and service levels. Section 7.3 concentrates on service function information, which comprises information that clarify the function of the providing service component. In terms of enterprise ontology, it describes the production act of the service and all other aspects that are involved in the production world. In Section 7.4, we describe service usage information, which makes clear how a service consuming party can make use of the service. Regarding enterprise ontology, this holds a clear description of the communication acts involved during service usage. In Section 7.5, we concentrate on terminology, which describes the terminology used in the service specification.

7.1 Service Provider Information

In Chapter 5, it clearly came forward that information about the service provider should be part of every service specification. The importance to contact the service provider was signified in Chapter 5, wherein we stated that a service consuming party could feel some urge to contact the provider concerning the service. We could consider, for instance, situations wherein a fault has arisen after service execution whereby the fault condition denoted to contact the service provider. Contact information is therefore included in service provider
Figure 7.1: The Generic Service Specification Framework
information. It comprises the name of the contact person and the phone number, email or address depending on the possibilities to contact the contact person. Next to this, service provider information should provide the provider name and a description about the service provider. A description may contain background information about the organization of the service provider and the domain wherein the organization operates (e.g. finance or logistics).

7.2 Service Contract Information

Service contract information includes one or several contract options from which the stakeholders can choose. A contract option consists of a particular service level and the costing details for using the service with this particular service level. Service contract information is of interest to management stakeholders. In Figure 7.2, we converted ‘contract option’. We discuss service level and costing details here below.

![Figure 7.2: Service contract information](image)

Service Level

As discussed in Section 6.4, a service level denotes the quality level of the service delivered. It is based on the quality aspects that are defined in service function information and service usage information. The service providing party might define different service levels in order to anticipate on the various needs of distinct stakeholders. Literature shows that service levels are generally determined on the basis of the service’s performance and availability [45] [30]. Yet, the service provider can decide himself which quality aspects he selects for defining service levels.

Costing Details

Costing details expose the price and charge approach belonged to that price. As mentioned in Section 6.4, there are different approaches on deducting the costs for using the service. The Transactional model comprises the simplest manner and can be regarded as a pay-as-you-go approach. It means that the service should be paid for each time the service is used. Within this approach, the service providing party could set up different service levels to be provided from which each of these levels costs differently. For instance, a service which is
7.3 Service Function Information

The Membership or Subscription model refers to the model which concerns periodic access of the service by the service consuming party. Usage costs are paid per period, which might be weekly, monthly or annual. Different types of memberships could be offered on the basis of the number of times the service will be used. For instance, the management stakeholder could select a weekly membership for using the service five times a week or ten times a week. The Lease or License model refers to the model that is based on charges by volume of transactions. In that case, the service is paid for the number of times the service is used.

7.3 Service Function Information

Service function information comprises information that clarifies what kind of service is provided. It is of interest to management stakeholders and service consuming parties as well. Management stakeholders need to know whether the service provides the functionality which meets the business requirements. Service consuming parties are interested whether the service supports them in performing their own activities. In terms of enterprise ontology, service function information describes the production act of the service and all other aspects that are involved in the production world. Figure 7.3 illustrates the aspects of service function information in more detail. We discuss them here below.

Function Description

Function description contains the elements that describe the function of the providing service component. As discussed earlier, the function of a providing service component is determined by the production act and can be described in terms of the relationships between the input and output variables. Function denotes the P-act of the service. Supported optional service calls indicates whether the providing service component supports the performance of a reject act or a cancel act as a service call. Input states the messages and data that should be handed over prior to service execution. Input data could be, for instance, the number of drinks and dinners sold in case of a turnover service. Likewise, output comprises the messages and data that result from the service execution. In Chapter 4, we made clear that business services and ICT services can be distinguished on the basis of their production type, which is either datalogical, infological or ontological. Hence, there are six different types of services. As part of the function description, we therefore define the production type of the service and whether it is a business or an ICT service in type. We could consider, for instance, ‘ontological business service’ as a service type.

Function Conditions

As previously explained, it should be clear to which conditions the input and output should hold in order to enable successful service execution. Preconditions state P-facts as conditions that should always hold prior to the execution of the service. As explained in Section 5.2, we should clearly discern production facta and production stata as types of production
facts. Since we can distinguish between datalogical, infological and datalogical productions, we can divide preconditions that are defined as production facts in the same manner. Datalogical preconditions comprise the P-facts that result from datalogical productions and have to be in existence prior to the service execution. Infological preconditions are P-facts that stem from infological productions and should be in existence before service execution. Likewise, ontological preconditions define P-facts, which are the result of ontological productions and have to be in existence prior to the service execution. Statum preconditions define P-facts which do not result from any kind of production.

Postconditions can also be classified into four types in the same manner as preconditions. Datalogical postconditions are the P-facts that should exist after service execution and that resulted from datalogical productions. Infological postconditions comprise the P-facts resulted from infological productions that have to be in existence subsequent to service execution. Likewise, ontological postconditions are the P-facts that have to be in existence after service execution and that resulted from ontological productions. Statum postconditions denote the P-facts that have to be in existence after service execution, but did not result from

Figure 7.3: Service function information
any type of production. Invariants are conditions that are preconditions and postconditions as well and should be complied with throughout the service execution.

Fault conditions define the faults that raise subsequent to service execution. They should be accompanied with messages that clarify the stakeholders how to handle the faults. These messages could, for instance, state that the service has to be invoked again or that the service consuming party should contact the service provider.

### Function Quality Aspects

In Chapter 6, we elaborated the different quality aspects that can denote the service quality. As can be derived from Section 6.6, we can relate quality aspects either to service function information or to service usage information. Quality aspects that indicate the quality of the production act of a service should be described in *function quality aspects*. As exposed in Section 6.6, these quality aspects are suitability, accuracy, maturity, fault tolerance, recoverability, time behaviour, changeability, stability, testability and reusability. Note that the service providing party is not under any obligation to measure all of these quality aspects. The service providing party should select the quality aspects to be measured that are applicable to the service concerned.

### 7.4 Service Usage Information

Service usage information clarifies how the service consuming party can successfully make use of the service as described in the service function information. This information is of interest to service consuming parties. Service usage information concerns how the service consuming party can communicate with the providing service component.

#### Location

Regarding SOA, the physical location of a providing service component should be hidden to service consuming parties. The consuming service component can connect to the service component by using what are termed location endpoints. Such endpoints are for instance URLs. Next to this, there are other options for specifying service location such as defining the TCP/IP hostname and port number [41]. Yet, the specification of locations depends on the communication protocol available [51]. Logically, this holds for ICT services. In terms of business services, it is usual to denote the physical location. Within an organization, the location might be specified as the department location of the providing service component such as for instance the second floor room 2.110. Yet, it is not needed to specify the physical location of the providing service component, since service consuming parties might find them on the basis of the phone number or email address.

#### Protocols

As discussed in Section 5.2.1, successful communication between the consuming service component and the providing service component is enabled using protocols. Protocols
define the rules governing the syntax, semantics and synchronization of communication. Regarding ICT services, typical examples of protocols are Internet Protocol (IP), Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP) [24] and Simple Object Access Protocol (SOAP). Regarding business services, one might think, for instance, of specific rules or manners that should be observed as part of the royal household. These protocols would probably differ from protocols in a bar.

**Usage Quality Aspects**

*Usage quality aspects* includes the quality aspects that denote the quality of the communication between the consuming service component and the providing service component. As can be derived from Section 6.6, these quality aspects are interoperability, compliance, security, maturity, fault tolerance, recoverability, time behaviour, changeability, stability, testability and availability. One might note that a number of these quality aspects were already mentioned as part of 'function quality aspects'. As explained in Section 6.6, this is due to the fact that these quality aspects focus on the actual service execution and the communication as well.

### 7.5 Terminology

As explained in Section 5.1.1, there should be syntactic and semantic understanding between the stakeholders and the service providing party concerning the service specification. Syntactic understanding is enabled by clarifying the form wherein the specification is presented. Logically, this is not stated in the specification itself. For instance, it makes no sense to state in the specification that the specification is written in English, while this statement itself is written in English. Semantic understanding can be ensured by using the terminology used by the service consuming party. This implies that the service providing party should have clearly in mind to whom the service specification is intended.

Each service specification should deal with management stakeholders, who judge the service, and service consuming parties, who make use of the service. Management stakeholders judge to what extent the service meet business requirements and whether the investments, in terms of charges for using the service, are worth using the service. Service consuming parties can be distinguished in distinct types, which depends on the service type as mentioned in Section 4.2 and 4.3.

The classification of the types of stakeholders makes it more clear what terminology should be used in the service specification. This is a step forward in enabling semantic understanding. It should be noted, however, that it is still insufficient to ensure semantic understanding. Although the same type of stakeholder basically uses the same terminology, it is not quite certain whether the terms used are also interpreted in the same manner. Hence, the terms used in the specification should be unambiguous and should therefore be clearly defined.

As can be seen, service contract, function and usage information refer to 'terminology' meaning that the terms used in these areas of concern should comply with the definitions given in 'terminology'. Since these areas of concern have distinct predefined sets of stake-
holders, it can be stated that the terminology used in these areas of concern differs between each other in any case. Therefore, it is possible to divide ‘terminology’ in three sections each focusing on one of these areas of concern. Regarding service contract information, the service provider should only deal with management stakeholders. Since service contract information provide information on the different contract options to enter into an agreement with the service providing party regarding the use of the service, it can be expected that service contract information mainly comprises of juridical terms. On the other hand, service usage information concentrates on the communication between the consuming service component and the providing consuming service component, which requires knowledge on whether the service is provided by human beings or ICT systems. Regarding ICT services, for instance, information is described using technical terms, which clearly indicate ICT systems as providing service components. Service function information is intended for management stakeholders and service consuming parties as well. This holds that the function of the providing service component should be described in such a way that it is understandable to all types of stakeholders. As mentioned earlier, service function information does not concern whether the service is realized and implemented using human beings or ICT systems. In other words, service function information does not consist of technical terms in any case. This makes service function information easily accessible for the distinct stakeholders. In order to be perfectly clear, graphical models may be incorporated as part of service function information’s section in ‘terminology’. One might think, for instance, of use case models which incorporate textual descriptions and graphical models for exposing functionality.

All in all, ‘terminology’ describes the terms used in the service specification. The set of terms depends on the type of stakeholders that are involved in the service. Yet, it is already clear that service contract information, service function information and service usage information have distinct sets of stakeholders. Therefore, it is possible to distinguish between three sections in ‘terminology’. In order to avoid any ambiguity concerning the terms, each term should be clearly defined in ‘terminology’.
Chapter 8

Case Study

The Generic Service Specification Framework adheres to a well-defined theory whereby the aspects in the framework could be identified. So far, we discussed service specification and its role in SOA only from a theoretical and scientific point of view. Yet, in the end, SOA is a practical concept which enables organizations to cope with changing business requirements wherein service specification takes an important part. Relying on a theoretical foundation does not guarantee that the framework will actually be useful in practice. Hence, it is important to verify the usefulness of the framework in practice. What we mean by ‘usefulness’ in this case is whether the area of concerns and service aspects in the framework fulfill the needs of service consumers regarding service information expected in service specification. Therefore, a case study was conducted at a large Dutch governmental organization which applied SOA to one of its departments.

This chapter discusses the case study, which was carried out. Section 8.1 outlines the research questions, which should be answered in order to verify the framework. In Section 8.2, we explain why we chose a case study as a manner to answer the research questions for verifying the framework. In order to perceive full answers on the research questions, a decent case study approach is needed, which is given in Section 8.3. Section 8.4 studies the service aspects that are included in service specification by the organization and the actors that are involved in service specification in a certain way. Using the knowledge and insights gained so far, we examined the Generic Service Specification Framework with two experts of the organization in Section 8.5. The chapter ends with an evaluation on the findings from the case study in Section 8.6.

8.1 Research Questions

The use of the Generic Service Specification Framework should lead to service specifications, which contain the service information needed by distinct stakeholders. The main question for researching the usefulness of the framework can therefore be defined as follows:

To what extent is the Generic Service Specification Framework useful for developing
8.2 A Case Study as Research Method

The search for answers to the research questions given above requires a well thought-out research method. Research methods are a set of skills, assumptions and practices which help the researcher to address a class of research questions [17]. Typical research methods are case studies, action research, or ethnography [10]. Regarding our purpose of this research and the research questions defined, a case study is the most suitable research method. A case study examines a phenomenon in its natural setting, employing multiple methods of data collection to gather information from one or a few entities (people, groups, or organizations) [11]. In terms of our research, we consider the phenomenon to be the service specification framework used in an organization, which applies SOA. By comparing this framework with our framework, we attempt to judge on the usefulness of our framework. In other words, we are interested to know how services are specified in an organization using SOA and how the Generic Service Specification Framework can be useful in that organization. As also mentioned by Benbasat et al., a case study is useful in the study of ‘why’ and ‘how’ questions [11]. Typical data collection methods are questionnaires, interviews, passive observations and document analyses.

8.3 Case Study Approach

8.3.1 Research Phases

In order to determine to what extent the Generic Service Specification Framework is useful, we proposed the research questions to be answered in Section 8.1. In order to have a clear and full answer on these questions, the case study consists of the following phases:

1. Perceiving an overall picture of the role of service specification in the organization:
   What is meant by the role of service specification in the organization is to what extent various actors are involved in the development and use of service specifications.

2. Studying the service specification framework used in the organization:
   By studying the framework used in the organization, we gain a clear insight into the

service specifications in practice that fulfill the needs of the distinct stakeholders?

In order to obtain a clear and full answer, we should study the following questions:

• What are the differences between the Generic Service Specification Framework and the service specification framework used by the organization?

• What service providing parties and stakeholders are involved in the development and use of service specifications?

• To what extent does the Generic Service Specification Framework fulfill stakeholders’ needs?
service aspects that are described in the organization’s service specification and why these aspects are included. This aids us in realizing which service aspects are needed and expected by stakeholders in service specifications in actual fact.

3. Evaluating the Generic Service Specification Framework by experts in the organization:
   In order to further determine whether the Generic Service Specification Framework is useful in practice, the framework should be examined by experts who have experiences with service specifications in practice.

8.3.2 Data Collection Methods

The completion of the phases above requires a precise selection of specific data that can clarify the role and use of service specifications and the actors involved. Several data collection methods exist as mentioned in Section 8.2. According to Benbasat et al., using multiple data collection methods lends greater support to the researcher’s conclusions [11]. However, the usage of data collection methods also depends on the possibilities in the organization for using these methods. Therefore, we collected data using document analysis and interviews as methods.

Several documents have been provided by the organization for analysis. By analyzing these documents, a clear insight has been gained into how the development and use of service specification is structured in the organization. More importantly, these documents include examples of service specifications. Next to document analysis, two experts, who are directly involved in the development and use of service specifications in the organization, have been interviewed. By interviewing these experts, we perceived the experiences of using service specifications in the organization and their judgments on the current framework used in the organization. More specifically, the interviewees clarified the documents provided and explained the development, use and management of service specifications. Another reason for the interview was to have a clear examination of the Generic Service Specification Framework by the experts. With their knowledge and expertise, they commented upon the Generic Service Specification Framework and examined the framework on its usefulness, which formed a solid ground for the verification of the specification framework.

8.4 Service Specification in the Organization

In order to have a clear view on how services are specified in the organization, we examined the organization’s service specification framework and the actors that are involved in the development and use of service specifications.

8.4.1 The Organization’s Service Specification Framework

In order to obtain an overall picture of the role of service specification in the organization, it is significant to know what the organization actually means by ‘service specification’. In
addition, knowing their notion on ‘service specification’ clarifies how service specifications
are used in the organization and eventually whether our framework would fit into their
manner of specifying services. During the meeting with the experts in the organization,
documents provided showed that the organization does not view service specification from
a broad perspective as we do. Next to this, it was found that the concept of ‘service’,
apparently was differently interpreted in the organization. A service is regarded as a set of
coherent operations, which fits into the notion of services and operations used by WSDL.
It goes without saying that we regard such an operation as a service. However, this known
fact implies that the organization regards service specification as the specification of a set
of coherent operations. In this section, we will use the concept of ‘service’ as defined by
the organization.

In one of the documents provided was further stated that service specification in the
organization is lightly based on WSDL. A service specification in the organization consists
of an ‘interface’, which defines the set of operations and its input and output messages, and a
‘binding’ section, which indicates the location of the service implementation. Additionally,
a ‘type’ section in the service specification defines the model for describing the syntax
of the data exchanged during service usage. Input and output messages should refer to
definitions in the syntax model. Figure 8.1 gives a representation of the structure of a
service specification in the organization.

At first sight, it seems that service specification is only viewed from a technical perspec-
tive on the basis of WSDL. Yet, there are other documents in the organization that clearly
contribute to the specification of a service, but they are not regarded as a service specifi-
cation according to organization’s notion on ‘service specification’. The organization makes
use of use case models in order to describe the functional aspects of a service. A use case
is developed for each particular operation of the service. Such a use case includes an activ-
Case Study 8.4 Service Specification in the Organization

*Activity diagram*, which visualizes the activities performed regarding an operation. For instance, the operation ‘register person’ in the organization consists of activities such as ‘validate person’ and ‘establish subscription’. The activity diagram is further clarified within the *process description*, which describes the different activities. Next to this, each use case states the purpose and the pre- and postconditions of an operation. Use case models are developed by requirements specifiers and are intended to give business analysts a clear insight into the service.

Each service has a *contract*, which contains the Service Level Agreement (SLA) and the policies regarding the service. A SLA includes the agreements made between the service providing party and the stakeholder concerning the service level (e.g. availability). Policies comprise the security aspects of the service. Basically, only one contract is offered for each particular service, which holds that each stakeholder makes use of the same SLA. The adjustment of the SLA or the policies leads to a new version of the contract, which is also available to all stakeholders. Since a contract represents a set of operations, a change in one of the operations already requires a new version of the contract. This is regarded as a disadvantage in the organization.

The organization implemented a Canonical Data Model (CDM) in order to define the common data that are shared between different parties in the organization and external parties. The data in the CDM are subdivided into distinct domains on the basis of a particular party. Parties are, for instance, the departments of the Dutch government. Each domain contains definitions of the data that is exchanged with that particular party. In that way, the organization aims at semantic understanding concerning the data between different parties. Terms used in service specifications have to refer to definitions in the CDM. Note that the type section of the service specification as depicted in Figure 8.1 includes the domain of the CDM used.

It can be seen that ‘service specification’ in the organization encompass a greater whole than just the aspects shown in Figure 8.1. Figure 8.2 depicts an overview of the service specification including the other aspects which also involves service specification. For the sake of convenience, we call this whole the service specification framework of the organization.

### 8.4.2 Actors Involved in Service Specification

During the interview, it became that distinct actors are involved in service specification through the whole organization. These actors are *business analysts, architects, requirements specifiers, developers, information management* and the *service competence center*.

Business analysts validate whether a service is useful to be included in a certain business process. In order to do this, they make use of the use case models. As a consequence, they can be regarded as stakeholders. More specifically, they are management stakeholders since they do not use the service themselves.

Architects indicate where services should be used in the organization on the basis of the ICT-architecture. They decide which applications are suitable to be encapsulated as services and what kind of services should stem from these applications. They set out the directions for the specification and implementation of services. Next to this, they are currently creating awareness regarding SOA in the organization.
Requirements specifiers further elaborate on the ideas of the architects and develop the use case models for the services. The use case models only concern with the functionality of the service i.e. what the service actually does. The models are intended for business analysts and therefore requirement specifiers should have a clear view on the requirements of business analysts. The requirements specifiers are actually the ones who design the whole service, since the use case models contain all functional aspects of the service. The development of the use case models happens in an iterative way. The use case model is submitted to the service competence center (SCC) for validation after each iteration. This point will be elaborated on later.

Developers implement services and their interfaces. When the interfaces are implemented, they should use definitions from the CDM. If the service is complex, a technical specification of the service is developed prior to the implementation.

The information management is responsible for the CDM. They attend to make people in the organization using the same terms and definitions when data is exchanged.

The service competence center is a group practice of information management, functional management, architects, developers, testers and other people who have any knowledge on methods and techniques. It defines the guidelines and agreements to which aspects of service specification should hold. One might think, for instance, of guidelines which state when something can be called a ‘service’ or an ‘operation’. The service competence center verifies whether the use case model has been set up according to the guidelines and agreements.

8.5 Verification of the Generic Service Specification Framework

During the meeting, we also discussed on the Generic Service Specification Framework. Generally, the experts were very positive about the specification framework. They did not
note, as they say, any odd things in the framework. They were very interested in the final version of the specification framework. Here below we give a more detailed overview of the verification of the Generic Service Specification Framework. Each subsection will discuss a particular area of concern of the framework.

8.5.1 Service Provider Information

The two experts both agree on the fact that information of the service provider should be available to stakeholders. They agree on the fact that stakeholders should be able to contact the service provider in case of problems or other issues regarding the service. The experts proposed to make a clear distinction between internal and external stakeholders. Internal stakeholders are stakeholders within the organization itself. In that case, it is sufficient to define the function of the contact person together with its email. The service provider can be regarded as the owner of that particular service in the organization. External stakeholders are stakeholders from external organizations. From that point of view, the organization is regarded as the service providing party delivering the service to external stakeholders. It is not recommended to provide personal details of the contact person towards external organizations.

Besides the distinction between internal and external stakeholders, the two experts suggest to define the role or function of the contact person instead of his name. It is more likely that the name of the contact person will change more often due to reorganizations than a specific function or role within the organization. As one of the experts stated, the more the service specification depends on the organization, the more often you have to change the service specification if the organization changes.

8.5.2 Service Contract Information

Service contract information can be compared to the contract, which contains the SLAs and policies of a particular service used in the organization under study. The two experts stated that it is important to expose the different service levels that can be provided. The organization uses SLAs as a set of agreements that have been made between the service providing party and the stakeholders concerning the service level that is provided. According to the experts, a SLA should contain the availability of the service and its response time. Additionally, the service providing party and the stakeholders could make agreements on the reliability of the service. The policies concern with the security aspects of the service.

As mentioned above, only one contract is designed for a particular service and several contract versions could exist if one of the operations has been changed. This was stated in one of the documents received from the organization. However, the two experts indicated that contracts are basically defined in the organization for each particular internal or external party. Such a specific service specification is appropriate for this organization, since the internal and external parties that use the service are already predetermined. A service in the organization will actually be provided if the service providing party and the stakeholders agree on the service level. Within our framework, we rather define the service specification to cater for a broad public without having prior knowledge of the stakeholders. Because
of this, defining specific contracts as the organization under study would not fit within the function of our framework. This has been acknowledged by the two experts.

Another clear difference between both frameworks is the inclusion of charges for using the service. Currently, charges have not been defined yet in the organization for using the service. However, the experts acknowledged that charges should be part of the specification and they are planning to include this into their own framework. Internal parties will then be charged as well for using services. Before SOA was implemented in the organization, each process existing from several applications had one particular owner who also owns the underlying applications. Due to SOA, applications will also provide services to other applications. Hence, a process now consists of applications of different owners. A particular process owner who makes use of a service from a different owner should be charged for using that service. The two experts regard settling charges as a necessity and as something which is quite logical. According to them, a service owner does not want to fully foot the bill of the service development by himself, while other parties will make use of his service as well.

### 8.5.3 Service Function Information

Service function information covers the functional specification of the organization’s framework, which is the use case model. The two experts regarded the service aspect ‘Function’ as an important aspect, since it states what the service actually does. ‘Type’ was a service aspect which was unknown to these experts and which was not specified as part of the service specification in the organization. The experts stated that they do not make any distinctions between different types of services. Since we made the distinctions of service types on the basis of enterprise ontology, it was not quite clear to the experts what rationale was behind including these service aspects in service specification. By explaining them that a service type actually states to what extent the service is enterprise-specific and thus the reusability of a service, they acknowledged the significance of defining the service type.

The importance of defining pre- and postconditions also clearly comes forward since both frameworks enable to specify them. Regarding fault conditions, the two experts clearly denoted that a fault condition should always accompany with messages so that the service consuming party can properly react to the fault arisen without having to contact the service provider. If it concerns a technical fault, the service provider should solve this problem as the experts said. In that case, it is impossible for the service consuming party to react properly to the fault arisen.

Our framework differently deals with service quality than the organization’s framework. Requirements specifiers using the organization’s approach incorporate all quality aspects to be specified in the functional specification of a service. They do not distinguish between quality aspects that are related to the function of the providing service component or the communication with the providing service component. The two experts understand this distinction, yet, they have not encounter any problems concerning the current way specifying quality aspects. Regarding our framework, the experts agreed on the fact that the Extended ISO-model is a suitable model to be used as a basis for identifying quality aspects.
8.5.4 Service Usage Information

Regarding service usage information, we did not discuss much about the service aspects in this area of concern. The experts found it quite logical to define the service location and protocols used. Yet, more explanation was needed concerning ‘usage quality’, since they do not make any distinction between quality aspects concerning the functionality of the service and quality regarding the communication with the providing service component. We discussed the quality aspect ‘interoperability’, which is a part of ‘usage quality’, so that the distinction between function and usage quality became even clearer.

8.5.5 Terminology

The two experts realize the importance of ‘terminology’ as an area of concern in the specification. This idea on a ‘terminology’ area has already been implemented in the organization by establishing a CDM. Contrary ‘terminology’, the organization centralized the CDM being available to all parties within the organization that concern data exchange. With this in mind, it was quite interesting to reconsider the inclusion of a ‘terminology’ area in the Generic Service Specification Framework. A centralized management of common used data would be more efficient and would ensure ambiguity of terms more certainly, since definitions have not to be specified for each particular service specification. However, contrary to the framework used in the organization under study, we aim at service specifications used by stakeholders within and outside the organization as well without the fact that a service providing party has to know the stakeholder beforehand. The organization under study provides the services to different internal parties and external parties that are predefined. Regarding each party, a particular domain in the CDM has been defined containing the data that is exchanged with that particular party. Regarding our specification framework, a worldwide CDM should have to exist then. Since such a data model does not exist, terms used in the service specification should be defined in a separate area of concern within the specification.

8.6 Evaluation

This chapter discussed the case study, which was carried out at a large Dutch governmental organization in order to verify the Generic Service Specification Framework on its usefulness in practice. In order to verify the framework successfully, we had to investigate what kind of service information is actually specified in practice and what actors make use of service specifications. Since the organization gave the opportunity to analyze several documents and to interview experts on this topic, we were able to gain a clear insight into the role of service specification in practice.

What we first noticed was that ‘service specification’ is viewed from a technical perspective in the organization. However, a further analysis on the documents made clear that other documents were available which were not regarded as ‘service specification’, but should actually be counted as ‘service specification’. Use case models, contracts and CDM have their own particular purpose on describing services. For instance, use case models are used by business analysts in order to know the function of the providing service component.
It was found that the Generic Service Specification Framework and the organization’s framework were in general comparable to each other. Each area of concern could be referred to one of the aspects of the organization’s framework. However, the content of the areas of concern differ at some points. We could consider, for instance, service function information which is comparable to the use cases in the organization. Such a use case focuses on the activities that compose the operation using an activity diagram and a process description, while service function information does not concern the underlying activities of a service.

Another important note to mention is that it has also been verified that the Generic Service Specification Framework focuses on the different types of stakeholders. As the two experts stated, business analysts, architects, and developers should be regarded as the stakeholders in the organization. Business analysts make use of the use case models and the contracts in order to judge whether the service really fits into their needs.

In all, it can be noted that the Generic Service Specification Framework has been successfully verified in practice. According to the experts in the organization, the Generic Service Specification Framework enables to provide service information needed by the distinct types of stakeholders. The experts stated that they were very interested in the final version of the Generic Service Specification Framework.
Part IV

Conclusions
Chapter 9

Contributions

As part of the conclusions, we will enumerate the main contributions of the project.

At first, a service specification framework has been developed on the basis of a well-defined and proven scientific theory. Regarding the SOA domain, this is an approach which deals with service specification in a different way than approaches that currently exist. What is well-founded by adhering to a theory is the inclusion of each aspect in the Generic Service Specification Framework.

Secondly, enterprise ontology gave the opportunity to view the concept of ‘service’ from an ontological perspective. It enabled us to define the concept of ‘service’ without considering how a service is realized or implemented. From this point of view, we were not only able to have a clear view on services implemented using ICT systems, but also on the realization of the concept of ‘service’ using human beings. These distinctions of a service facilitated the development of a service specification framework, which can be applied to ICT services and business services as well. With this, the framework stands out from other specification frameworks within SOA, which only focus on ICT services.

Thirdly, enterprise ontology gave us a clear insight into the capabilities of human beings and ICT systems in performing acts and the relations between them. It gave the opportunity to clearly denote the differences and relations between business services and ICT services. The view of enterprise ontology on ICT systems and how they support human beings in their activities provide a means to judge the business value of an ICT service i.e. to what extent an ICT service support business services performed by human beings. Since both types of services can be specified using the Generic Service Specification Framework, the framework makes business people easier to make these judgements. Each particular service aspect of ICT services and business services can be compared with each other.

Finally, enterprise ontology enabled us to develop a framework having a proper balance between business-oriented and technical-oriented service information, which concentrates on stakeholders at a management and an operation level. While service function information provides valuable information for stakeholders at both levels, service contract information and service usage information focus on management stakeholders and service consuming parties, respectively. Neither business-oriented service information nor technical-oriented service information has the upper hand in the Generic Service Specification Framework.
This research focused on the development of a service specification framework. By applying the enterprise ontology as a theoretical foundation to the framework, we differently dealt with service specification than other approaches on service specification. We used a theoretical foundation which is basically unrelated to service specification and to the SOA domain. Therefore, it was needed to study to what extent enterprise ontology could provide adequate ground for specifying services in SOA as was indicated in the introduction of this thesis. In order to make use of enterprise ontology to its full capacity, we had to investigate this concept and its theory in great detail. This means that each element of enterprise ontology which could be of any relevance to the thesis had to be discussed. Only in that way, it was possible to enable a full understanding of the contribution of enterprise ontology to the development of a service specification framework.

Besides the use of enterprise ontology for developing the framework, it was of great significance to have a clear understanding of the concept of ‘service’. In order to have a clear insight into the service aspects that should form the framework, it was first important to know what a service actually implies. Here, the value of enterprise ontology already became clear, since the axioms and the organization theorem of enterprise ontology were all of great relevance for constituting the concept of ‘service’. The fact that ‘service’ was identified as part of a transaction also indicated the contribution that enterprise ontology could provide in developing the framework, since transactions can be regarded as the main elements in enterprise ontology. Enterprise ontology further enabled to describe the concept of ‘service’ without considering its realization and implementation using human beings and ICT systems. As could be noted further in the research, this known fact laid the foundation of the Generic Service Specification Framework. Next to this, the determination of the concept of ‘service’ paved the way for new insights into aspects that involve services. Composite and atomic services have been studied carefully and led to the determination of the difference between ‘service’ and the system that provides a service, the service component. The different types of service consuming parties were identified in a well-considered way using the organization theorem. In all, what enterprise ontology actually showed was that understanding the concept of ‘service’ needs a thorough study in itself. In order to comprehend the concept of ‘service’ to its fullest, you should also have a clear understanding of its realization and implementation and the different service types that stemmed from that
Conclusion

Understanding the concept of 'service' created a roadmap for further research in the identification of the role of service specification according to enterprise ontology. The distinction between C-acts and P-acts that form a service provided ground for the formation of the specification framework. It gave a clear rationale for the difference between service function information and service usage information i.e. a description of what the service offers and information about how to use the service. By uncovering the construction of C-acts and P-acts using the distinction axiom, we were able to build up the main structure of the service specification framework and its contents. The necessity of service provider information, pre- and postconditions, invariants and fault conditions could be argued using the theory. However, during this research, we found out the limitations of enterprise ontology for developing the framework. Enterprise ontology did not give the possibility to identify quality aspects of services. The specification of quality aspects of services are of great importance and cannot be lacked in service specifications. Yet, enterprise ontology set out the directions for finding a mean to identify service quality aspects. The thing was to find a mean, which enabled us to incorporate quality aspects into the structure of the framework at that moment and which fitted into the notion on 'service' and 'service specification' we acquired. This search led us to an interesting view on 'quality' of Salmela. Salmela distinguished IS quality, which concern quality aspect on the development and use of ICT systems, and business integration quality, which indicated the costs, in terms of investments, and benefits for using ICT systems. Here, enterprise ontology’s limitation for the specification framework came clearly forward; it only focuses on the operation of an enterprise. What fall out of the enterprise ontology scope are management issues regarding the regulation of the operation of an enterprise. Defining costs for using services and the service’s benefits should be regarded as a management issue.

The Extended ISO-model was further used in order to identify the quality aspects of a service that form IS quality. An important reason for using this model was the fact that it provides an extensive overview of aspects which could somehow or other indicate the quality of a service. Yet, the decisive factor here was that quality aspects were described independently from technology or any other environment. This made the Extended ISO-model also suitable for identifying quality aspects for business services. A thorough study resulted into the identification of quality aspects that state something about the quality of the function of the providing service component and about the quality of the communication with the providing service component. The search for means to define the costs for and benefits of using services led to the introduction of a new area of concern in the service specification framework. Service contract information provides an overview of distinct contract options which can be made between the service providing party and the stakeholders. A contract denotes the service level provided and its charges. Since the interpretation of benefits of using a particular service depends on the stakeholders, providing distinct service levels is the best way to anticipate on the needs of stakeholders. Charges can be defined using distinct possibilities such as memberships or lease.

So, as questioned in the introduction, how can the enterprise ontology theory provide ground for specifying services in SOA? Concluding from the above, the main contribution of enterprise ontology is that it enabled us to develop another way of thinking about the
Conclusion

correlation of service. This way of thinking laid the foundation of the identification of service aspects to be specified. Although it did not provide any opportunities to identify quality aspects, it clearly was of great value for identifying quality aspects.

From a theoretical point of view, the Generic Service Specification Framework includes the aspects that are relevant for stakeholders in order to select and use services. Yet, this did not guarantee that the framework was usable and useful in practice. Consequently, a practical case study was needed in order to investigate the usefulness of the framework in practice. The case study was carried out at a large Dutch governmental organization, where SOA was implemented in one of its business units. Two experts, who were directly involved in the implementation of SOA, took a glance on the framework. It was found that the Generic Service Specification Framework on the whole showed resemblances to their own specification framework. They regarded the Generic Service Specification Framework as useful. According to them, the framework covers the service aspects that are relevant for different types of stakeholders.

So, how does the service specification framework support stakeholders in service selection and service usage? This question was asked in the introduction. The Generic Service Specification Framework supports stakeholders since it clearly distinguishes five areas of concern. Each area of concern refers to a certain purpose for using a service specification framework. For instance, service function information comprises relevant information needed to select a service, whether this selection is based on the functionality or on the quality of the service. Service usage information contains important information on using the service. It includes relevant information needed in order to communicate with the providing service component. From a theoretical point of view, the inclusion of each area of concern and its content has clearly been argued. In practice, the framework’s value has been proved during the case study. It is, however, not as persistent yet as the theoretical view on the framework implies. Being a great value within one particular organization does not necessarily mean that this also holds for other organizations.

Now we have provided clear answers to the research questions, we consider the project goal given in the introduction:

*Develop a service specification framework using a robust theoretical foundation which outlines the service properties that should be part of every service specification in order to support service selection and usage by stakeholders*

The above and the framework itself actually already stated whether the goal has been achieved. In all, we can conclude that we have successfully developed a service specification framework on the basis of a well-defined theory, which supports each type of stakeholder. A great factor for this success was the theory that was used for developing the framework. It gave the opportunity to take another glance on ‘service’ and ‘service specification’. Moreover, it even provided another way of thinking about services and service specifications.
Chapter 11

Future Work

On the basis of enterprise ontology, a framework has been developed providing a clear structure of the service specification and the aspects to be specified. Each aspect has a clear rationale for its inclusion in the framework. Nevertheless, the specification framework is far from being a complete solution for specifying services.

A clear understandable specification is not fully achieved by just defining what service aspects have to be specified. Besides answering the ‘what’-question i.e. what service aspects should be specified, it is of great significance to answer the ‘how’-question as well i.e. how these service aspects should be specified. Understanding concerning the content of a service specification relies on syntactic and semantic understanding of the content between the service providing party and the stakeholders. Semantic understanding is enabled by including ‘terminology’ as an area of concern in service specification. Syntactic understanding is enabled by clarifying the form wherein the content of the service specification is presented. Logically, this is not stated in the specification itself, yet, it is clearly of great importance in order to fully support service providing parties and stakeholders. Therefore, it is needed to research which specification standards are appropriate to specify each particular service aspect of the specification framework.

If one takes a glance on standards in SOA for specifying services, one would find a wide variety of standards each specifying a particular service property. Yet, there is clearly still need for other specification standards and several consortia, such as the World Wide Web Consortium (W3C) and the Organization for the Advancement of Structured Information Standards (OASIS), are trying to fulfill this need by developing other specification standards. Yet, the other side of the coin is that this development has lead to a morbid growth of specification standards, which actually make service specification even more complicated. As already mentioned by Walkerdine et al., it is not realistic to expect that service providing parties will support all of the specification standards existed [66].

As can be seen, the assignment of standards can make the specification a more powerful tool for specifying services. Yet, an accurate study is needed in the field of specification standards in order find the most appropriate standards that fit into the Generic Service Specification Framework. It is of great significance to think outside the box. In other words, one should not restrict himself to the SOA domain in finding suitable specification standards. By thinking outside the box, one extends his research domain in finding suitable specification standards.
Future Work

Another main focus for future research is the application of the framework to business services. During the verification of the framework, we merely concentrated on the specification of ICT services in the organization, since we aimed at the support of stakeholders in SOA. Yet, it is of great interest to investigate to what extent business services can be specified in practice using the framework. When business services and ICT services are specified using the framework, judging to what extent an ICT service supports a business service can be simplified, since the same service aspects are specified. Therefore, research regarding the application of the framework to business services can provide valuable insights into business service support using ICT services. In addition, this research could further confirm the power of the Generic Service Specification Framework as a tool for service specification.
Abbreviations

CDM Canonical Data Model
CIAO Cooperation and Interoperability - Architecture and Ontology
DEMO Design and Engineering Methodology for Organizations
EEMCS Electrical Engineering, Mathematics and Computer Science
HTTP Hypertext Transfer Protocol
ICT Information and Communication Technology
IEC International Electrotechnical Commission
IP Internet Protocol
ISO International Organization for Standardization
OASIS Organization for the Advancement of Structured Information Standards
SCC Service Competence Center
SeCSE Service Centric System Engineering
SLA Service Level Agreement
SOA Service Oriented Architecture
SOAP Simple Object Access Protocol
TCP Transmission Control Protocol
TPM Technology Policy and Management
TQM Total Quality Management
UDDI Universal Description, Discovery and Integration
**Future Work**

**URI** Uniform Resource Identifier

**URL** Uniform Resource Locator

**W3C** World Wide Web Consortium

**WSDL** Web Service Description Language

**WSLA** Web Service Level Agreements

**XML** Extensible Markup Language
Bibliography


Appendix A

Identification of Quality Aspects Using the Extended ISO-model

This chapter studies the main quality characteristics and the subcharacteristics of the Extended ISO-model and whether they can be related to ICT services and business services. The Extended ISO-model explains the quality (sub)characteristics in terms of software products, which can be viewed as ICT systems. As we already mentioned, we can regard ICT systems as providing service components. Due to the nature of Extended ISO-model, it can also be applied to human beings as providing service components. Nevertheless, the Extended ISO-model still concerns ICT systems. Therefore we study the quality (sub)characteristics as follows: At first we examine whether a particular quality subcharacteristic can be used as a quality aspect for ICT services. Then, we investigate whether it can also be related to business services. If necessary, we clarify our findings using examples. In these examples, we make a clear distinction between the providing service component and the service provider as parts of the service providing party in order to clearly state the one that does the actual service performance and the one that takes the final responsibility for the performance.

A.1 Functionality

The Extended ISO-model defines *functionality* as follows: ‘a set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs’ [63]. In terms of services, ‘functionality’ clearly focuses on the services that providing service components deliver. Logically, we should further study which quality subcharacteristics can be assigned as quality aspects for ICT services or business services. ‘Functionality’ embraces *suitability, accuracy, interoperability, compliance, security* and *traceability* as subcharacteristics. Here below, we discuss these subcharacteristics in more detail.
A.1 Functionality Identification of Quality Aspects Using the Extended ISO-model

A.1.1 Suitability

Suitability is an attribute of software that bears on the presence and appropriateness of a set of functions for specified tasks [63]. Since ‘suitability’ concerns the set of functions of software, we are able to relate this quality aspect to ICT services. In terms of ICT services, suitability holds the extent to which an ICT service can support people in the enterprise in their activities.

There are several indicators for measuring ‘suitability’ as a quality characteristic. For instance, coverage ratio indicates the percentage of desired functionality that is actually present in the software product [63]. We can interpret this in terms of ICT services as the percentage of desired functionality of the ICT service that the providing service component provides to the people in the organizations of the enterprise. Another indicator of suitability is the scaled coverage percentage. It denotes the percentage of preferred functionality that is actually present in the software product, indexed according to the relative importance of the required functionality for the enforcement of the business activity [63]. As ‘coverage ratio’, we can use this indicator for measuring ‘suitability’ in ICT services. In that case, it can be regarded as the percentage of preferred supporting functionality that the ICT service provides to the people in the enterprise. In other words, it measures to what extent the ICT service can actually take over the acts performed by the people in the enterprise.

It is now interesting to find out whether we can denote ‘suitability’ as a quality aspect for business services. If we regard the previous paragraphs, ‘suitability’ concerns the extent to which an ICT service can support the people in the enterprise in their activities. From a general perspective, the support of people in their activities does not need to be just based on ICT services. It goes without saying that people can be supported in their activities by other people in the enterprise. Therefore, we can regard ‘suitability’ as a quality aspect of business services. In terms of business services, ‘suitability’ indicates to what extent the business service supports other people in the enterprise. Let us clarify this using an example. We consider a service ‘retrieve financial reports’, which is offered by an archivist. An accountant makes use of this service by requesting the financial reports of 2006. He needs these reports in order to compare them with the financial reports of 2007. The service ‘retrieve financial reports’ is a typical datalogical business service. It goes without saying that this service supports the accountant in performing his own activities. The example underlines that we can consider ‘suitability’ as a quality aspect for business services. Both indicators for ‘suitability’, presented in the previous paragraph, can also be applied to business services. ‘Coverage ratio’ indicates whether the desired functionality can be delivered by the business service. For instance, if the accountant requests the archivist for the financial reports 2006, the service provided by the archivist does not fully offer the desired functionality if the archivist can only hand over the financial reports of the first three months of 2006. The same holds for ‘scaled coverage percentage’ as an indicator for suitability. The archivist can hand over the financial reports of 2006 to the accountant. However, if these reports are unimportant to the account, the scaled coverage percentage would be low.

Let us consider Salmela’s view on quality, which was discussed in Section 6.1. As can be seen from above, ‘suitability’ quantifies to what extent the service supports the service consuming parties in the performance of their activities. Therefore, ‘suitability’ can be
related to business integration quality. Regarding the areas of concern of the service specification framework, it fits into service function information, since it indicates the quality of the function i.e. the P-act of the service.

A.1.2 Accuracy

‘Accuracy’ embraces attributes of software that bears on the provision of right or agreed results or effects [63]. Differently said, it indicates to what extent the ICT system provides the right or agreed results, when an end-user makes use of the ICT system. In order to measure ‘accuracy’, the right or agreed results should be defined in some kind of document whereby the end-user can check whether the ICT system actually provide the results as defined in that document. From a service consuming party’s perspective, it is interesting to know the degree of accuracy of an ICT service. The more the ICT service provides incorrect or unagreed results, the less the ICT service is accurate. Regarding ICT services, right results can be defined in the service specification for instance in terms of postconditions (see Section 5.1.1).

In order to determine the degree of accuracy, we can make use of the indicator failure ratio. ‘Failure ratio’ comprises the ratio of incorrect processed transactions to the total of presented transactions [63]. Note that ‘transaction’ does not relate to a transaction in enterprise ontology. In order to determine the failure ratio of an ICT service, one should execute the service a number of times. Then, one should determine the number of service executions that did not provide the right or agreed results. With these two values, the failure ratio can be calculated. Other indicators for ‘accuracy’ focus on the distinct functionalities that a certain ICT system can provide. Yet, since we regard a service is one particular functionality, these indicators are not suitable for measuring ‘accuracy’ in ICT services.

Now the question arises whether we can consider ‘accuracy’ as a quality aspect for business services. Regarding business services, human beings perform the service and provide the service result to the service consuming party. Logically, it can be stated that human beings can also somehow provide incorrect results. Let us clarify this with the example of the accountant and the archivist. If the accountant requests the financial reports of 2006 and the archivist actually delivers the financial reports of 2006, we can regard this as a right service result. However, if the archivist delivers the financial reports of 2005 instead of 2006, this may be regarded as an incorrect service result. The more often the wrong financial reports are delivered, the less accurate the archivist is. Clearly, we can also make use of the indicator ‘failure ratio’ for measuring accuracy in business services.

It can be noted that ‘accuracy’ provides information on service quality from a service consuming party’s perspective. It focuses on the quality of the service result. ‘Accuracy’ can therefore be referred to IS user quality as part of Salmela’s view on quality and to service function information as part of the generic service specification framework.

A.1.3 Interoperability

The Extended ISO-model defines ‘interoperability‘ as the ability to interact with specified systems [63]. Regarding ICT services, this holds the extent to which a providing service
A.1 Functionality Identification of Quality Aspects Using the Extended ISO-model

Component can interact with consuming service components by providing ICT services to each other. The Extended ISO-model adheres interoperability to a number of indicators such as the matched data format ratio, matched interface ratio and the effort per interaction. ‘Matched data format ratio’ indicates the ratio of data formats matched to those of the other system in the interoperation [63]. As can be derived from Section 2.4 and 4.3, ICT systems should agree on the form aspects of data sent to each other in order to make interactions possible. Therefore, ‘matched data format ratio’ can be useful for measuring interoperability of ICT services. In the SOA domain, the Extensible Markup Language (XML) is commonly used as data format for services. The matched interface ratio is the ratio of interfaces matched to those of the other systems in interoperation [63]. An interface is regarded as an intermediary in order to communicate with other systems. Systems with matched interfaces are able to communicate with each other. Hence, a higher value of the indicator ‘matched interface ratio’ is preferable. There are several ways for defining interfaces for ICT services. One might think, for instance, of the Web Service Description Language (WSDL) [15], which is the most well-known language in SOA for defining interfaces. However, interfaces may be defined using other languages such as the Interface Description Language (IDL) of the Common Object Request Broker Architecture (CORBA) [26]. As the term already suggests, ‘effort per interaction’ indicates the effort needed to realize interoperability per unity size of interoperability [63]. In other words, it measures the effort needed to realize interoperability in ICT services and to what extent ICT services are actually interoperable.

One might note that ‘interoperability’ should also be viewed as an essential quality aspect of business services. Regarding the performance of business services, the service consuming party as a human being interacts with the service providing party as a human being or with the providing service component in case of delegation. Recall Section 2.4, which gave an overview of the three distinct human abilities that play a role in the performance of C-acts. The ability to interact relies on the forma and informa ability if we consider ICT services. Since interactions between human beings concern the forma, informa and the performa ability as well, the realization of interoperability in business services is more complicated. Besides syntactic and semantic understanding, there should also be social understanding between human beings in interaction. Let us, for instance, again consider the archivist that hands over the financial reports of 2006 to the accountant. A request of the accountant can be performed by speaking (the forma ability). Furthermore, the request should be formulated in such a way that the archivist can interpret it (the informa ability). Next to this, the accountant and the archivist should both engage into commitment in order to consume and provide the service (the performa ability).

If we consider the indicators used for measuring ‘interoperability’, it can be noted that the ‘matched data format ratio’ can be applied in order to measure interoperability of business services. For instance, there is a match if the service providing party and the service consuming party use spoken word for interaction. If we consider the example of the accountant and the archivist, one might think, for instance, of the accountant requesting the financial reports by telephone. In that case, both the accountant and the archivist should communicate together by speaking. There is no matched data format if the archivist could only deal with requests by email in this case. What is actually more difficult to answer,
is the question whether ‘matched interface ratio’ can also be used for measuring ‘interoperability’ in business services. Regarding ICT services, an interface is an intermediary in order to enable communication between an ICT system as a consuming service component and an ICT system as a providing service component. Regarding business services, there is no clear distinction between the providing service component and the interface through which communication can be enabled. The service consuming party communicates with the one who will perform the promise act as part of the service whereby this person is part of the providing service component. Therefore, the indicator ‘matched interface ratio’ is not applicable to business services. On the other hand, the indicator ‘effort per interaction’ can clearly be used as an indicator for measuring interoperability in business services. This was actually clarified in the previous paragraph.

As can be seen, ‘interoperability’ concentrates on the communication between the service consuming party and the providing service component. It concerns the external details of the providing service component. Hence, if we regard Salmela’s view on quality, we can relate ‘interoperability’ to the IS user quality. Furthermore, it can be related to service usage information as part of the service specification framework.

A.1.4 Compliance

Compliance comprises of attributes of software that make the software adhere to the application related standards, conventions or regulations in laws and similar prescriptions [63]. We should regard compliance as a quality aspect for ICT services, since the realization of SOA strongly relies on standards and other prescriptions. As can be seen above regarding ICT services, ‘interoperability’ between providing service components can only be realized if there is syntactic understanding by using the same data format. In order to enable such a data format, it should be clear to what norms the format should keep. In that case, standards can clarify these norms. ‘Compliance’ and ‘interoperability’ are thus strongly related to each other. We can select a number of indicators for measuring compliance. Standardized data format ratio indicates the ratio of standardized data formats to the data formats to be standardized [63]. Above, we already gave an example of a standard for defining a data format, which was XML. The more the data format used by ICT services are based on a specific standard, the greater the chance providing service components can provide ICT services to each other. Logically, a higher value of the indicator ‘standardized data format ratio’ is preferable. The indicator standardized interface ratio further underlines the strong relation of ‘compliance’ with the quality subcharacteristic ‘interoperability’. As with the standardized data format ratio, a higher value of the standardized interface ratio is preferable. The section above concerning ‘interoperability’ already explained the aim of interfaces.

Let us now study whether we can relate ‘compliance’ to business services. If we regard the definition of ‘compliance’, it conveys the impression that we actually may consider ‘compliance’ as a quality aspect for business services. Prescriptions are commonly used in business in order to put the activities within the enterprise on the right lines. These are for instance rules and regulations that conform to policies in the enterprise. Let us consider the indicator ’standardized data format ratio’. If we relate this to business services,
standardization of the data format gives a clear notion on how a service consuming party can request a service. If we regard the accountant and archivist example, we can consider clear rules on how input data should be handed over to the archivist in order to deliver the financial reports of 2006. One might think, for instance, of a request form which should be filled in and handed over to the archivist in order to obtain the financial reports of 2006. In this example, we assume that no standards or other prescriptions are known on in what format this form should be delivered to the archivist. The accountant could send this request form by email and later on visit the archivist on the assumption that he can pick up the financial reports. However, the archivist had to disappoint the accountant. The archivist could not deal with the request of the accountant, since the request form had to be handed over to the archivist on paper personally. As a consequence, the accountant will now have a negative view on the service delivered by the archivist, since the archivist did not state in what data form the request from should be handed over. Therefore, standardized data formats can positively influence the accountant’s view on the service quality. Hence, a higher value of the indicator ‘standardized data format ratio’ is preferable. The indicator ‘standardized interface ratio’ cannot, however, clearly be related to business services. As discussed in ‘interoperability’, interfaces are not used very often in business services. Therefore, we do not need to regard ‘standardized interface ratio’ as an indicator for measuring ‘compliance’ in business services.

Obviously, ‘compliance’ focuses on standards, regulation and other prescriptions concerning the communication between the service consuming party and the providing service component. Therefore, it should be included in service usage information. Measurements regarding ‘compliance’ involve the use of the service. In other words, ‘compliance’ can be related to IS user quality.

A.1.5 Security

Security embraces the attributes of software that bear in its ability to prevent unauthorized access, whether accidental or deliberate, to programs and data [63]. ICT services should also include ‘security’ as a quality aspect, since unauthorized access is an important issue to deal with. An ICT service might, for instance, only be used by organization A and not by organization B. In that case, organization B is unauthorized for using the ICT service and therefore access by organization B should be prevented.

In order to determine the degree of ‘security’ in ICT services, we should measure the hacker-resistance and the ciphered data ratio. The Extended ISO-model defines ‘hacker-resistance’ as the inability of a group of renowned hackers to infringe on the software product within a redefined period of time, while physical access to the software product does exist [63]. ‘Hacker-resistance’ is quantified by determining the time period wherein hackers should try to infringe on the software product and the number of hackers that succeeded. If none of the hackers succeeded, it is said to be hacker-resistant [63]. If we consider this in terms of ICT services in SOA, we can interpret this as the extent to which ICT services are resistant to unauthorized parties that are trying to make use of the service. In this case, the degree of ‘hacker-resistance’ can also be determined by selecting hackers who try to connect with the providing service component and to execute the service. A hacker has
Identification of Quality Aspects Using the Extended ISO-model

A.1 Functionality

succeeded if the providing service component performs a promise act indicating that it will execute the service. Another indicator is the *ciphered data ratio*, which is defined as the ratio of ciphered data to the data be ciphered [63]. A higher ratio implies better security of the software concerned. Ciphered data can be related to data encryption in SOA. It should be in place to preserve privacy [63]. When SOA is realized using Web services, standards such as WS-Security [2] can encrypt the data that is sent between the service consuming party and the providing service component. Hence, ‘ciphered data ratio’ is applicable to ICT services.

Regarding business services, we should also deal with ‘security’ as a quality aspect. We can clarify this using the indicators discussed above and the example of the accountant and the archivist. The indicator ‘hacker-resistance’ can be interpreted as malicious people who are trying to make use of the business service while they are not authorized. For instance, a person with a knife could threaten the archivist with death if the archivist does not deliver him the financial reports of 2006. If the archivist is situated in a room apart from that person, that person cannot do him any harm. Else, the armed person could grab the archivist and force him to deliver the financial reports. In this situation, the archivist is more secure if he is located in another room than service consuming parties. Regarding ‘hacker-resistance’, it can be tested in what way and how often malicious people are able to force the archivist to deliver the financial reports within a certain amount of time. If neither of the malicious people succeeds, the business service is hacker-resistant. The indicator ‘ciphered data ratio’ can also be regarded as data encryption. It focuses on the fact that data sent between the service consuming party and the providing service component cannot be read by other people. As with ICT services, it should be in place to preserve privacy. Let us consider, for instance, that the financial reports are delivered to the accountant on a USB stick. While the archivist is going to the accountant’s office, he accidentally drops the USB stick on the floor. The stick is then taken by someone else, who could view these reports on its own computer. If the reports are confidential, it is not desirable that other people besides the accountant can read the reports. Data encryption would secure the data on the USB stick from other people.

The above makes clear that we should relate ‘security’ to IS user quality. The fact is, ‘security’ concerns the service quality during the communication of the service consuming party with the providing service component. Therefore, it fits into service usage information.

A.1.6 Traceability

Traceability refers to attributes of software that bear on the effort needed to verify correctness of data processing on required points [63]. Contrary to the subcharacteristics discussed so far, we do not need to consider ‘traceability’ as a quality aspect for ICT services or business services. By verifying data correctness during the process on required points, the internal details of the providing service component are concerned. This is not interesting for service consuming parties. Service consuming parties only concern the correctness of data after service execution. If the required point is only set at the end of the execution, this characteristic is essentially the same as ‘accuracy’ [63]. This underpins the fact that
there is no need to include ‘traceability’ as a quality aspect for ICT and business services. In terms of business services, we can clarify this using the example of the accountant and the archivist. After the accountant requested the archivist to deliver the financial reports of 2006, he is only interested in the fact that he actually receives those reports from the archivist. The accountant does not care whether other reports or data are used during the performance of the service by the archivist as long as he receives the financial reports of 2006.

### A.2 Reliability

*Reliability* is defined as a set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period [63]. Regarding services, this holds to what extent the providing service component maintain its level of performance under stated conditions for a stated period concerning the service that the service component provides. ‘Reliability’ can be further divided into the subcharacteristics *maturity, fault tolerance, recoverability, availability and degradability*. We describe them here below.

#### A.2.1 Maturity

The Extended ISO-model defines ‘maturity’ as the attributes of software that bear on the frequency of failure by faults in the software [63]. We can interpret ‘maturity’ in terms of ICT services as the frequency of service failure by faults in the providing service component. There are a number of indicators defined for measuring ‘maturity’. Yet, regarding ICT services, we only consider those indicators that are of interest to service consuming parties. Because of this, we can use *mean time between failures* (MTBF) as an indicator for ‘maturity’. MTBF holds the average time that passes between two failures [63]. Logically, the higher the average time, the higher the degree of ‘maturity’. The indicator *test density* states the ratio of test volume (e.g. number of test cases) conducted during the development phase to the unit volume of a product tested and released [63]. More test cases conducted indicate a more thorough study on faults in the providing service components. This implies less chance on service failures. A higher value of the test density denotes a higher degree of maturity of the service.

It is feasible to include ‘maturity’ as a quality aspect for business services. If we relate ‘maturity’ to our notion on ‘service’, ‘maturity’ implies the frequency of incorrect performance of one of the C-acts or P-act in the service. Regarding business services, human beings can also make mistakes in their performance of one of the C-acts or P-act. In order to clarify this, we should consider the indicators discussed above. MTBF holds the average time that passes between two mistakes that human beings make in performing the service. If we again consider the accountant and archivist example, the archivist could make a mistake by delivering the financial reports of 2005 instead of the ones from 2006. He can make the same mistake the next day or not until the next month. If the archivist makes the same mistake twice within in two days, it clearly denotes a lower degree of ‘maturity’ than when the same mistake is made not until the next month. As with ICT services, a higher value of MTBF is preferable. Regarding business services, we can also make use of the indicator
Identification of Quality Aspects Using the Extended ISO-model A.2 Reliability

‘test density’ for measuring maturity. Yet, we usually do not speak of testing how human beings perform their acts. We should rather regard ‘test density’ as the ratio of the number of practices that are done before the service is available to the unit volume of the set of human beings that have practiced for performing the service. Let us clarify this using the example of the accountant and the archivist. Here, we consider the archivist as the delegator, who authorizes two other employees as functionaries for performing the service. It should be noted that the delegator still takes the final responsibility for the service. The concept of ‘delegation’ was explained in Section 4.4. Before the two employees are actually able to deliver the service to service consuming parties, they should have to get used to the work first. If not, these employees will make mistakes when delivering the service to service consuming parties. The more these employees practice, the less the chance will be on making mistakes. What is further meant with the unit volume of the set of human beings is the number of employees that are involved in performing the service. For instance, in case of two employees, one employee could perform the promise act of the service and the other one the execute and state act. When three employees are authorized to perform the service, each of them could perform one particular act of the service. All in all, it is clear that we can consider ‘maturity’ as a quality aspect for business services.

As can be seen, ‘maturity’ concerns how human beings and ICT systems perform the acts composing the service. It focuses on the failures during service execution and communication with the providing service component. It should therefore be mentioned in service function information as well as in service usage information. We can relate ‘maturity’ to IS user quality, since it affects the service consuming party’s view on the quality of the service.

A.2.2 Fault tolerance

Fault tolerance embraces attributes of software that bear on its ability to maintain a specified level of performance in cases of software faults or of infringement of its specified interface [63]. Fault tolerance is interpretable in terms of ICT services. In that case, we can regard fault tolerance as the ability to maintain a specified level of service performance when a particular C-act or P-act causes a fault. In other words, it concerns to what extent the providing service component can still correctly provide the ICT service while at least one of the C-acts or P-act composing the service is not correctly performed.

In order to measure ‘fault tolerance’ of ICT services, we can make use of the indicator operational/input error detection ratio. This indicator is the ratio of number of erroneous operations or inputs detected by a software system to the number of erroneous operations or inputs conducted during a given period of time [63]. In terms of ICT services, we can regard ‘inputs’ as the input data that the service consuming party hands over and that is needed for service execution. Erroneous inputs thus hold input data provided by the service consuming party which should lead to the performance of the decline act by the providing service component instead of the promise act. ‘Operations’ should be viewed as the performance of the C-acts and P-act that composes the service. Consequently, ‘erroneous operations’ denotes improper performance of at least one of these C-acts and P-act. Logically, the providing service component should aim at detecting each erroneous operation or input that has been conducted. If not, improper service performance could lead to an incorrect...
A.2 Reliability

Identification of Quality Aspects Using the Extended ISO-model

service result. Furthermore, erroneous inputs could lead to incorrect processing of the input data and eventually result in incorrect service output. Hence, a high value of the indicator ‘operational/input error detection ratio’ is preferable.

Now it is interesting to investigate whether we may regard ‘fault tolerance’ as a quality aspect of business services as well. As stated above, ‘fault tolerance’ concerns to what extent the service can properly be performed while at least one of the C-acts or P-act is not correctly performed. Since these C-acts and P-act can be performed by human beings, whether correctly or not, ‘fault tolerance’ can be viewed as a quality aspect for business services. The indicator ‘operational/input error detection ratio’ is also suitable to measure ‘fault tolerance’ in business services. Let us further clarify this using the accountant and archivist example. Here, we again concern the fact that the archivist has delegated the authority to perform the service to other employees. In order to request the financial reports of 2006, the accountant had to fill in a request form which should be handed over to one of the employees. The financial reports will then be delivered to his office. In this case, an input error is when the request form has not been filled in correctly. For instance, the accountant could accidentally fill in another office name than his own office. The employees could detect this mistake by verifying whether the information in the form is correct. An erroneous operation could be, for instance, that one of the employees gets the financial reports of 2005 from the archive instead of the ones from 2006. The detection of this erroneous operation implies that the other employees could notify the employee that he obtained the wrong financial reports from the archive.

The above showed that ‘fault tolerance’ concerns the performance of the C-acts and P-act that compose the service. Therefore, it should be included in service function information and service usage information. As with ‘maturity’, ‘fault tolerance’ affects the service consuming party’s view on the quality of the service and should thus be related to IS user quality.

A.2.3 Recoverability

The Extended ISO-model defines ‘recoverability’ as the attributes of software that bear on the capability to re-establish its level of performance and recover the data directly affected in case of a failure and on the time and effort needed for it [63]. At first sight, it is questionable whether ‘recoverability’ is of any concern to service consuming parties. Service providers are responsible for re-establishments of service performance and the effort needed is only of their concern. Yet, it can be stated that the effort needed for re-establishments influences the time needed for re-establishments. Let us consider the fact that one person is responsible for the re-establishment of the performance level of a providing service component. Logically, putting more effort on the re-establishment implies more time spent on the re-establishment by this person. Hence, ‘recoverability’ should be included as a quality aspect in ICT service specification. Indicators that concern the time needed for re-establishments directly indicate the effect of re-establishments to service consuming parties. The longer the time needed for re-establishments, the longer the providing service components do not perform regarding their level of performance. If there are no indicators for measuring the time needed, we can make use of the indicators that indicate the effort needed for re-establishments. Let us now
discuss the indicators for measuring ‘recoverability’ in more detail.

Mean time to repair (MTTR) indicates the average time needed to recover the damaged software product [63]. Regarding ICT services, it indicates the average time that the service cannot be used if the damaged providing service component is repaired by the provider. Logically, a lower average time implies better ‘recoverability’ of the ICT service. The indicator auto recovery ratio is the ratio of automatically resolved failures (by the software product) to the total numbers of failures for which automatic recovery would be preferred [63]. In terms of ICT services, it indicates to what extent the providing service component that provides the ICT service can automatically resolve the service failures performed without any intervention of human beings to repair the providing service component. In order to realize automatic resolution, failures that could arise should already be identified during the development of the providing service component. The developer should further develop the providing service component in such a way that it is able to cope with the failures identified. Automatic resolution implies that no intervention of the service provider, who is responsible for the providing service component, is needed during service usage. The providing service component resolves the failures itself and the service consuming party does not need to contact the service provider to notify the failures. Mean fault life is the mean time between introduction of a fault and removal of the fault [63]. Regarding ICT services, the introduction of a fault implies the moment that a service consuming party notifies a fault during service usage. Consequently, the service consuming party will contact the service provider. This person or company is also responsible for removing the service fault so that the ICT service can be correctly performed. Logically, a lower mean time is preferable.

Recoverability in business services concerns faults made by human beings performing the service and the time and effort needed for recovery. We can make this clearer on the basis of the indicators discussed above. As stated above, the indicator MTTR concerns the average time needed to recover the damaged providing service component. In business services, such a providing service component is composed of a set of human beings. Here, damage means that at least one of the human beings does not correctly perform his act or that one of the C-acts or P-act is not performed. If a human being makes a fault in his performance, one should draw his attention to the fault and make clear how to perform the act correctly. The providing service component is recovered if the human being concerned does not make faults anymore. Let us again regard the example wherein the archivist delegates authority to other employees for delivering financial reports. If one of these employees does not correctly perform his act, the archivist can clarify him how mistakes should be prevented. If this employee still does not correctly perform his act, the providing service component is still damaged. The archivist can take actions by dismissing this employee. If the other employees can correctly perform the act of the dismissed employee, the providing service component is recovered. As MTTR, the indicator mean fault time can also be used for measuring ‘recoverability’ in business services. The introduction of a fault refers to the fact that an incorrect performance of a human being is discovered. The removal of the fault implies that the human being performs correctly again.

All in all, ‘recoverability’ concerns to the time and effort needed to recover the damaged providing service component. From a service consuming party’s perspective, it is interesting to know how long the providing service component is damaged. The longer the
A.2 Reliability

Identification of Quality Aspects Using the Extended ISO-model

providing service component is damaged, the longer the service consuming party makes use of a service that is not properly executed. As can be seen, the degree of ‘recoverability’ affects the service consuming party’s view on the quality of the service. Recovery of the damaged providing service component demands interferences of the service provider and may eventually lead to modifications in the providing service component. In other words, the providing service component may be re-developed. Therefore, we can relate ‘recoverability’ to IS work quality.

A.2.4 Availability

‘Availability’ denotes the attributes of software that bear on the amount of time the product is available to the user at the time it is needed [63]. We can regard ‘availability’ as a quality aspect for ICT services. In that case, it concerns the amount of time that the service is available to the service consuming party at the time the service is needed. An indicator for ‘availability’ is the relative availability ratio. It is the ratio of time the software product is available to the time it is needed [63]. In terms of ICT services, it is the ratio time of the service that is available to service consuming parties to the time it is needed by service consuming parties. As a service consumer party, it is desirable that an ICT service is available whenever he needs to use the service. Hence, a higher value of the relative availability ratio is preferable.

‘Availability’ can also be regarded as a quality aspect in business services. It depends on the human beings that are available to perform the business service. Let us clarify this using the indicator ‘relative availability ratio’. For instance, considering the accountant and archivist example, the archivist works at the office each Thursday and Friday and the accountant only at Monday. In order to obtain the financial reports, the accountant should personally request this to the archivist. In that case, the service ‘delivery of financial reports’ is unavailable to the accountant whenever he needs those reports. Consequently, the relative availability ratio is zero then. As with ICT services, a higher value of the relative availability ratio is preferable in business services.

It can be noted that ‘availability’ denotes the availability of the providing service component in order to communicate with it as a service consuming party. Therefore, we should mention ‘availability’ in service usage information. It goes without saying that ‘availability’ affects the service consuming party’s view on the quality of the service component, whereby we should relate ‘availability’ to IS user quality.

A.2.5 Degradability

The Extended ISO-model defines degradability as the attributes of software that bear on the effort needed to re-establish the essential functionality after a breakdown [63]. In our opinion, degradability should not be concerned as a quality aspect for ICT and business services, since we state that a service offers one specific functionality to service consuming parties. Hence, we cannot make any distinction between essential functionalities and less important functionalities regarding a particular service.
A.3 Usability

The Extended ISO-model defines usability as a set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users [63]. Unlike the main quality characteristics discussed so far, it is questionable whether we should consider usability as a main quality aspect for ICT services. Let us consider the quality subcharacteristics of ‘usability’ as depicted in Figure 6.4. Subcharacteristics such as attractivity, clarity and user-friendliness clearly focus on how an ICT system presents the functions that it provides. Differently said, ‘usability’ concerns in what way human beings can make use of services that ICT systems provide and how these services are presented to these human beings. Now the question arises whether we should consider the presentation of ICT services and the access to them as a quality aspect of ICT services. In [62] a clear overview is given of the essential parts in SOA. These essential parts are portals and services among other things [62]. As can be derived from that overview, a portal provides an access point to ICT services in the form of a user interface. As can be seen, ‘usability’ and its subcharacteristics refer to the quality of the user interface rather than the ICT services themselves. Since portals and services are different parts in SOA, we do not need to view ‘usability’ and its subcharacteristics as quality aspects of ICT services.

Regarding business services, it is also questionable whether we should regard the presentation and access to business services as a quality aspect. Let us again consider the accountant and archivist example taking into account that the archivist is located in another building. We describe two situations in broad outline. In the first situation, the archivist is the only person in the building. Since the archivist is one his own, it takes a lot of time to serve one person. The building has no chairs, toilets or any other facilities to make customers more comfortable. In the second situation, there is an information desk where the accountant can request the financial reports. While the archivist deal with the request, the accountant can sit down at a chair reading some magazines. In addition, there are toilets for just in case.

Above, we gave two examples, which concerned how the accountant can make use of the service and how this service is presented to the accountant. It goes without saying that in the second example the service is presented in a better way than in the first example. Furthermore, it is easier for the accountant to retrieve the financial reports in the second example than in the first example. Logically, the accountant would prefer the second situation for retrieving the financial reports. Despite this clear difference between the two examples, it does not state anything about the quality of the service itself. Since the building and its facilities and the service are different parts, it is not needed to regard ‘usability’ and its subcharacteristics as quality aspects of ICT services.

A.4 Efficiency

Efficiency comprises a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions [63]. On the basis of this definition, it is not clear whether the main quality characteristic ‘efficiency’ is of any concern to service consuming parties. Therefore, we should further
study the quality subcharacteristics in order to have a clear view on this. As stated by the Extended ISO-model, ‘efficiency’ in software products is determined by measuring the indicators time behaviour and resource behaviour. We discuss them here below.

A.4.1 Time behaviour

Time behaviour comprises the attributes of software that bear on response and processing times and on throughput rates in performing its function [63]. This quality subcharacteristic should also be included in ICT service specification. Regarding service consuming parties, it is of great interest to have an overview of aspects such as response and processing time. It is logical to state that service consuming parties prefer less response and processing time of the ICT service. The sooner they can receive the service result the better. Several indicators are proposed for measuring time behaviour in software products, which can also be used regarding ICT services. The indicator batch turnaround time is the time that passes between start and finish of background processing [63]. If we relate this to ICT services, it is the time that passes between the performance of the promise act and the performance of the state act of the ICT service.

Processing time denotes the average and maximum time a user needs for a certain processing task with a certain usage [63]. It is determined by registering the time when the user starts the first action of the processing task and the time when the user is ready to start another processing task [63]. Let us consider the pattern of C-acts and P-act that are performed by the service consuming party and the providing service component during service usage (see Section 4.1). The request act is the first action performed by the service consuming party. We can interpret ‘the time when the user is ready to start another processing task’ as the time when the service consuming party is able to perform another C-act. As can be seen in Section 4.1, this is when the service consuming party is able to perform another service call. This can be the performance of a request act for a next service after the current service conversation has been ended by the performance of the accept act. Yet, another service call can also be made by performing a reject act in the current service conversation. This is feasible after the performance of the state act by the providing service component. The indicator turnaround time indicates the speed of processing by measuring the elapsed time between the beginning of process requirement and gaining the result of the process [63]. Regarding our notion on ICT services, the holds the elapsed time between the performance of the request act by the service consuming party and the performance of the state act by the providing service component. The response time indicates the speed of processing by measuring the elapsed time between the end of inquiry or request to the computer system and the start of response [63]. If we relate this to ICT services, this implies the elapsed time between the end of the performance of the request act and the start of the performance of the promise act by the service component. Other indicators proposed for measuring the time behaviour do not fit into the domain of services. These indicators concern the internal part of the providing service component such as the indicator average internal transaction time.

‘Time behaviour’ also plays an important role as a quality aspect in business service
specification. Let us clarify this using the indicators discussed above. As with ICT services, ‘batch turnaround time’ indicates the time that passes between the performance of the promise act and the performance of the state act in business services. For instance, if we again regard the example concerning the accountant and the archivist, the measurement of the batch turnaround time of this business service starts when the archivist promises the accountant to deliver the financial reports of 2006. It ends when the archivist states that the delivery has been done i.e. the financial reports are available to the accountant. ‘Processing time’ can also be used as an indicator for time behaviour in business services in the same manner as with ICT services. Regarding business services, service consuming parties can also perform another request act and a reject act as well. The indicator ‘turnaround time’ is also useful to be included in business service specification. Regarding the example of the accountant and the archivist, ‘turnaround time’ is the elapsed time between the moment that the accountant has made his request for the delivery of the financial reports and the moment that the archivist confirms to deliver those reports.

As can be seen, it is of great interest for service consuming parties to denote the time behaviour of the service, since they aim at obtaining the service result as soon as possible. Changes of the measurement values of the indicators affect the view of service consuming parties on the quality of the service. For instance, a service consuming party logically prefers a service that provides the service result within a second rather than a service that needs five minutes to present the service result. The time behaviour does not only depend on the time that the providing service component needs to perform the actual service, but also on the communication with the providing service component. One might think, for instance, of the time needed to receive a service request. Therefore, we should mention ‘time behaviour’ in service function information as well as in service usage information. Furthermore, we can relate ‘time behaviour’ to IS user quality as part of Salmela’s view on quality.

A.4.2 Resource behaviour

Resource behaviour is defined as the attributes of software that bear on the amount of resources used and the duration of such use in performing its function [63]. If we relate this to ICT services, ‘resource behaviour’ concerns the amount of resources used by the providing service component in order to provide the service. If we consider the indicators for measuring ‘resource behaviour’, it can be noted that these indicators focus on hardware as resources. Indicators such as internal memory occupancy, processor occupancy, CPU utilization and main memory utilization are logically not of any interest to service consuming parties. These indicators fully concern the internal part of the providing service component. Therefore, we do not include ‘resource behaviour’ in the service specification as a quality aspect for services.

A.5 Maintainability

Maintainability comprises a set of attributes that bear on the effort needed to make specified modifications [63]. The definition of ‘maintainability’ is in our opinion too general
in order to determine whether ‘maintainability’ is of any concern to service consuming parties. Let us therefore study its subcharacteristics in more detail. According to the Extended ISO-model, maintainability can be divided into the subcharacteristics analysability, changeability, stability, testability, manageability and reusability.

A.5.1 Analysability

Analysability indicates the attributes of software that bear on the effort needed for diagnosis of deficiencies or causes of failures, or for identification of parts to be modified [63]. It should be noted that the diagnosis or identification of failures and parts to be modified are not of interest for service consuming parties. It is a specific task of the persons that are responsible for the providing service component to diagnose failures and parts to be modified. Hence, analysability should not be viewed as a quality aspect for ICT or business services.

A.5.2 Changeability

Changeability denotes the attributes of software that bear on the effort needed for modification, fault removal or for environmental change [63]. As the indicators for ‘changeability’ will clarify, ‘changeability’ focuses on changes made due to defects or failures of the software. The internal part of service components providing ICT services also need to be modified in case of defects or failures. This affects the service operation and consequently it influences the service consuming party in the performance of his activities. In this sense, the indicator mean failure treatment time can clearly underline the effect of changes in the providing service component for the service consuming party. ‘Mean failure treatment time’ indicates the mean time for the failure occurrence to the restoration for end users [63]. The longer the mean time, the longer the ICT service does not properly perform. Logically, a lower value of this indicator is preferred. Other indicators proposed for determining changeability concern modifications of the internal part of the providing service component such as the indicator modification effort per unit volume. A service consuming party does not care how much effort is needed to modify the service component. He only concerns the extent to which these modifications affect the use of the service. Eventually, these effects can be related to ‘mean failure treatment time’.

At first sight, it is hard to imagine how ‘changeability’ can be regarded as a quality aspect in business service specification. Regarding human beings, we do not speak of restorations or of making changes in the service component due to defects. It is, however, feasible to regard ‘changeability’ as a quality aspect for business services. As with ICT services, the indicator ‘mean failure treatment time’ denotes the time from a failure occurrence to the restoration for service consuming parties. Let us make this clear using the accountant and archivist example. We again regard the fact that the archivist has delegated authorization to other employees for delivering the financial reports. Failure occurrence can be viewed as the moment that the accountant, the archivist or the other employees notices that one of the employees does not properly perform his act in order to deliver the financial reports. The archivist, who takes the final responsibility for the delivery of the financial reports, should
'restore' this fault for the service consuming parties. Restoration could hold, for instance, that the archivist replaces the employee concerned by another employee.

As can be seen, 'changeability' concerns modifications made in the providing service component and the effects of these modifications made for service consuming parties. Hence, we can relate 'changeability' to IS work quality, since it concerns the maintenance of the providing service component. In addition, modifications can be made regarding the C-act or the P-act as part of the service. Therefore, we should include 'changeability' in service function information and in service usage information.

A.5.3 Stability

The Extended ISO-model describes stability as the attributes of software that bear on the risk of unexpected effect of modifications [63]. In order to determine whether we can denote 'stability', we should find out what is meant by 'unexpected effect'. Let us consider the indicators for 'stability' in that case. The Extended ISO-model introduces only one indicator for measuring 'stability'. \textit{Ratio of new faults at revision} indicates the ratio of new faults made at the revision [63]. What can be derived from this definition is that faults are meant by 'unexpected effect'. The previous sections already made clear that faults affect the service consuming party in using the service and his communication with the providing service component. Thus, we can state that 'stability' implies the risk of new faults that arises after the providing service component has been modified. Logically a lower value of the indicator 'ratio of new faults at revision' is desirable.

In order to explain whether 'stability' can be regarded as a quality aspect for business services, we should consider the accountant and archivist example. We should further regard delegation of the authorization to other employees to deliver the financial reports of 2006. Modifications in the providing service component might be, for instance, the replacement of one of the employees as a result of improper performance of the employee that has been replaced. However, the replacement of the employee does not immediately mean that the service is properly provided again by the employees. Even though the new employee does not make the same mistake as the former employee, he could also improperly perform by making another kind of fault in his performance. Hence, this can be regarded as a new fault at the revision. Consequently, we may also use the indicator 'ratio of new faults at revision' for measuring 'stability' in business services.

The above shows that 'stability' concerns the number of faults that may arise as a result of modifications in the providing service component. Since such faults arise during the usage of the service, it can be stated that the degree of 'stability' is of great concern to service consuming parties. The degree of 'stability' is determined by the modifications made within the providing service component. Therefore, we should relate 'stability' to IS work quality of Salmela’s view on quality. Next to this, faults may arise regarding the communication with the providing service component and the actual service execution. Hence, 'stability' should be included in service function information and service usage information.
A.5.4 Testability

Testability comprises the attributes of software that bear on the effort needed to validate the software product [63]. Testability can be interpreted in terms of ICT services. In that case, it concerns the effort needed to validate the providing service component. Regarding service consuming parties, it is desirable to have the opportunity to test the ICT service before actual usage. In that way, the service consuming party can check whether the ICT service really performs as it should be. The degree of testability influences the quality of the ICT service. We are able to measure testability on the basis of the indicator *mean user’s work time to verify the fault correction*. As the term already suggests, this indicator denotes the mean user’s work time to verify the fault correction. If the service consuming party notifies a fault when using the service, he will contact the service provider. This person or company is also responsible for the fact that the fault is resolved. When the fault has been resolved, the service consuming party would likely to verify the fault correction before he actually will use the service again. Particularly, when there are expenses involved, the service consuming party would like to be sure that the service can correctly perform. It is rational to state that the service consuming party prefers less time to verify the fault correction. Other indicators have been listed in the Extended ISO-model, yet, they are not appropriate regarding services, since they clearly focus on ‘testability’ from a developer’s perspective. One might think, for instance, of the indicator *test work time per changed source code line* which is the work time to test updated and added source code per unit line in case of correcting faults and/or case of remodelling [63]. It goes without saying that service consuming parties do not concern source code.

In terms of business services, it seems to be unlikely that service consuming party test the service before actual usage. As stated earlier, we should regard ‘tests’ rather as practices that are performed by employees before the service is available to service consuming parties. Regarding the example of the accountant and the archivist, we can consider the archivist that authorizes two other employees for performing the service through delegation. In that case, these two employees form the providing service component that provides the service. A fault in the providing service component could be, for instance, that one of these employees makes a bad job of his work. The archivist can take actions by dismissing this employee and replacing him by another employee. If this employee makes a good job of his work, the fault is resolved. Logically, it is hard to imagine that the accountant as a service consuming party tests the employees whether they properly deliver the financial reports of 2006. Regarding the definition of ‘testability’, the archivist as the service provider is the one who validates the providing service component. Yet, let us consider another example. Here, one might think of people who have been signed up at an employment agency. A company might want to utilize an employee from the employment agency for a long period of time, for instance, two years. However, it first wants to see whether the employee performs appropriately. Hence, the company and the employment agency can enter into agreement that it may dismiss the employee any time within the first two months. In this example, the employment agency can be regarded as the service provider and the employee as the one who does the actual service performance. The company can be viewed as the service consuming party. The first two months can be interpreted as a test period for the employee. Hence, we can speak of
As can be seen, ‘testability’ deals with the effort and time needed in order to examine whether the providing service component provide operates as it should be. Although ‘testability’ may involve service consuming parties, it is mainly an activity performed during the development or maintenance of the service component. As can be noted, ‘testability’ should be included in service function information and service usage information. ‘Testability’ refers to the validation of the providing service component and the service as a whole, which implies the validation of the actual service execution and the communication with the providing service component as well.

A.5.5 Manageability

Manageability is defined as the attributes of software that bear on the effort needed to (re)establish its running status [63]. On the basis of this definition, it is not clear yet to state whether ‘manageability’ is of any concern to service consuming parties. Let us therefore discuss the indicators for measuring ‘manageability’.

The Extended ISO-model proposes only one indicator in order to measure ‘manageability’. The indicator ‘control effort ratio’ is the ratio of effort put in controlling the software product in man-hours to the number of hours the product is available to the users [63]. What this indicator measures in terms of ICT services is the effort put in controlling the providing service component and the service that it provides to the time the service is available to service consuming parties. Regarding the definition of ‘control effort ratio’, controlling the providing service component and the service does not affect the service consuming party in using the service. As a consequence, the effort put in controlling the providing service component and the service is not noticeable to service consuming parties. From this we can state that ‘manageability’ does not need to be regarded as a quality aspect for services.

A.5.6 Reusability

Reusability denotes the attributes of software that bear on its potential for complete or partial reuse in another software product [63]. Regarding ICT services, ‘reusability’ indicates to what extent the service can be used for other purposes. Let us consider, for instance, a removal service which is able to remove documents that are represented in the Portable Document Format (PDF) [28]. A service consuming party might make use of the service in order to remove for instance registration forms represented in PDF. Yet, it can also be used for other purposes such as the removal of declaration forms as long as the forms are represented in PDF. On the other hand, a calculation service for instance should also deal with the content aspects of data. This makes a calculation service less generic than a removal service. Generally, infological ICT services are less generic than datalogical ICT services. Likewise, ontological ICT services less generic than infological ICT services. Terlouw also denotes this by stating that unlike the other ICT services, ontological ICT cannot be divided into categories since they are specific to a certain industry [54]. It goes without saying that the degree of reusability of a service depends on how generic the service is. The more generic a service is, the higher the degree of reusability is.
Although it is clear that `reusability` can be regarded as a quality aspect for ICT services, it is questionable whether the indicators proposed by the Extended ISO-model are suitable for measuring `reusability` concerning ICT services. The Extended ISO-model defines two indicators for measuring `reusability`, which are `ratio reusable parts` and `ratio reused parts`. These indicators define the ratio of reusable or reused parts of the software product to the total number of parts of the software product [63]. As can been above, reusability in ICT services concerns the reusability of the providing service component as a whole, since a service consuming party is only interested in the service that the providing service component as a whole provides. Therefore, these indicators or not suitable for measuring `reusability` of ICT services.

It can be noted that what we stated in the first paragraph also holds for business services. The degree of reusability is based on the fact whether the service is datalogical, infological or ontological. As explained in Section 4.2, we can make this same distinction with business services.

As can be seen, `reusability` concerns the functionality of the providing service component and to what extent this functionality can be reused for other purposes. The degree of `reusability` does not concern the communication between the service consuming party and the providing service component. Therefore, we should relate `reusability` to IS work quality and service function information.

### A.6 Portability

Portability denotes a set of attributes that bear on the ability of software to be transferred from one environment to another [63]. If we interpret this in terms of ICT services, `portability` holds the ability of the providing service component, which provides the ICT service, to be transferred from one environment to another. As stated by Papazoglou et al., services should have their definitions and location information stored in a repository and be accessible by a variety of clients that can locate and invoke the services irrespective of their location [40]. In other words, the physical location of ICT systems providing services should be hidden from the service consuming parties. This implies that service consuming parties do not concern about the physical location of providing service components. In that way, they do not care whether the providing service component is able to be transferred from one environment to another. Hence, we do not need to consider `portability` as a quality aspect of services.