FINDING AN OPTIMUM IN SERVICE GRANULARITY

2010

René Wiersma

HU University of Applied Sciences
Faculty of Science Engineering

Master of Informatics thesis on:

Finding an Optimum in Service Granularity

Author

R.W. Wiersma
HU University of Applied Sciences, Faculty Science and Engineering

Supervisors

Drs. ing. J.P.P. Ravesteyn, HU University of Applied Sciences, Faculty Science and Engineering
Prof. dr. ir. R.G. Slot MBA, HU University of Applied Sciences, Faculty Science and Engineering

Examination Committee

Prof. drs. A.J.G. Silvius MBA, HU University of Applied Sciences, Faculty Science and Engineering
Drs. ing. J.P.P. Ravesteyn, HU University of Applied Sciences, Faculty Science and Engineering
Prof. dr. ir. R.G. Slot MBA, HU University of Applied Sciences, Faculty Science and Engineering

PREFACE

My interest in service granularity started when I was working in my first Service Oriented Architecture (SOA) project in 2005. At that time I was responsible for orchestrating business processes with the Business Process Execution Layer (BPEL) process manager as part of the Oracle SOA Suite. Besides orchestrating business processes I also defined and developed the services to be orchestrated. At that time I could not find any models helping me in defining the services I was creating. In the beginning of the project I formulated several concepts and started implementing them, but after a while I had to reconsider them because changes occurred. Those changes were very volatile and diverse, and most of them had impact on the defined service model.

When I visited the LAC2006 (Landelijk Architectuur Congres) in Nieuwegein I met Claudia Steghuis. She was telling me about her master thesis in service granularity. That caught my interest, because I faced that kind of questions myself that time. I read her master thesis and I felt comfortable with her research findings. Her study helped me in recognizing service granularity, but it did not provide a method how to optimize service granularity.

In 2008 I decided to start my master study in Informatics. The first semester I followed several modules that were about or related with Design & Engineering Methodology for Organizations (DEMO). The methodology itself has been developed by professor Dietz at Delft University of Technology. The knowledge I obtained about DEMO and my interest in service granularity fused into a research topic for my master study. During the several modules and some practical experience as well, I discovered the possibilities of DEMO.

It was a paper assignment of the Business Process Management course lectured by Pascal Ravesteyn that challenged me to do some research about the relationship between DEMO and service granularity. That research was so exiting that I decided to let it be my thesis subject as well. Pascal encouraged me to write an academic article about finding an optimum in service granularity and helped me to publish it at the International Information Management Association (IIMA) conference in the Netherlands. Accordingly I asked Pascal to be my promoter for my research project. I would like to thank Pascal for his positive and critical feedback and contribution to this project.

In the spring of 2009 I started my explorative case study at Pretium Telecom in Haarlem. Thanks to the passionate drive and having an eye for finding business opportunities of my business partner and very good friend Ruben van der Zwan, director of Yenlo B.V., he arranged an introduction for me at Pretium Telecom. Together we triggered the interest of the CIO to get involved in my research project. I would like to thank Ruben for arranging this great opportunity for me and I would like to thank Pretium Telecom for their hospitality and support in my research project. As a result of this explorative case study Ruben arranged publication of my findings in two issues of a prestigious business process management journal in the Netherlands.

I would like to thank my fellow student Edward van Dipten in the Master in Informatics for his favor to match my research objective with a real life project at his company Alpha. Thanks to his involvement he gave me the opportunity to execute and validate my improving case study. Without his commitment this research would not have been completed in the way it is now.

I would like the give a special thanks to Hans Mulder for helping me during the validation process of this research project. I admire his positivism and ability to challenge people in an inspiring way. Thanks to his involvement and speed of acting the validation process of my research project became a success. Hans, really thanks for helping me out, because I was quite stuck during this phase in my research project.

I would like to thank the complete expert panel Joop, Ab, Linda, Erik, Jan, Edward, Gert-Jan and Robbert-Jan for their opinions and discussions and especially Robbert-Jan, Ab and Ton for their inspiring contribution and drive in the improving case study.

And most of all I would like to thank my girlfriend Virág for her trust, her patience and the elbow-room she has given me. Without her unwavering support I would not have been able to write this thesis.

René Wiersma, Rotterdam 24 August 2010

SUMMARY

In the past 10 years an important change has occurred in how to develop, integrate and reuse information systems. A new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox & Kreger, 2005).

However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. The quest for the right granularity is complex if the promises of flexibility and reusability must be obtained. Decisions about the level of service granularity are typically made using various heuristics (Feuerlicht & Wijayaweera, 2007) and service designers usually make a best guess as to how services will be used (Stevens, 2002). A method for service definition is important in environments where a lot of services are available and the set of available services changes over time.

The question how services should share organizational principles or how services should be modeled to obtain flexibility and be able to adopt organizational changes is not answered yet. The quest for service granularity has been addressed in many articles (De Jong & Dietz, 2010), (Feuerlicht & Wijayaweera, 2007), (Rosen, 2007), (Papazoglou & Van den Heuvel, 2006), (Foody, 2005), (Sims, 2005) but none of these sources answers the question how to define service granularity properly; neither do they provide some kind of concrete guidelines. Most sources just mention the importance of finding a right service granularity.

Finding an optimum in service granularity is the main theme of this research. This optimum must be independent of the context and applicable for all organizations. Researching the main theme is directed by the concepts and several determinants that are valid for service granularity and the utilization of the Design & Engineering Methodology for Organizations (DEMO). This objective leads to the central research question:

How to define the right granularity for services such that the determinants of granularity are balanced with the proposed business benefits of flexibility and reusability using DEMO's informational construction modeling from a service-oriented perspective?

In this research an extensive review of literature sources on service granularity and DEMO has been performed. The topic has been explored from different angles to get a complete picture of the service granularity environment.

To test how to define the right granularity of services two explorative case studies and one improving case study at organizations using the positivist case study research method have been conducted. The explorative case studies resulted in a conceptual model that reflected the process and activities that have been performed. The objective of the improving case study was to test and improve the conceptual model from the explorative case studies. The applied research approach followed the action research cycle of (Baskerville & Wood-Harper, 1996). In the action taking phase the collaborative modeling approach is applied. In total three iterations of the action research cycle were conducted in which the conceptual

framework is extended by including the results of each step. This is also known as the spiral towards understanding (Carroll & Swatman, 2000). The findings of the improved case study have been validated by an expert panel.

The process of the improving case study was aimed at exploring the concept of the granularity of services and by collaborative modeling we tried to find an optimum. The improving case study showed that the right granularity of services is not achieved out of the box. We have seen that optimizing service granularity is a process of several iterations and that reaching an optimum is something subjective. The participants of the improving case study agreed that the end result is an optimum model of information services, because of the achieved consensus about the results that could be substantiated whenever necessary and is directive to further design software.

After the improving case study the results were validated by an expert panel. The expert panel concluded that DEMO is applicable in modeling the essence of the informational organization. The DEMO infological construction model supports the modeling of services in a fashion that services could be optimized to their environment. By extending the information services of DEMO with the activities that are stated in the conceptual model, the participants concluded that a coherent and relevant set of activities is provided that contributes to optimizing service granularity.

Table of Contents

PREFACE **5**

SUMMARY **7**

1. INTRODUCTION **15**

1.1. Problem statement 15
1.2. Research questions 16
1.3. Research method 16
1.4. Literature review 19
1.5. Case studies 19
1.6. Expert panel 22
1.7. Research conclusions and recommendations 22
1.8. Practical relevance 22
1.9. Scientific relevance 23
1.10. Structure of the thesis report 24

2. SOA & GRANULARITY **25**

2.1. Overview of business drivers for SOA 25
2.1.1. Business benefits for SOA 25
2.1.2. Summary 27
2.2. Introduction in service granularity 28
2.2.1. The concepts of services 28
2.2.2. The concepts of granularity 31
2.2.3. Service granularity in this research project 32
2.2.4. Summary 33

3. STRATEGIES & TECHNIQUES TO DETERMINE SERVICE GRANULARITY **34**

3.1. Overview of methods for defining service granularity 34
3.1.1. Top-down 35
3.1.2. Bottom-up 37
3.1.3. Meet-in-the-middle 38
3.1.4. Middle-out 39
3.1.5. Granularity scheme 41
3.1.6. Summary 45
3.2. An overview of techniques to define service granularity 46
3.2.1. DEMO 46
3.2.2. UML Use Cases 47
3.2.3. UML Activity Diagrams 47
3.2.4. Petri nets 48

3.2.5. BPMN ... 48
3.2.6. BPEL4WS ... 49
3.2.7. Summary and conclusion ... 49
3.3. Getting granularity right ... 51

4. INTRODUCTION TO THE ONTOLOGY OF THE INFORMATIONAL ORGANIZATION ... 53
4.1. Why DEMO? ... 53
4.2. Information organization ... 54
4.3. PSI-theory ... 55
4.3.1. The operation axiom ... 56
4.3.2. The transaction axiom ... 57
4.3.3. The composition axiom ... 60
4.3.4. The distinction axiom ... 62
4.3.5. The organization theorem ... 64
4.4. Aspect models ... 66
4.4.1. Construction model ... 66
4.4.2. Process model ... 68
4.4.3. State model ... 69
4.4.4. Action model ... 70
4.5. Summary ... 71

5. EXPLORATIVE CASE STUDIES ... 73
5.1. Case study introduction ... 73
5.2. Construction of the information organization ... 74
5.2.1. Why the informational construction diagram? ... 75
5.2.2. Infological analysis ... 75
5.2.3. Intellect system ... 76
5.2.4. Production acts and facts ... 77
5.2.5. Infological modeling notation ... 77
5.3. Examining the case study results ... 77
5.3.1. Conceptual framework ... 77
5.3.2. Plan ... 77
5.3.3. Collect data ... 78
5.3.4. Analyze ... 78
5.3.5. Reflect ... 78
5.4. Findings ... 79
5.5. Summary ... 84

6. IMPROVING CASE STUDY ... 85
6.1. Case study introduction ... 85
6.2. Research approach ... 87

6.2.1. Participants of the collaborative session 87
6.2.2. The action research cycle 88
6.3. Summary 109

7. VALIDATION 111
7.1. Diagnosing 111
7.2. Action planning 112
7.3. Action taking 113
7.4. Evaluation 114
7.5. Learning 114

8. RESEARCH CONCLUSIONS AND RECOMMENDATIONS 118
8.1. Answers to research questions 118
8.2. Discussion 122
8.3. Recommendations for future research 126

BIBLIOGRAPHY 127

APPENDIXES 133
A. Overview service granularity determinants 133
B. Legend modeling I-organization 136
C. Alpha’s global actor transaction diagram 137
D. Alpha’s ontological models CA700 138
E. Template infological analysis 140
F. Alpha’s infological construction model 144
G. Service granularity grid 147
H. Influencing service granularity determinants 149
I. Alha’s ontological model after second iteration 153
J. Use case template 156
K. Alpha’s infological contruction model after third iteration 157
L. Authorization matrix 158
M. E-mail invitation 159
N. Set-up GDSS meeting room 160
O. GDSS propositions 161
P. Results GDSS 166
Q. IIMA Article 193

List of figures

Figure 1: Research design..18
Figure 2: The structured-case research method..20
Figure 3: Iterative action research process..21
Figure 4: Porter's Competitive Strategies (Daft, 2007)..27
Figure 5: Expressions of service types..30
Figure 6: Activities and deliverables of top-down strategy..35
Figure 7: Activities and deliverables of service-oriented analysis..36
Figure 8: Activities and deliverables of bottom-up strategy..37
Figure 9: Activities and deliverables of meet-in-the-middle strategy..39
Figure 10: Activities and deliverables of middle-out strategy..40
Figure 11: Three levels of service granularity..41
Figure 12: Application service composed of business services (McGovern, Sims, Jain, & Little, 2006)..44
Figure 13: Three activity levels in perspective..55
Figure 14: Standard notation of a coordination act..56
Figure 15: Graphical representation of the operation axiom..57
Figure 16: Basic Transction Pattern..58
Figure 17: Standard Transaction Pattern..59
Figure 18: The structure of enclosing a transaction..61
Figure 19: The structure of self-activation..62
Figure 20: The process of performing a coordination act..63
Figure 21: The layered integration of an enterprise..65
Figure 22: Enterprise aspect models of DEMO..66
Figure 23: Interaction model..67
Figure 24: Interstriction model..68
Figure 25: Process model..69
Figure 26: State model..70
Figure 27: Action rules..70
Figure 28: Conceptual model for information analysis..81
Figure 29: Generic System Development Process..83
Figure 30: Performa invokes infological transaction..89
Figure 31: Performa transforming into informa..90
Figure 32: Organization Construction Diagram and Process Structure Diagram..95
Figure 33: Improved conceptual model after the first iteration..98
Figure 34: Improved conceptual model after the second iteration..103
Figure 35: Improved conceptual model after the third iteration..107
Figure 36: Improved conceptual model after the GDSS..117
Figure 37: Legend for the interaction model of the I-organization..136
Figure 38: Global Actor Transaction Diagram of Alpha..137
Figure 39: Detailed Organization Construction Diagram of CA700 after the first iteration..138
Figure 40: Process Structure Diagram of CA700 after the first iteration..139
Figure 41: State Diagram of CA700 after the first iteration..139
Figure 42: Infological model CA700 after the first iteration..146
Figure 43: Detailed Organization Construction Diagram of CA700 after the second iteration..153
Figure 44: Process Structure Diagram of CA700 after the second iteration..154

Figure 45: State Diagram of CA700 after the second iteration 154
Figure 46: Infological model CA700 after the second iteration 155
Figure 47: Infological model CA700 after the third iteration 157
Figure 48: Invitation to participate in GDSS session 159
Figure 49: Set-up of the meeting room to facilitate the GDSS 160
Figure 50: Summarizing graph of GDSS on service granularity 166
Figure 51: Variability diagram of GDSS GR2 170
Figure 52: Variability diagram of GDSS GR5 170
Figure 53: Variability diagram of GDSS GR1 170
Figure 54: Variability diagram of GDSS GR3 170
Figure 55: Variability diagram of GDSS GR4 171
Figure 56: Variability diagram of GDSS GR6 171
Figure 57: Variability diagram of GDSS GR7 171
Figure 58: Summarizing graph of GDSS on the process of information analyses 172
Figure 59: Variability diagram of GDSS IM3 176
Figure 60: Variability diagram of GDSS IM1 176
Figure 61: Variability diagram of GDSS IM5 176
Figure 62: Variability diagram of GDSS IM7 176
Figure 63: Variability diagram of GDSS IM2 177
Figure 64: Variability diagram of GDSS IM4 177
Figure 65: Variability diagram of GDSS IM6 177
Figure 66: Summarizing graph of GDSS on the activities in the conceptual model 178
Figure 67: Variability diagram of GDSS AB1 182
Figure 68: Variability diagram of GDSS AB3 182
Figure 69: Variability diagram of GDSS AB2 182
Figure 70: Variability diagram of GDSS AB4 182
Figure 71: Summarizing graph of GDSS on the contribution of DEMO in finding an optimum in service granularity 183
Figure 72: Variability diagram of GDSS DM7 187
Figure 73: Variability diagram of GDSS DM2 187
Figure 74: Variability diagram of GDSS DM6 187
Figure 75: Variability diagram of GDSS DM4 187
Figure 76: Variability diagram of GDSS DM3 187
Figure 77: Variability diagram of GDSS DM5 187
Figure 78: Variability diagram of GDSS DM1 188

List of tables

Table 1: *SOA definitions* 25
Table 2: *Service definitions* 28
Table 3: *Classification of service granularity aspects grouped by service type (Steghuis, 2006)* 31
Table 4: *Overview of assessed criteria per modeling technique* 50
Table 5: *Transaction-result table* 67
Table 6: *Infological analysis table* 76
Table 7: *DEMO supporting information service granularity determinants* 79
Table 8: *Extended Information Use Table* 95
Table 9: *eIUT – B-A01 Registrar* 96
Table 10: *eIUT – B-A01 Registrar (complete information analysis)* 96
Table 11: *Service granularity determinant descriptions* 133
Table 12: *Service granularity grid* 147
Table 13: *Conclusions of the service granularity grid* 149
Table 14: *Authorization matrix for actor transformations* 158
Table 15: *GDSS propositions about service granularity* 161
Table 16: *GDSS propositions about the process of information analyses* 162
Table 17: *GDSS propositions about activities in the conceptual model* 163
Table 18: *GDSS propositions about the contribution of DEMO in finding an optimum in service granularity* 164
Table 19: *GDSS propositions for the expert panel discussion* 165
Table 20: *Summarizing table of GDSS on service granularity* 167
Table 21: *Summarizing table of GDSS on the process of information analyses* 173
Table 22: *Summarizing table of GDSS on the activities of the conceptual model* 179
Table 23: *Summarizing table of GDSS on the contribution of DEMO in finding an optimum in service granularity* 184

1. INTRODUCTION

1.1. Problem statement

In the past 10 years an important change has occurred in how to develop, integrate and reuse information systems. A new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox & Kreger, 2005). The SOA concept benefits business because IT can be developed and implemented much faster and at lower development costs. Also it makes organizations processes more adaptable to change. Even the definition of Service-Oriented Architecture points at the common benefits of flexibility and reusability. Regarding the aspect of flexibility Weske (2007) refers to the definition of Burbeck (2000) that services gain flexibility by runtime coupling with the service registry. This dynamic coupling of services is not reached most of the times. Contrary to the definition of Burbeck (2000), Weske (2007) states better to speak about enterprise services. Reuse is reached if a service contains functionality with a clear business value and can be used directly (Weske, 2007).

Besides this SOA also requires that organizations evaluate their business models to fit service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner, customer, and supplier relationships (Papazoglou & Van den Heuvel, 2006). However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. The quest for the right granularity is complex if the promises of flexibility and reusability must be obtained. Decisions about the level of service granularity are typically made using various heuristics (Feuerlicht & Wijayaweera, 2007) and service designers usually make a best guess as to how services will be used (Stevens, 2002). A method for service definition is important in environments where a lot of services are available and the set of available services changes over time. Burbeck (2000) states services to be grounded on shared organizational principles. These principles makes sure services operate without errors, support flexibility and can be joined together to fit in business processes.

The question how services should share organizational principles or how services should be modeled to obtain flexibility and be able to adopt organizational changes is not answered yet. The quest for service granularity has been addressed in many articles (De Jong & Dietz, 2010), (Feuerlicht & Wijayaweera, 2007), (Rosen, 2007), (Papazoglou & Van den Heuvel, 2006), (Foody, 2005), (Sims, 2005) but none of these sources answers the question of how to define service granularity properly; neither do they provide some kind of concrete guidelines. Most sources just mention the importance of finding a right service granularity.

While Steghuis (2006) and others state the importance of optimum service granularity this research tries to answer the HOW. By using the Design & Engineering Methodology for Organizations (DEMO) (Dietz, 2006a) this research tries to discover a guideline for finding an optimum service granularity on the informational level of organizations.

1.2. Research questions

The main objective of this research is the quest of finding an optimum in service granularity. This quest is guided by the concepts and several determinants that are valid for service granularity and the utilization of the DEMO methodology. This objective leads to the main research question:

How to define the right granularity for services such that the determinants of granularity are balanced with the proposed business benefits of flexibility and reusability using DEMO's informational construction modeling from a service-oriented perspective?

The following sub questions are formulated, supporting answering the main research question.

1. What are the main business drivers for service-oriented architectures?
2. What are the concepts and determinants of service granularity?
3. What methods are available to define service granularity?
4. What models are used to express service granularity?
5. What is a right[1] service granularity?
6. What is the DEMO informational construction model?

In order to answer these questions and eventually have a best practice how to define right service granularity, the approach has to be tested. The research design and methodology strives to facilitate that objective.

1.3. Research method

The Information System Research Framework of Hevner, March, Park, & Ram (2004) has been chosen as a starting point in developing a guideline for finding an optimum in service granularity. This is based on the fact that Hevner et al. (2004) propagate that studies in the IS research domain contain both descriptive and prescriptive research.

The descriptive part of the research (knowledge-producing activity) aims to understand and explain how service granularity is defined, while the prescriptive approach (knowledge-using activity) aims at improving service oriented architecture (March & Smith, 1995); (Hevner et al., 2004).

This research consists of two major activities based on the framework. First a literature study of existing research was conducted (the knowledge base). Based upon this study it was decided to use the DEMO method as a foundation to develop this research's guideline on defining service granularity.

Secondly by using and extending the DEMO method in three case studies the guidelines on how to define optimum service granularity were developed. In the framework of Hevner et al. (2004) these activities are related to the "environment" and "develop/build" aspects.

The research method is presented in figure 1 on page 18. It includes the following activities and deliverables. The deliverables are prefixed with a "D" and a sequence number.

[1] In this thesis a right service granularity is synonym with an optimum service granularity.

In D1 the research problem is specified as well as the research objectives and the research questions that substantiate the research objectives.

In D2 until D7 extensive review of literature sources on service granularity and DEMO is discussed. To get a complete picture of the service granularity environment, the topic is explored from different angles.

Guided by the case study structure of Bryman & Bell (2007) two explorative case studies and one improving case study have been executed at organizations using the positivist case study research method, in parallel with the literature review. The first explorative case study has been executed at Pretium Telecom in Haarlem, the Netherlands. The second explorative case study is about a fictitious value chain as part of an assignment at the HU University of Applied Sciences in Utrecht, the Netherlands. The improving case study is conducted at a national government organization which is responsible for the coordination and operation in big international projects. The case study has been performed in a collaborative fashion. With collaborative modeling the different views of the actors on the research subject could be integrated easily, since collaborative modeling has proven to be a successful approach (Stirna & Kirikova, 2008). For confidentiality reasons the company name including the names of the attendees and the name of the projects are scrambled. The name “Alpha” is used when this thesis refers to this company.

After finalizing the literature review and the case studies the results and findings so far were assessed by experts via an expert panel review in D10 of this research design. These experts carefully assessed questions about the importance of granularity, the research approach, the findings and the applicability of DEMO in correlation with service granularity.

D11 led to final conclusions, discussions and recommendations derived from the research. Further research is suggested.

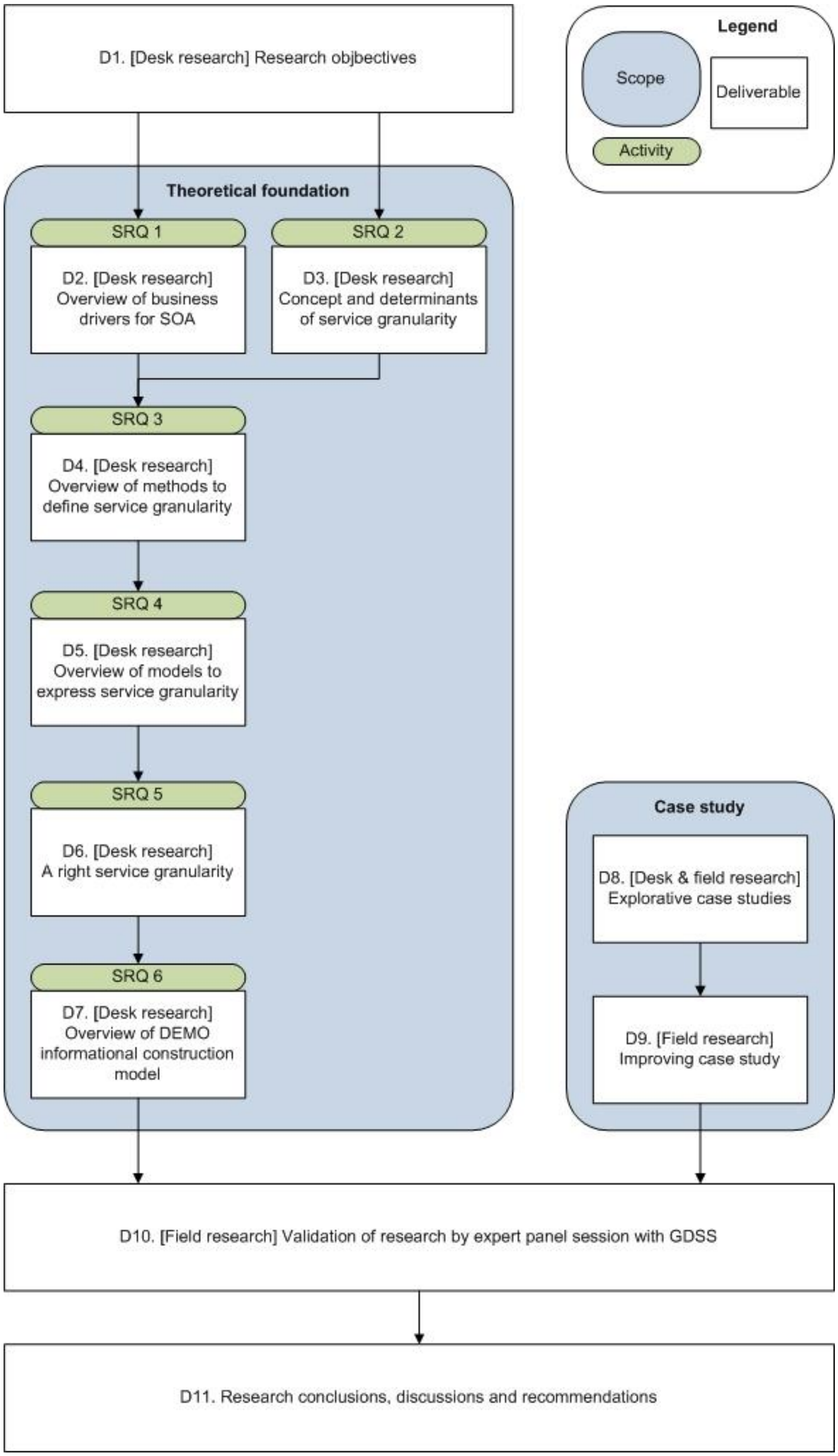


Figure 1: Research design

1.4. Literature review

The research objectives determined what approaches and theories were selected from the literature available on the subject of this thesis. The literature review is used as a predominant factor in the entire research project, as a source of theory creation, phenomena observation and the search for new insights and new perspectives that could influence the research process. The literature review process constantly reflected on new insights in the context of already existing and new literature on this research topic. A bibliography of relevant literature is enclosed in the bibliography paragraph.

1.5. Case studies

To extend the literature findings, three case studies were conducted to investigate in practice how an optimum in service granularity could be found by using the informational construction model of DEMO. The case study research approach has been chosen because the topic of this thesis is tightly related to software engineering. Since software engineering is a multidisciplinary area it also involves an area where case studies are normally conducted (Runeson & Höst, 2009).

Given the fact that many literature sources mention the importance of a right granularity and none of reviewed literature sources that are used in this research explains how that can be obtained, it seems appropriate to apply a qualitative approach to the research question. Specifically, an approach based on the structured case method is advised by Carroll & Swatman (2000) and Yin (2003) for the explorative case studies and apply the action research approach (Baskerville & Wood-Harper, 1996) for the improving case study. These methods are used for several reasons: (i) it was found particularly well-suited for Information System (IS) research situations in which an in-depth investigation is needed, but in which the phenomenon in question cannot be studied outside the context in which it occurs, (ii) it offers a great deal of flexibility in terms of research perspectives to be adopted and qualitative data collection methods, and (iii) case studies open up opportunities to get the subtle data needed to increase the understanding of complex IS phenomena.

Explorative and improving case studies are one of the four types as defined by Robson (2002) and adjusted by Runeson & Höst (2009). These four types are:

- Exploratory; finding out what is happening, seeking new insights and generating ideas and hypotheses for new research.
- Descriptive; portraying a situation or phenomenon.
- Explanatory; seeking an explanation of a situation or a problem, mostly but not necessary in the form of a causal relationship.
- Improving; trying to improve a certain aspect of the studies phenomenon.

The two explorative case studies were used to formulate the hypotheses in this research. It helped to get an understanding of what is the actual state in the area of service granularity and what is the contribution of DEMO in this area. Exploring the area in two case studies resulted in ideas about the conceptual framework, which is the foundation within the improving case study. The improving case study continues the research of finding an optimum service granularity. The basic assumptions for the improving case study are the findings of the explorative case studies. The cohesion of both the two types of case studies is shown later in this paragraph.

Klein & Myers (1999) define three types of case study depending on the research perspective; positivist, critical and interpretative. The case studies in this research use the positivist case study research method, because the studies have a typically exploratory nature.

For the explorative case studies the structured case method has been applied. The structured case method provides a framework that includes five phases, which are thoroughly described and aims to build theory in a rigorous manner (Carroll & Swatman, 2000). The structured case research method of Carroll & Swatman (2000) is shown in figure 2.

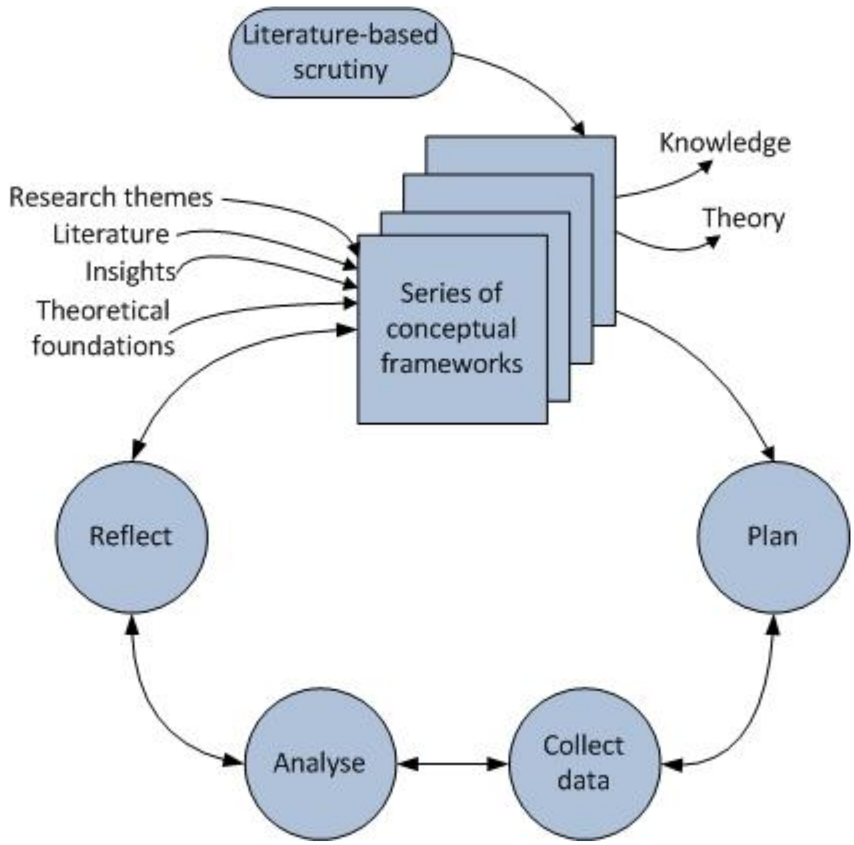


Figure 2: The structured-case research method (Carroll & Swatman, 2000)

For the improving case study the action research approach has been applied. Action research is closely related to case studies (Runeson & Höst, 2009). To structure the case study the action research cycle from Baskerville & Wood-Harper (1996) is used. The method distinguishes the following activities (1) diagnosing, (2) action planning, (3) action taking, (4) evaluating, and (5) specifying learning. See figure 3. A collaborative modeling approach for the case study's action taking phase is followed. Collaborative modeling can be defined as the joint creation of a graphical representation of a system or process (Renger, Kolfschoten, & de Vreede, 2008). Barjis, Kolfschoten, & Verbraeck (2009) distinguish two perspectives on collaborative modeling; system perspective and process perspective. The system perspective supports problem solving and changing organizations. The process perspective concentrates on supporting development of process models.

Richardson & Andersen (1995) described five essential roles that should be present in collaborative modeling: the facilitator, modeler / reflector, process coach, recorder and gatekeeper. In this approach the modeler and facilitator constructs the model in dialogue with the group. The recorder and process coach assists the facilitator in technology support and in dynamics of individuals and subgroups. Finally the gatekeeper is the medium between the facilitation / modeling team and the participants from the organization.

A key objective in collaborative modeling is sense making (Barjis et al., 2009). In order to create overlap in knowledge, participants need not only share information about the model elements and relations, they also need to create shared meaning with respect to these elements and their relations. Barjis et al. (2009) says that sense making usually requires some development of shared meaning of concepts, labels and terms. It also includes the development of a common understanding of context and the perspective of different stakeholders with respect to the model. The dialogue between modeler and stakeholders is critical in this process, to translate the tacit integrated perspective on the business process to a modeling language Barjis et al. (2009).

Action research is used more often in collaborative modeling research as it requires modelers (researchers) to make an active intervention in a group of participants, to study the effect (Morton, Ackermann, & Belton, 2003). As modeling requires highly advanced skills, it is difficult to train others to make this intervention to observe this effect. The participants of the workshop sessions were operational business architects and technical system architects. This enabled the workshops to discuss about granularity from a business perspective as well as from an IT perspective.

The explorative case studies provided the conceptual framework for the improving case study research. The research process is an iterative process in which the conceptual framework is extended by including the results of each step. See figure 3. This is also known as the spiral towards understanding (Carroll & Swatman, 2000). In this research project, the first cycle of the process is passed through. The research starts with the conceptual framework from the explorative case studies and adapts this according to the improving case study results.

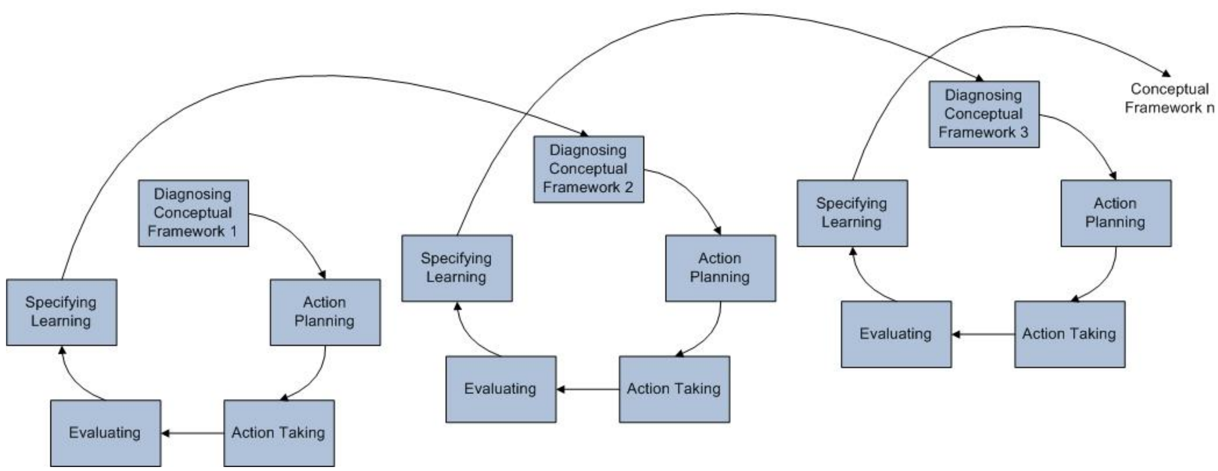


Figure 3: Iterative action research process

1.6. Expert panel

Parallel to the constant review of literature the findings and methods are tested in expert panel interviews. An expert panel of eight members is brought together. Some of these experts are seen as authorities in the area of DEMO and SOA from a practitioner's point of view as well as from an academic point of view. The other participants of the expert group are experts in the field of systems analysis and systems engineering at Alpha. To measure, record and validate their input an advanced supporting system called Meetingworks GDSS[2] was used.

Group Support Sessions (GSS) are an active way of qualitative research. The objective of a GSS is to:

- Collect data and knowledge;
- Exchange data and knowledge;
- Analyze data and knowledge;
- Manupilate data and knowledge;
- Judge data and knowledge;
- Make choices on data and knowledge.

A GSS is applicable in diverse forms of research, like brainstorming or evaluation sessions. A GSS can electronically be supported by a software system. This system can be used during the actual GSS meeting where it contributes in facilitating the electronic and verbal communications. The GSS meeting is prepared by creating an agenda and uploads that agenda into the system. The agenda exists of several formulated steps that are elaborated on during the GSS.

1.7. Research conclusions and recommendations

A wealth of data was collected and interpreted to answer the research questions. During the gathering of information, new insights were observed that formed the input for the final recommendations. A number of recommendations for further research into the testing and improvement of the usability of framework are suggested in order to further optimize service granularity and the application of DEMO in realizing software systems.

1.8. Practical relevance

Organizations who are implementing an architecture based on services will face the quest of granularity sooner or later. The quest for the right granularity is complex if the promises of flexibility and reusability must be obtained. Most of the times a senior technical specialist answers this question on gut feeling. This bottom up approach, starting on a technical level, is not aligned with the vision of achieving business benefits while developing business processes.

When information services are not optimized, organizations will face difficulties to change as fast as their surrounding environment changes. This could lead to competitive disadvantage, reducing market share or even damage of image. Therefore the informational service model must be able to adjust as easy

[2] Group Decision Support System. See http://www.meetingworks.nl

as business changes. This is why most of the authors on service-oriented architectures mention the aspect of getting granularity right. Although this is a very difficult question, organizations will benefit if there is a model available that helps in getting the granularity right.

1.9. Scientific relevance

Much literature that is referenced in this research refers to the topic of granularity, without explaining what it is, how it can be obtained and how it contributes to the goals of SOA.

The definition of service-oriented architecture points at the common benefits of flexibility and reusability. Regarding the aspect of flexibility, Weske (2007) refers to the definition of Burbeck (2000) that services gain flexibility by runtime coupling with the service registry. This dynamic coupling of services is not reached most of the times. Contrary to the definition of Burbeck (2000), Weske (2007) says that it is better to speak about enterprise services. Reuse is reached if a service contains functionality with a clear business value and can be used directly (Weske, 2007). Service-oriented architectures are important in environments where a lot of services are available and the set of available services changes over time. Services should be grounded on shared organizational principles (Burbeck, 2000). These principles makes sure services operate without errors, support flexibility and can be joined together to fit in business processes.

The question how services should share organizational principles or how services should be modeled to obtain flexibility and be able to adopt organizational changes is not answered yet. Steghuis (2006) and extensive literature research in this thesis found that the quest for service granularity has been addressed quite some times in many articles and other sources (De Jong & Dietz, 2010); (Hoogervorst, 2009); (Feuerlicht & Meesathit, 2004), (Foody, 2005); (Papazoglou & Van den Heuvel, 2006); (Sims, 2005); (Zimmerman, Krogdahl, & Gee, 2004); (Rosen, 2007). None of these sources answered the question of service granularity properly or provide some kind of concrete guidelines. Most sources just mention the fact of finding a right service granularity.

Steghuis (2006) did research for service granularity within service-oriented architectures. Steghuis' research provided a framework with several aspects of service granularity per business layer. In the conclusion of her master's thesis she formulates some further research questions. One of these questions is about how to get granularity right, using the aspects of granularity found by her research. Steghuis (2006) explains that although the framework for service granularity helps, it still is not a concrete guideline.

Mulder (2006) has concluded, grounded in extensive research, that DEMO is an eligible methodology to (re)design organizations in an integrated way. Because a right grained service model supports the information requests of an organization at best, this thesis uses DEMO's informational construction modeling as the intermediate layer between business processes and IT. This is where information is created and provided to the business process. The informational construction diagram has not been explored widely. Some studies are currently in progress, for example by (De Jong, 2009).

The main contribution of this research project to the Body of Knowledge is in the field of service granularity, service-oriented architecture and DEMO.

1.10. Structure of the thesis report

The paragraph gives an overview of the structure of the thesis report.

The literature study is discussed in chapters 2, 3 and 4. Chapter 2 discusses the business benefits of SOA. The most important business drivers are captured from SOA definitions that are given by authorities in the field. Furthermore service granularity is introduced. The subject is illustrated by the concepts of services and the concepts of granularity. A definition of what service granularity is in this research is also given. Chapter 3 gives an overview of the strategies for defining service granularity. Literature research on this topic learned that four strategies for service identification can be distinguished. After illustrating the strategies for service identification the granularity scheme is discussed. The granularity scheme defines three levels of granularity that guides towards a clear level of granularity for services. Furthermore the chapter discusses several models and techniques to define granularity. Chapter 4 introduces the ontology of the informational organization. This research project is founded on the underlying methodology of this theory.

Chapter 5 discusses the two explorative case studies that have been conducted for this research projects. The explorative case studies brought a conceptual model how to define the right service granularity that is aligned with the central research question.

Chapter 6 discusses the three iterations of the improving case study that have been conducted to improve the conceptual model from the explorative case studies. First the research method is described, after which the diagnosing, action planning, action taking, evaluation and learning over all iterations are carried out and the improved conceptual model is presented.

Chapter 7 presents the validation of the research, in which the coherence of the research is described and the consensus is reviewed by the expert panel. At last, the conclusions and future work are discussed in chapter 8.

2. SOA & GRANULARITY

This chapter discusses sub questions 1 "*What are the main business drivers for service-oriented architectures?*" and 2 "*What are the concepts and determinants of service granularity?*" supported by a profound foundation in literature. Answering these sub questions contributes to the understanding of the central role that services play in this thesis. At first in paragraph 2.1 an overview is provided about the business drivers for SOA. This paragraph illustrates the business benefits when the SOA philosophy is embedded in the organization and what kind of advantages and organization will face when service granularity is optimized. The most important business drivers are captured from SOA definitions that are given by authorities in the field. Later, in paragraph 2.2, service granularity is introduced. The subject is illustrated by the concepts of services in paragraph 2.2.1 and the concepts of granularity in paragraph 2.2.2. Finally in paragraph 2.2.3 is defined what service granularity is in this research. To close every main paragraph a summary is given at the end of each main paragraph.

2.1. Overview of business drivers for SOA

The concepts of services and granularity, including strategies how to determine granularity of services are part of the SOA way of thinking. Therefore the business drivers for SOA are introduced before the concepts of services and granularity in particular are illustrated.

2.1.1. Business benefits for SOA

SOA is no commercial-of-the-shelf product. It is a concept, a way of thinking how to create and manage systems, supporting business processes in an optimum way. The services minded philosophy should support this. Thinking in services must enable reusability of functionality and defining an orchestration of activities that is flexible for business process changes (Wiersma, 2009c).

SOA is developed to address a business need to make IT more responsive and adaptive to the constantly changing business conditions (Steghuis, 2006). It promises advantages like leveraging existing assets, simplifying integration and managing complexity, being more responsive, having a faster time-to-market, reducing costs, and increasing reuse of software components (Endrei, et al., 2004).

In literature many definitions are found for SOA. All these definitions carry several benefits of SOA.

Table 1 elaborates some of these definitions.

Table 1: *SOA definitions*

Reference	SOA definition
(Weske, 2007)	Service-oriented architectures are software architectures that provide an environment for describing and finding software services, and for binding to services.
(Papazoglou M. , 2003)	SOA is a way of reorganizing a portfolio of previously siloed software applications and support infrastructure into an

Reference	SOA definition
	interconnected set of services, each accessible through standard interfaces and messaging protocols.
(Lankhorst, 2005)	SOA represents a set of design principles that enable units of functionality to be provided and consumed as services.

From the definition of Weske can be learned that SOA supports a heterogeneous environment in an interoperable fashion based on services. A heterogeneous environment is able to integrate applications which are part of a business process that cross departments. With SOA the business can overcome the barrier of integrating products from multiple vendors and across different platforms (Endrei, et al., 2004). The definition of Papazoglou also points at integration of a heterogeneous environment that is based on services, but adds that integration is standardized. A standardized integration requires a standardized interface protocol that hides the business logic behind it. From the business process point of view the exposed functionality is treated like a black box. With a standardized interface and an orchestration of services that works together as the business process then changes in business process execution is easy to implement. Besides flexibility, services can also be reused in multiple business processes. Orchestration of services becomes a set of connected services in a certain order that together forms the business process and aims to realize business goals (Weske, 2007).

The definition of Lankshorst says that SOA is not just a bunch of services connected to each other, but that the orchestration of services must be well structured and guided by organizational and information technology design principles. The structure of services must be sound, enabling a certain unit of functionality to the business that contains business value for the service provider as well as the service consumer. The service requester and service consumer pattern is substantiated by the required interaction between at least two actors who are interchanging information to perform essential business activities that are grounded in their organizational responsibilities.

Endrei, et al. (2004) says that IT executives in today’s business face two pressures: heterogeneity and change. These pressures are caused by the range of different systems, applications and architectures of different ages and technologies. This application landscape must be kept operational and continuously changed. The pressure to change is driven by globalization, increased competition or the ambition to stay ahead of competition, cost reduction, changed organizational strategy, etc.. There are numerous arguments to come up with why organizations change. For example, in case the strategy of the organization changes this has also an effect on the desired degree of flexibility in business processes. For example a low-cost leadership strategy from Porter (Daft, 2007), which has an orientation on efficiency, focuses always on standardization, so the processes of these kinds of organizations are more in the left half of Porter’s Competitive Strategies model and are less flexible. In contrast to the differentiation strategy from Porter (Daft, 2007), which has a learning orientation, and focuses on flexibility, so its processes will be more on the right side of Porter’s Competitive Strategies model. See figure 4 for Porter’s Competitive Strategies model.

Figure 4: Porter's Competitive Strategies (Daft, 2007)

The IT department needs to adapt to changes as quickly as possible and could not always provide the level of service that was required. SOA promises to release those pressures and supports flexibility in business processes and increased reusability of services.

2.1.2. Summary

This main paragraph provided an overview of the main business drivers for SOA. Embedding a SOA in organizations is mostly driven from two pressures on IT departments. The first pressure is caused by the heterogeneous environment. All applications, systems and other IT solutions must interact together to support the business processes. This IT landscape with a variety of components is difficult to maintain without a standardized integration protocol. The second pressure is caused by the constantly changing business. The IT environment must be adaptive and flexible to support the required changes.

SOA is able to release those pressures. The various definitions of SOA all points at the main advantages: flexibility in business processes and reusability of services within a heterogeneous environment.

Instead of creating the most flexible and reusable SOA, one has to consider the main business strategy. The SOA must be supportive to that strategy and not become a goal on its own.

2.2. Introduction in service granularity

Many studies refer to the importance of service granularity when software systems are designed. The introduction already mentioned several studies about service granularity. The common gap in all these studies is a description of a methodological approach of how to solve the quest for getting the granularity right. To get a better understanding this main paragraph addresses service granularity. Both words of the term carry specific meaning within the information technology science. Therefore both words are illustrated separately. The first paragraph addresses services in its many forms. The second paragraph discusses the topic of granularity and mentions the several aspects of granularity. The third paragraph states the utilization of the concepts in this thesis. The last paragraph briefly summarizes all this.

2.2.1. The concepts of services

Services are ambiguously defined. There exists no literature that holds the one and only definition of a service worldwide. This means that defining what a service exactly is about is not that common (Arsanjani, Ghosh, Allam, Abdollah, Ganapathy, & Holley, 2008). Looking at the world outside of IT, one sees different kinds of services that are provided by several companies. For example, one can order groceries online at Albert Heijn. The service Albert Heijn provides is in essence the order picking and delivery of the ordered goods at a specific address. But one can also say that the internet application that allows the customer to order groceries is also a service. Another example of a service is issuing money on the consumer market for cash transactions by the national bank. The national bank provides that service to companies who are dealing with consumers who are paying in cash and need money in return. A third example of a service is lecturing at a University. Lecturing provides intangible services to clients, in this case students. Substantiated by these examples, one cannot define a service, because the context and the purpose of the service differ in almost all occasions.

Looking at IT again, services in IT share the same characteristics as services in non-IT environments. In both business science and computer science a service is regarded as an interaction between a requesting party (often called consumer of customer) and an offering party (often called provider or supplier) (Terlouw & Albani, 2010). Despite the common notion of a service, the context and purpose differ for the usage in supporting a business goal. To give a clue on service definitions in IT, table 2 elaborates on several service definitions that exist.

Table 2: *Service definitions*

Reference	Service definition
(Krafzig, Banke, & Slama, 2005)	A service is a software component of distinctive functional meaning that typically encapsulates a high level business concept.
(Brown, Delbaere, Eeles, Johnston, & Weaver, 2005)	A service is generally implemented as a coarse grained, discoverable software entity that exists as a single instance and interacts with applications and other services through a loosely

Reference	Service definition
	coupled (often asynchronous), message-based communication model.
(Papazoglou M. , 2003)	Services are self-describing, platform-agnostic computational elements that support rapid, low-cost composition of distributed applications. Services perform functions, which can be anything from simple requests to complicated business processes.
(Lankhorst, 2005)	A service is a unit of functionality that some entity makes available to its environment, and which has some value for certain entities in the environment.
(The Open Group, 2006)	A service is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports), is self-contained, may be composed of other services, and is a “black-box” to consumers of the service.
(Arsanjani, Ghosh, Allam, Abdollah, Ganapathy, & Holley, 2008)	From a business perspective, a service is a well-defined, encapsulated, reusable, business-aligned capability. From an information technology (IT) perspective, a service is a discoverable, invokable software resource that has a service description and interface and is configurable using policies.
(Terlouw & Albani, 2010)	A service is a universal pattern of coordination and production acts, performed by the executor of a transaction for the benefit of its initiator, in the order as stated in the standard pattern of a transaction. When implemented it has the ability • to get to know the coordination facts produced by the initiator and • to make available to the initiator the coordination facts produced by itself.

Summarizing from table 2 with service definitions one can state that a service encapsulates a specific unit of functionality, which is exposed to a business process and invoked by an interaction between a requesting party and an offering party. The service hides the heterogeneity of the business logic by its interface and the description of the service needs to include its meaning and intent.

Services exist on multiple layers, supporting business processes, information needs or infrastructure solutions. The layered appearances of services are also defined in literature as service classifications. Wiersma (2009b) summarized different service classifications of several authors. Papazoglou & Van den Heuvel (2006) classifies service types in Business Services, Infrastructure Services and Component Based Service Realizations. See picture on the left in figure 5. Erl (2005) classifies service types in Orchestration Services, Business Services and Application Services. See lower right picture in figure 5. Schekkerman (2004) classifies Business of Organization Services, Information (System) Services, and Technology Infrastructure Services as part of the Capgemini Integrated Architecture Framework. See upper right picture in figure 5.

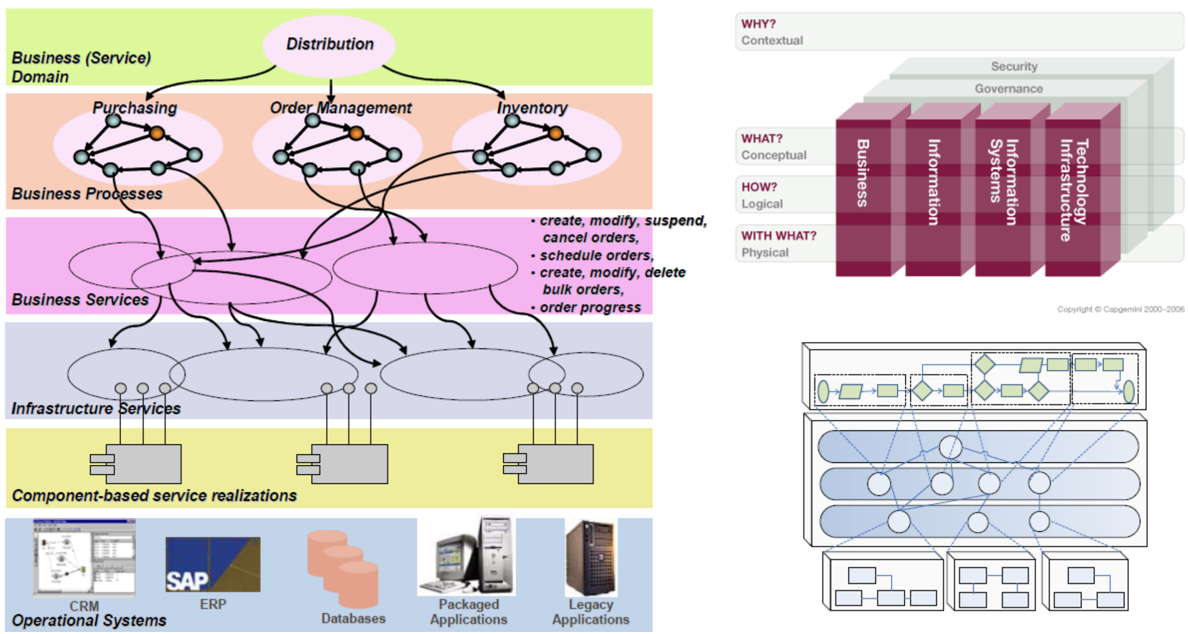


Figure 5: Expressions of service types

Terlouw & Albani (2010) wrote that six different types of services exist. There are human services that are executed by human beings, and there are IT services which are services executed by IT systems. Those two types of services are applicable per aspect type of the organization. This means that the services are applicable on the business layer, the information layer, and the data layer. Aalst & Hee (2008) defines not two types of services, but they define three types of services. They state that services are not primarily executed by human beings or IT systems, but the combination is possible as well. These so-called semi-automatic services consist of a human being service execution part as well as an IT-systems execution part.

2.2.2. The concepts of granularity

Granularity is a term that reflects the degree of modularity of a system. Papazoglou & Van den Heuvel (2006) define service granularity as the unit of modularity of services. It is the amount of functionality that is exposed by a service. There exist two types of granularity of services. Fine grained services typically implement a single atomic operation and exchange limited amounts of data. Coarse grained services implement high-level business functions (Feuerlicht & Wijayaweera, 2007). It follows that fine grained services are highly reusable as they encapsulate simple functions that are readily reused (Papazoglou & Yang, 2002). However when designing services one must also consider the impact of using services over the internet and deal with the design constraints that this environment imposes. Such considerations include network latency and reliability and lead to a preference for coarse grained services that minimize the number of interactions needed to implement a given business function, reducing the complexity of the message interchange dialogue (Feuerlicht, 2006).

Dow, Ravesteyn, & Versendaal (2008) discussed a new trend in the SOA market that led to more granular services, called enterprise services. An enterprise service is a special type of web service where the operations form a functional piece (steps or tasks) of a business process. Enterprise service operations may be composed of more fine-grained web services which provide business-agnostic functionality, such as basic data access. This means that enterprise services are larger grained services that are compositions of smaller grained components or other artifacts (Papazoglou & Van den Heuvel, 2006). In the study of Dow, Ravesteyn, & Versendaal (2008) it has also been said that coarse granularity can be defined as such that services are related to the individual steps of a business process.

The composition of fine grained services into larger grained services can facilitate reuse when the concepts from component-based development are adopted (Yang, 2003). A composite web service can be used to implement complete business processes that consist of activities, flow control, data flows and process definitions. With Service Component Architecture (SCA), which is an industry effort by BEA, IBM and Oracle (BEA, IBM, & Oracle, 2005), components can be assembled into business applications by connecting the components via service references. The SCA approach combines the advantages of fine grained services with the advantages of coarse granularity services (Feuerlicht & Wijayaweera, 2007).

Steghuis (2006) concluded that several types of granularity exist for different kind of services. The different types of services are mentioned in paragraph 2.2.1. Based on literature study Steghuis (2006) made a classification of the aspects of granularity and drew it on the several service types. Table 3 shows the classification of aspects of service granularity grouped by service type. The granularity aspects are illustrated in appendix A.

Table 3: *Classification of service granularity aspects grouped by service type (Steghuis, 2006)*

Business Service	Information System Service	Software Service
Functionality	Functionality	Functionality
Flexibility in Business	Flexibility in Business	

Business Service	Information System Service	Software Service
processes	processes	
Problem Complexity	Cognitive and Structural Complexity	Cognitive and Structural Complexity
Reusability	Reusability	Reusability
Composability	Composability	Composability
	Reusability of Legacy	
Sourcing	Sourcing	
Genericity	Genericity	Genericity
Context-independence	Context-independence	Context-independence
	Performance	Performance

2.2.3. Service granularity in this thesis

Services are a cornerstone in service-oriented architectures. Lankhorst (2005) says that the service concept is the result of a separation of the "external" and "internal" behavior of a system. As such, it should be self-contained and have a clear purpose from the perspective of its environment. This research project adopts the service definitions of Lankhorst (2005) and Terlouw & Albani (2010). The combination of both definitions addresses the interaction of a service invocation as well as the intention of the service.

For the concept of service types this thesis distinguishes the service types Business Services, Information Services, and Data Services. The definition of these service types are aligned with several architecture frameworks, like the Integrated Architecture Framework of Capgemini and also with the Enterprise Engineering Framework (Op 't Land, De Jong, & Goedvolk, 2008).

Whenever this thesis refers to services and the context does not provide enough information about what type of service is mentioned, then the refered service is considered to be an information service of the automated – or semi-automated type, according to Aalst & Hee (2008).

2.2.4. Summary

This main paragraph illustrated the several concepts of services and granularity. Several definitions are given for a service, which concludes that services are ambiguously defined. Despite that there is a certain overlap in the definition of services; there exists no common definition within the information technology science. This research project adopts the service definitions of Lankhorst (2005) and Terlouw & Albani (2010). The combination of both definitions addresses the interaction of a service invocation as well as the intention of the service.

Granularity is about the unit of modularity of services. There exist two types of granularity of services. Fine grained services typically implement a single atomic operation and exchange limited amounts of data. Coarse grained services implement high-level business functions. With enterprise services, coarse grained services can be composed of more fine grained services. The composition of fine grained services into larger grained services can facilitate reuse and flexibility in orchestrating business processes.

Several aspects are applicable when service granularity is considered. These aspects are classified business services, information system services and software services, based on literature study of Steghuis (2006).

3. STRATEGIES & TECHNIQUES TO DETERMINE SERVICE GRANULARITY

This chapter discusses sub questions 3 "*What methods are available to define service granularity?*", 4 "*What models are used to express service granularity?*" and 5 "*What is a right service granularity?*" supported by a sound foundation in literature. At first in paragraph 3.1 an overview of the strategies for defining service granularity is provided. Extensive literature research learned that four strategies can be distinguished. After illustrating the strategies for service identification the granularity scheme is discussed in paragraph 3.1.5. The granularity scheme defines three levels of granularity that guides towards a clear level of granularity for services. The next main paragraph 3.2 provides an overview of models to define granularity using techniques that follow a top-down strategy, are well known in the information technology science or are massively utilized in today's IT projects. The last main paragraph 3.3 discusses what must be in place to get granularity right. To close every main paragraph a summary is given at the end of each main paragraph.

3.1. Overview of methods for defining service granularity

In the absence of a comprehensive design methodology decisions about the level of service granularity are typically made using various heuristics (Feuerlicht & Wijayaweera, 2007) and service designers usually make a best guess as to how services will be used (Stevens, 2002). Schmelzer (2006) suggests that granularity can be determined by studying the extent to which a service can be reused and proposes a design process for services which combines top-down and bottom-up approaches. Keen, et al. (2004) and Stevens (2002) suggests that a façade design pattern can be used to construct coarse grained services from fine grained services in order to minimize the number of message interchanges, which is a bottom-up centric approach. Terlouw, Terlouw, & Slinger (2009) provided an overview of service identification methods that are described in most detail and are published in articles. These are the *Service-Oriented Architecture Framework* (SOAF) (Erradi, Anand, & Kulkarni, 2006), developed by the Indian consulting company Infosys, *Service-Oriented Modeling and Architecture* (SOMA) as proposed by IBM (Arsanjani, Ghosh, Allam, Abdollah, Ganapathy, & Holley, 2008), and the method of Papazoglou and Van den Heuvel (Papazoglou & Van den Heuvel, 2006). Terlouw, Terlouw, & Slinger (2009) classifies SOAF (or, more precisely: its so-called execution view) as a meet-in-the-middle strategy for service identification. SOMA utilizes aspects of multiple service identification strategies, i.e. the top-down, bottom-up, and middle-out strategy. The method from Papazoglou and Van den Heuvel specifies in a large amount of detail how to execute certain activities, e.g. service interface specification and service deployment. For the realization of processes and services, they explain how to apply top-down, bottom-up, as well as meet-in-the-middle strategies.

The classification of several service identification strategies is also supported by Dow, Ravesteyn, & Versendaal (2008). They state that choosing a strategy is a key decision in the creation of services.

When services are identified the granularity of services must be defined. Without this, it becomes difficult to define how services are composed, and how the various scalability aspects are taken into account (McGovern, Sims, Jain, & Little, 2006). The granularity scheme of Herzum & Sims (2000) defines three levels of granularity that guides towards a clear level of granularity for services.

This main paragraph discusses the four service identification strategies that can be applied. Paragraph 3.1.1 addresses the top-down approach. Paragraph 3.1.2 illustrates the bottom-up strategy. Paragraph 3.1.3 discusses the meet-in-the-middle strategy, and paragraph 3.1.4 addresses the middle-out approach. When the four service identification strategies are illustrated the granularity scheme is briefly introduced in paragraph 3.1.5. The last paragraph summarizes this main paragraph.

3.1.1. Top-down

The top-down strategy is an "analysis" first approach; this means it starts with defining one or more enterprise business models (Terlouw, Terlouw, & Slinger, 2009). These models can either represent the current or desired state of business operations. Figure 6 (Terlouw, Terlouw, & Slinger, 2009) depicts the activities and deliverables of the top-down strategy as described by (Erl, 2005).

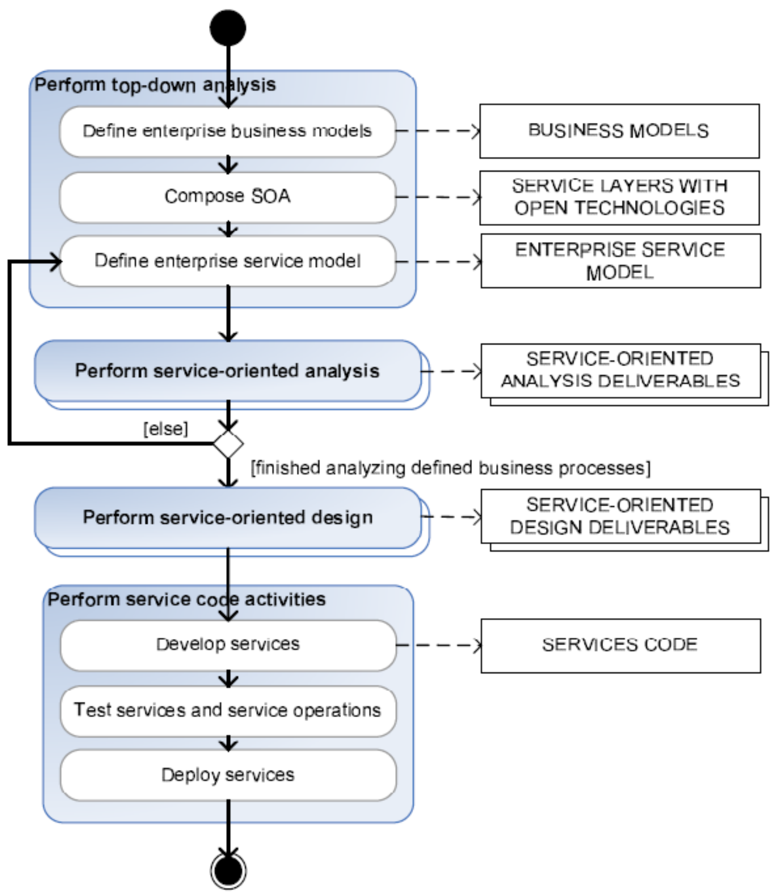


Figure 6: Activities and deliverables of top-down strategy

Figure 7 (Terlouw, Terlouw, & Slinger, 2009) exhibits the decomposition of the service-oriented analysis, which includes service-oriented modeling activities. These activities are in line with steps or task of business processes (Dow, Ravesteyn, & Versendaal, 2008).

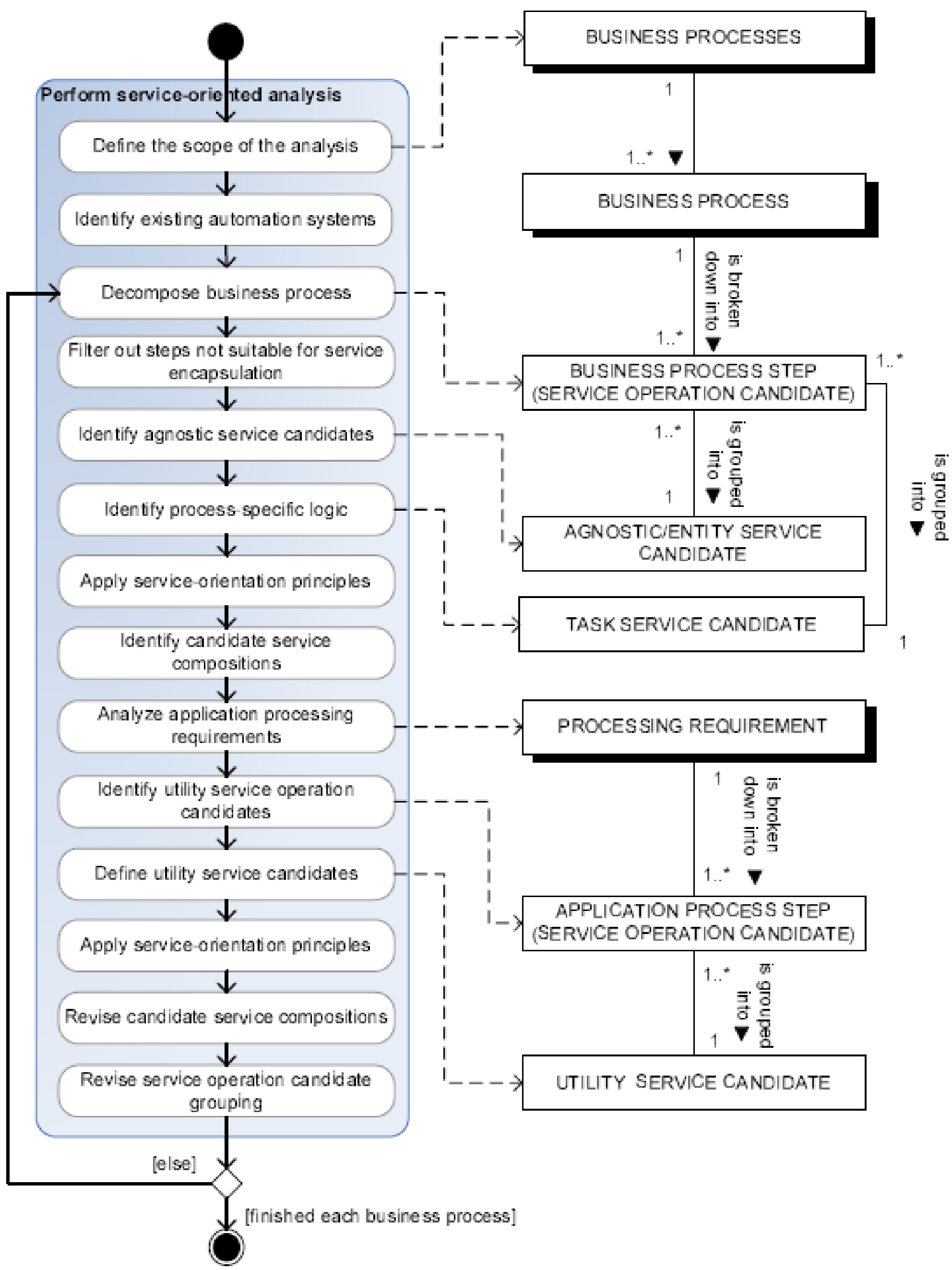


Figure 7: Activities and deliverables of service-oriented analysis

Terlouw, Terlouw, & Slinger (2009) describes that service-oriented analysis first defines what business processes should be automated to conform the current mature and defined business requirements. Each of the business processes is decomposed into granular steps that are afterward grouped into candidate services. A step can be assigned to a *task candidate service* that specifically belongs to the business process. Otherwise, it is unaware of the process, and often linked to a certain business entity, like an invoice or employee. These *agnostic candidate services* are likely to be used in multiple business processes. *Utility* *candidate services* are also created that could encapsulate granular processing steps of application requirements resulting from the analysis of business process steps. The decomposition activity is repeated for every business process, leading to revised task, entity, and utility service candidates.

Schmelzer (2006) points out that the candidate services identified using the top-down approach may not be reusable and recommends using bottom-up approach to identify reusable business logic in existing code and exposing this business logic as reusable fine grained services.

3.1.2. Bottom-up

Terlouw, Terlouw, & Slinger (2009) state that the bottom-up strategy encourages the creation of services as a means of fulfilling application-centric requirements. Marks & Bell (2006) state it is a progressive process of building services or assembling existing technologies to provide business solutions. The bottom-up strategy can tie services to their originating technology environments. This leads to tight coupling and should be avoided, because this will certainly miss out on achieving valuable benefits for the business (Dow, Ravesteyn, & Versendaal, 2008). Figure 8 (Terlouw, Terlouw, & Slinger, 2009) depicts the steps that are performed according to Erl (2005) during a bottom-up approach.

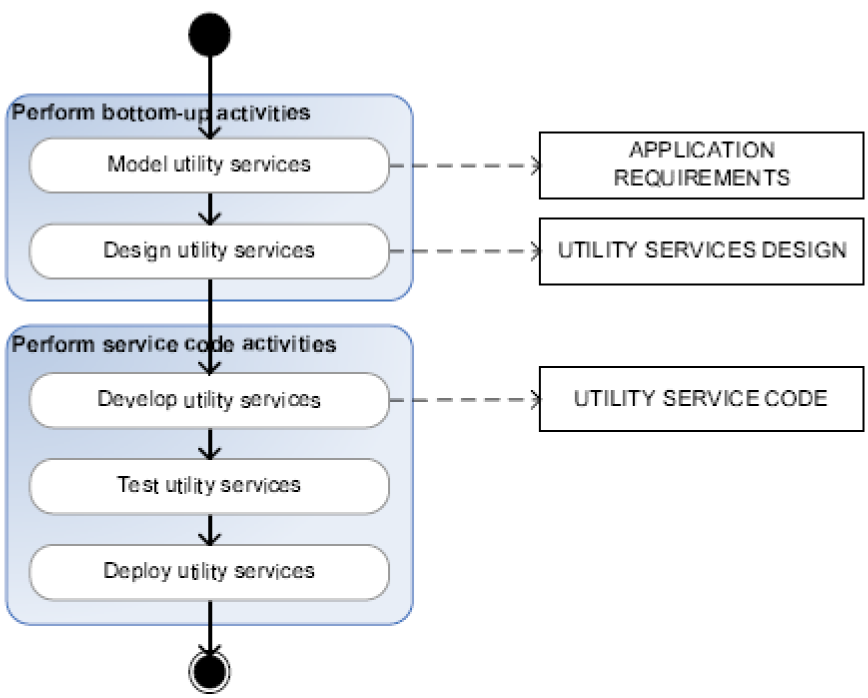


Figure 8: Activities and deliverables of bottom-up strategy

The first bottom-up sub activity consists of modeling the application requirements that can be fulfilled through the use of services. The second sub activity focuses on the design of these utility services, which can come into existence in several ways. Terlouw, Terlouw, & Slinger (2009) state they may be delivered by third party *wrapper services* or *auto-generated proxy services*. Wrapper services can be used for legacy system integration purposes that expose legacy functionality to service requesters. However, custom utility services may also be constructed, which desire a design process where existing design standards are applied. This could lead into more service-oriented utility services, because of their potential reusability.

The viewpoint of reusability is subscribed by Schmelzer (2006). He recommends using a bottom-up approach to identify reusable business logic in existing code and exposing this business logic as reusable fine grained services. Other alternatives for this approach include using Business Process Execution Layer (BPEL) compositions to externalize aggregate services built from low granularity service operations. Another emerging idea involves the use of service components to aggregate fine grained services into a higher-level coarse grained service, preserving the benefit of reuse inherent in low granularity services and at the same time taking advantage of coarse granularity, aggregated services to simplify the interaction dialogue (Feuerlicht & Wijayaweera, 2007).

Terlouw, Terlouw, & Slinger (2009) say that Marks & Bell (2006) also mention a *bottom-up design and analysis approach*. Unlike the approach mentioned by Marks & Bell (2006), the bottom-up delivery strategy of Erl (2005) assumes the business requirements have already been collected. Furthermore, Marks & Bell (2006) refer to application requirements as business requirements, and assume the incorporation of design standards. They also assume business logic to be incorporated, and call the development “design and construct”. The differences in semantic meaning of the activities to be performed make it hard to relate the two bottom-up approaches, says Terlouw, Terlouw, & Slinger (2009).

3.1.3. Meet-in-the-middle

Terlouw, Terlouw, & Slinger (2009) say that the method as proposed by Marks & Bell (2006) advocates a service identification strategy using both a top-down and bottom-up approach, applied in an iterative fashion. The service identification process is conducted in a top-down fashion, with a focus on *candidate business services* as in figure 7. In this case, current physical or technical environments should be disregarded. Instead, attention should be given to the organization’s operating units, and the relationships between those units. Marks & Bell (2006) propose to perform service construction itself in a bottom-up fashion. An iterative process follows to bring both approaches together. The approach of Marks & Bell (2006) resembles the meet-in-the-middle service identification strategy as proposed by Erl (2005). Figure 9 (Terlouw, Terlouw, & Slinger, 2009) depicts the steps of this strategy.

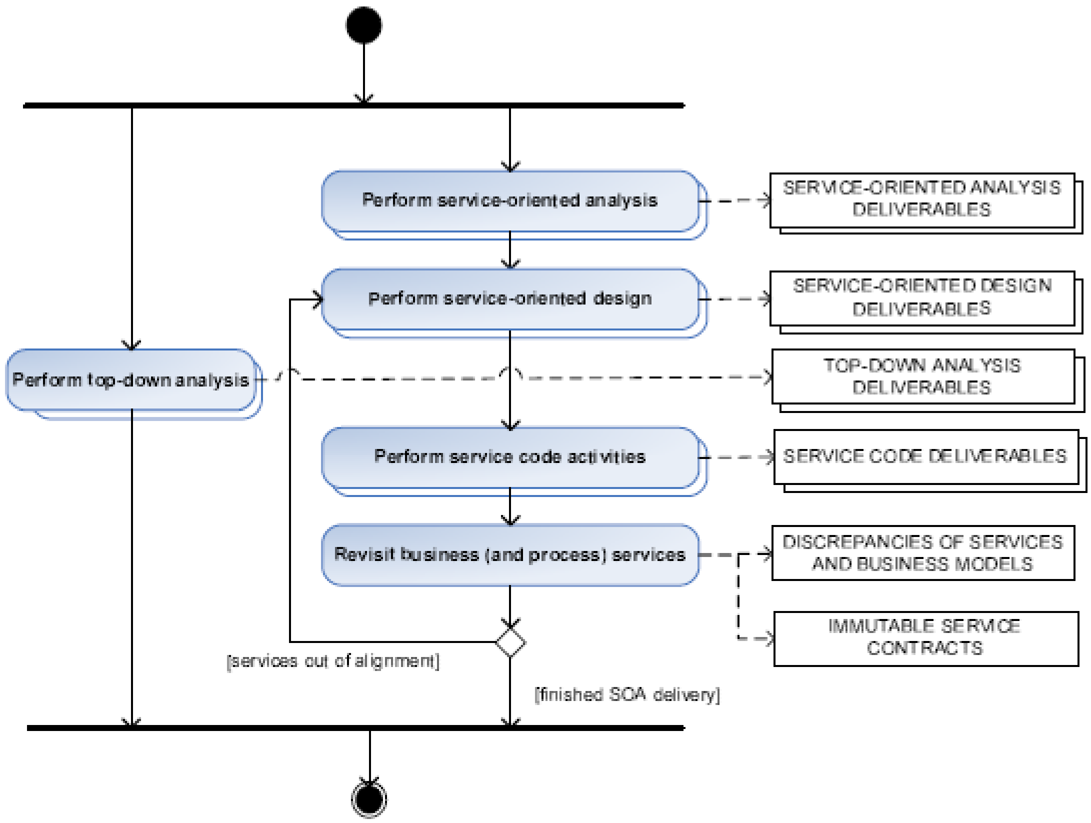


Figure 9: Activities and deliverables of meet-in-the-middle strategy

Meet-in-the-middle starts with a top-down approach. The top-down analysis differs from the top-down service identification strategy in the sense that it is an ongoing effort to further achieve the enterprise-wide analysis goals. When the analysis is sufficient progressed, service-oriented analysis is also initiated using the available business models, and other top-down analysis results. Service-oriented design and service code activities follow like in the top-down strategy. Since an ongoing top-down analysis is executed during these steps, services are subject to revision. Meet-in-the-middle therefore requires an extra activity, in which periodic reviews are performed to compare the design of services against the current state of the business models (Terlouw, Terlouw, & Slinger, 2009). From this point on, an iterative process follows. The service-oriented design and service code activities are repeated for the services out of alignment, followed by another revision.

3.1.4. Middle-out

Figure 10 (Terlouw, Terlouw, & Slinger, 2009) depicts the middle-out strategy according to Rosen, Lublinsky, Smith, & Balcer (2008). They state iterations take place within and between each activity, instead of them being linear. The path on the other hand depends on whether new business capabilities are created or existing capabilities are used. Since they do not mention how, the figure 10 of Terlouw, Terlouw, & Slinger (2009) provides every possible iteration.

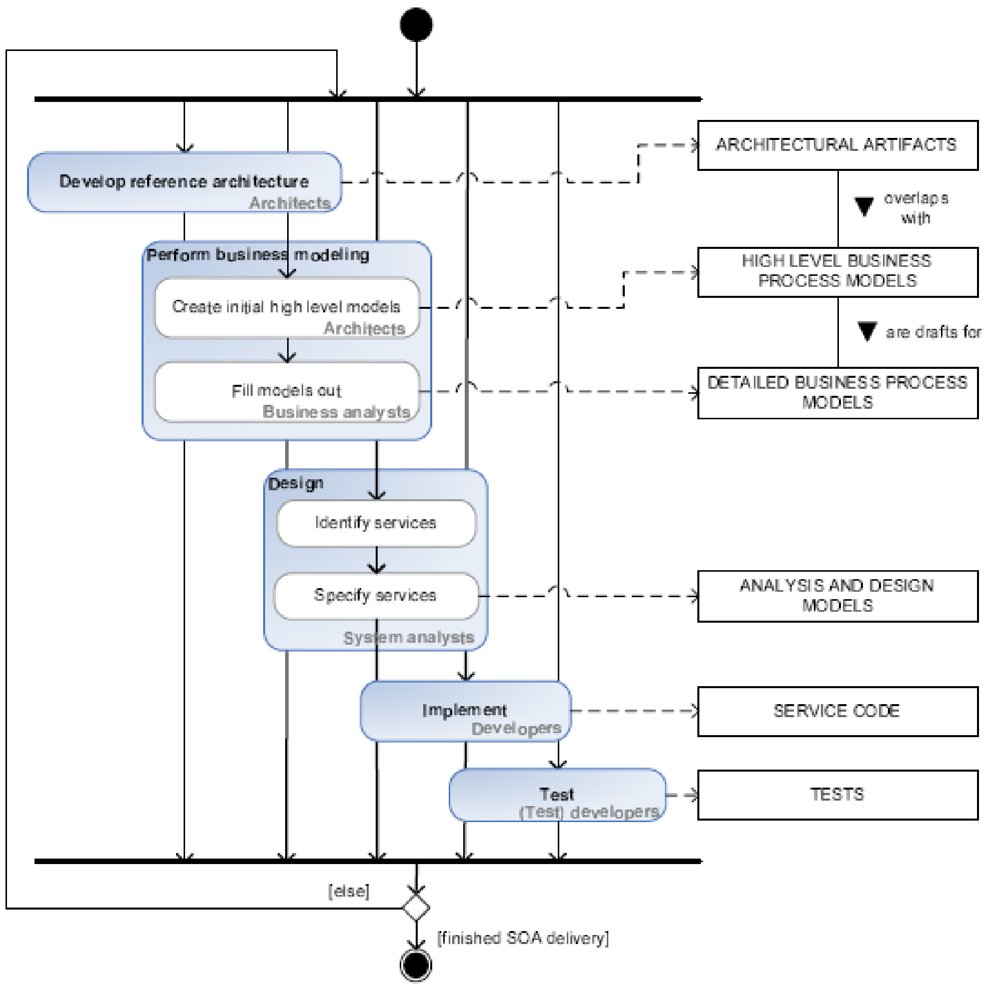


Figure 10: Activities and deliverables of middle-out strategy

The middle-out service identification strategy produces both higher-level business and information architecture and design artifacts, and working and deployed services. Rosen, Lublinsky, Smith, & Balcer (2008) explicitly mention roles and state every activity (i.e. business modeling, design, etc.) focuses on a special goal, like modeling the business context, or enabling services for use in solutions. An activity should not address or be influenced by concerns of other activities. Therefore, middle-out has no sequential activities, unlike the previous strategies (Terlouw, Terlouw, & Slinger, 2009). Performing independent activities is enabled by a stable center, called the *reference architecture*. This extra element provides the context for what the different architectures (business, information, application and technology) describe, what they look like, how they are related to each other, and how they work together to meet overall business goals. The reference architecture provides an overall taxonomy that defines the different types of services together with service groupings. Furthermore, responsibilities are assigned to each activity to help shape the overall service road map. It also provides proven design patterns, different types of applications and services, and technology standards and mappings for service implementations. The reference architecture provides the link between top-down and bottom-up aspects, says Terlouw, Terlouw, & Slinger (2009).

3.1.5. Granularity scheme

The previous paragraphs illustrated several service identification strategies. Services have to be identified before getting the granularity of these services right (McGovern, Sims, Jain, & Little, 2006). A service-oriented architecture must define clear levels of granularity for service implementations. Since services are assumed to provide certain units of functionality, then this means that service granularity must be defined. Without this, it becomes difficult to define how services are composed, and how the various scalability aspects are taken into account. The granularity scheme of Herzum & Sims (2000) (and used with some variation in Combine (2003)) defines three levels of granularity that guides towards a clear level of granularity for services. These levels are shown below.

1. Distributed Service
2. Business Service
3. Application Service

The relationships between these three are shown in figure 11. The most coarse grained is the application service, which consists of a number of business services, which in turn are composed of distributed services (McGovern, Sims, Jain, & Little, 2006).

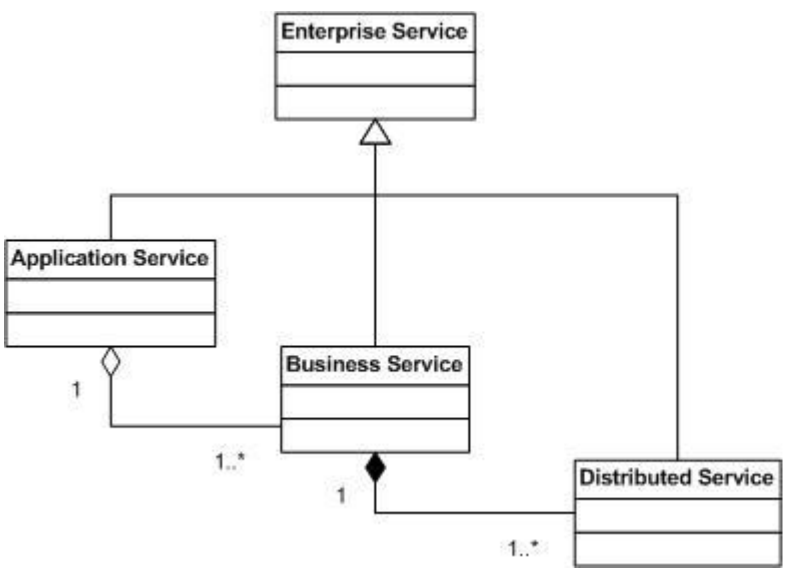


Figure 11: Three levels of service granularity (McGovern, Sims, Jain, & Little, 2006)

Although the mentioned levels of services carries different names in comparison with the service type classification of Steghuis (2006), there are similarities. The hierarchical representation of services is similar to the layered service definition in paragraph 2.2.1. The classification of three service types supports business processes, information needs and infrastructure solutions. When the service type classification of Steghuis (2006) is used, then the distributed services are closely related to implemented infrastructure components and therefore similar to software services. The business services implements the ability to provide information about business concepts. This is in accordance with the information system service classification. Finally an application services provides a total business solution. A complete business solution is based on essential activities that an organization performs (Fowler, 2008) and therefore an application service is similar to a business service in the service type classification of Steghuis (2006).

The enterprise service generalization in figure 11 shows the composition of more fine-grained services as stated by Dow, Ravesteyn, & Versendaal (2008).

The three levels of service granularity are briefly illustrated in the next paragraphs.

3.1.5.1. DISTRIBUTED SERVICES

McGovern, Sims, Jain, & Little (2006) say that a distributed service is typically implemented by platform component models such as Enterprise JavaBeans (EJB), CORBA component, COM component, or various BPM definitions. The distributed service is the smallest level of granularity, and is responsible for the implementation of service components.

The several different ways that a distributed service can be implemented illustrates a really important aspect of the distributed service concept: it is a design concept that maps well to a wide variety of different technologies (McGovern, Sims, Jain, & Little, 2006). Each distributed service provides a unit of functionality, some local, others of very wide scope. In addition, the distributed service concept is sufficiently closely specified to provide for rich design models at the Model Driven Architecture (MDA) Platform-Independent Model (PIM) level, with the concepts preferably presented through a UML profile in an appropriate modeling tool. Finally, a distributed service is large enough to be able to carry some overhead to help provide transparencies for the application developer.

3.1.5.2. BUSINESS SERVICES

McGovern, Sims, Jain, & Little (2006) says that a business service is a subclass of an enterprise service that implements all and only a given business concept (process or entity) in a system. As such, it also provides for the business concept’s distribution throughout the system, across all implementation layers like user interfaces, processing logic, business logic, database components and infrastructure resources.

A business service represent both process and entity business concepts. Thus an application consists of a number of business services that collaborate to deliver application functionality across all implementation layers. In turn, a distributed service within a business service is either a process or an entity distributed service based on what the business service is (McGovern, Sims, Jain, & Little, 2006). It is often useful to divide entity components into two layers: entity and “utility”. A utility business service is one that provides widely-used business services to both entity and process business services. Examples of this include an address book, a number generator, or a currency converter. Since these business services provide information to the business, the business services could be compared with the information system services type of Steghuis (2006).

McGovern, Sims, Jain, & Little (2006) recommend minimizing dependencies in services design so that services can be composed and re-composed (re-used) as freely as possible. A reference service architecture should feature specific interaction patterns that minimize dependencies.

McGovern, Sims, Jain, & Little (2006) list five valuable benefits of the business service concept. Those five benefits are stated below.

1. Link with requirements: it provides a direct link with the business requirements model.
2. Project Management Unit: as a cohesive collection of distributed services, the business service is an ideal unit of project management, and valuable and meaningful metrics can be easily derived. This makes for much more accurate prediction that is often the case. And when application development is mainly done through outsourcing, the business service, or an assembly of business services, is an ideal "module" to outsource.
3. Lifecycle Continuity: as it moves through its lifecycle, a business service is realized by different artifacts at different stages. At analysis time, the artifacts comprising the business service will be UML models, and documents. Design time will add detail to these artifacts, so that code can be produced. The code will consist of some distribution services, plus other artifacts such as Graphical User Interfaces (GUI) and database schema definitions. A service architecture will define what completeness means at the end of each stage, defines the stages, and shows how traceability is provided. Thus a business service encapsulates all development artifacts within it.
4. Domain Knowledge: the business service also acts as a focus for domain knowledge within an application development group. And since the business service extends across the all the implementation layers, there is not only shared domain expertise across the implementation layers, but also within a small team. All of this can be hugely useful in terms of skill growth.
5. Unit of Ownership: re-usable software artifacts are valuable enterprise assets, and must therefore be "owned" by a manager within the IT organization. However, to avoid one manager looking after a lot individual assets, asset ownership must be structured, such that ownership of a large asset implies ownership of embedded assets.

The business service is an ideal unit of asset management, since it "owns" all of the assets that go into the distribution services that make up the business service, and among these there will normally be a number that are re-usable.

3.1.5.3. APPLICATION SERVICES

McGovern, Sims, Jain, & Little (2006) say that business services collaborate to provide a business solution. The assemblies of business services that together provide such a solution effectively make an "application". When this assembly is deliberately produced to be a service in its own right, with one or more service-oriented interfaces, then the assembly is called an application service.

McGovern, Sims, Jain, & Little (2006) defines an application service as an assembly of collaborating business services that provides a defined set of business services.

Figure 12 shows an example of an application service. It consists of a composition of five business services and together providing a simplified Invoice Management service. The business function layers (process-entity-utility) are shown vertically and within each business service the implementation layers are represented as a stack of rectangles. The black solid arrows show intra- and inter-business service invocations. This is an architecture style that could be applied, but other architecture styles could disallow this and prescribes to route all invocations via the top-level business service.

This application service might be called the Invoice Management service. As with most application services, its interface is defined as being the interface of the topmost process business service (Invoice Manager in this example).

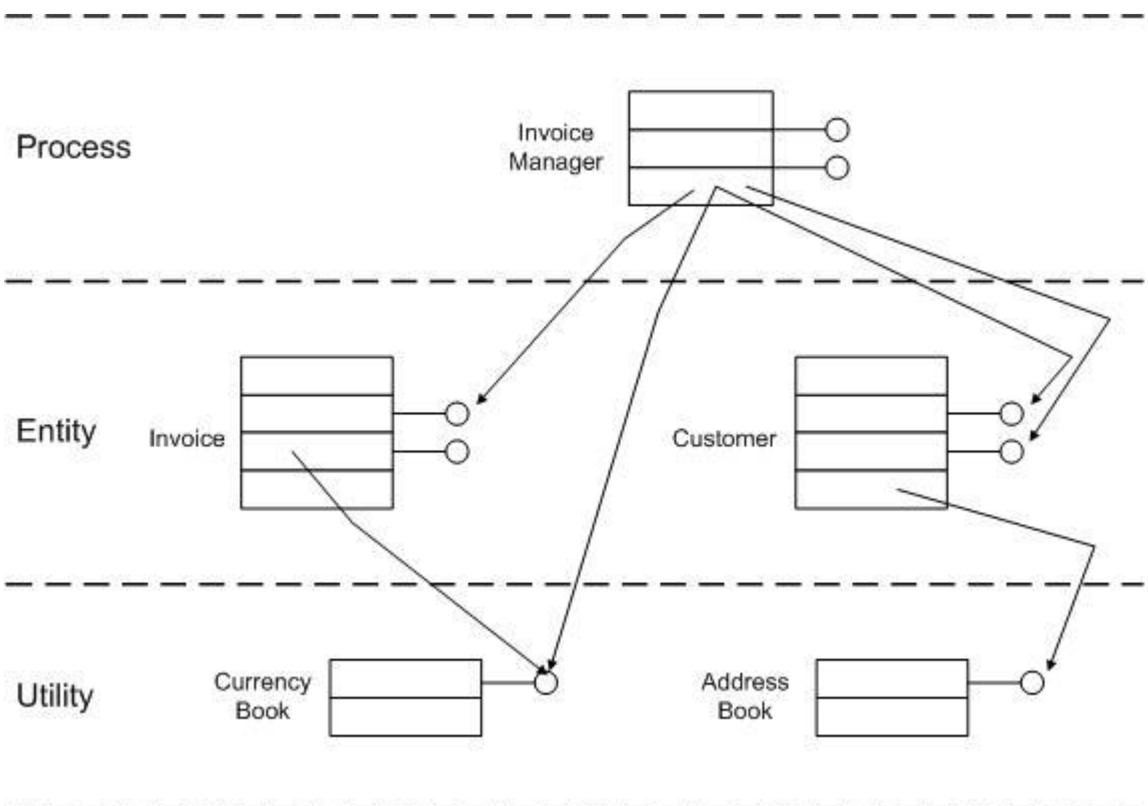


Figure 12: Application service composed of business services (McGovern, Sims, Jain, & Little, 2006)

3.1.6. Summary

This main paragraph discussed several techniques for service identification that are known in the information technology science. The available techniques can be converted into one of the four following strategies.

1. Top-down: is an “analysis” first approach;
2. Bottom-up: is a progressive process of building services or assembling existing technologies to provide business solutions;
3. Meet-in-the-middle: advocates a service identification strategy using both a top-down and bottom-up approach, applied in an iterative fashion;
4. Middle-out: produces both higher-level business and information architecture and design artifacts, and working and deployed services.

Every strategy has its own strengths and weaknesses. From a business point of view the top-down strategy would be preferable to the other strategies. But applying strictly one strategy is usually not practical. Dow, Ravesteyn, & Versendaal (2008) formulate that in practice it will always be a balancing game as to how extensive the top-down modeling should be.

When services are identified, then the right level of granularity needs to be determined. Based on the classification of Herzum & Sims (2000) three levels of granularity are provided in the granularity scheme.

1. Distributed services: are the most fine-grained services and mostly implemented by a specific platform component model;
2. Business services: are coarse-grained services, providing information about a certain business object;
3. Application services: are compositions of business services, providing a total solution for an essential business activity.

One should strive to find an optimum in business services, since that is the ideal unit of asset management and has the largest potential for software re-use (McGovern, Sims, Jain, & Little, 2006). In accordance with the service type classification of Steghuis (2006), the business services of McGovern, Sims, Jain, & Little (2006) are similar with the information system services of Steghuis (2006).

3.2. An overview of techniques to define service granularity

Considering the definition of service granularity of Papazoglou & Van den Heuvel (2006) in paragraph 2.2.2, it is all about getting the right unit of functionality within a service. In paragraph 3.1 several strategies are discussed how to approach the topic of services and service granularity. One of the conclusions was that one should strive to find an optimum in business services (McGovern, Sims, Jain, & Little, 2006). With respect to the research question of this thesis the best way to get the right business services is using a top-down strategy. Utilization of a top-down strategy is an "analysis" first strategy, which starts by analyzing the business and the proposed business benefits. The business services should be optimized in a way that the business goals can be achieved. This main paragraph provides an overview of the different analysis techniques to express service granularity in. Six different models are briefly discussed in the following paragraphs. These techniques are chosen because of their applicability to follow a top-down strategy, their well known foundation in the information technology science, their underlying formal semantic model or their massive utilization in today's IT projects. The models are:

- Design & Engineering Methodology for Organizations (DEMO)
- UML Use Cases
- UML Activity Diagrams
- Petri nets
- Business Process Modelling Notation (BPMN)
- Business Process Execution Language for Web Services (BPEL4WS)

The last paragraph summarizes this main paragraph and concludes the used analysis model that is applied in the case studies.

3.2.1. DEMO

The DEMO modelling language derives abstract models representing only essential activities, is based on formal semantics, and it is service-oriented. This counts for all layers of the organization (De Jong, 2008). The formal semantics of the methodology forces an analyst to focus only on the essential activities of the organization. The resulting DEMO model is representing the organization in its essence. Starting from this model ensures the analyst that essential business transactions must be supported within the different organizational layers. The layered dependencies are expressed in De Jong (2009) where, for example, the informational layer of the organization supports the business layer of the organization. The transactions that are modelled on the different layers of the organization are to be considered as services of that layer (Wiersma, 2009b). Because all transactions are synonym with services, and the informational layer is primary supportive to its business layer means that the informational layer can be modelled top-down only, starting in the business layer, and all layers of the organization that are modelled in DEMO are service-oriented.

When knowing the essence of the organization, the organization is better capable of defining the functional requirements for the informational layer. The informational components that are required to provide the information needed by the business are straightforward identified by the methodology (Dietz, 2003). This means that services on the information layer are primary defined supporting the essential business services. This methodology gives a clear guideline in defining the service model within the information layer of an organization. The determinants of service granularity can be tested against the transactions in the DEMO model.

3.2.2. UML Use Cases

There is a lot of literature on use cases, like (Hoogendoorn, 2003), (Dietz, 2003) and (Fowler, 2008). In Dietz (2003) is said that a use case is a construct for the definition of the behaviour of a system without revealing its internal structure (Cockburn, 1997). In Jacobson, Christerson, Jonsson, & Övergaard (1992) it is defined more specifically as a description of the complete course of events initiated by an actor and of the interaction between the actor and the (future) system. Dietz (2003) says that use cases are mostly applied in engineering methods that are based on the UML. One of the advantages of use cases is that it is very strong in describing scenarios. It describes how users interact with a system (Fowler, 2008). Each scenario is a kind of a process that could inherit other scenarios. For each scenario different roles can be required to execute it. Fowler (2008) refers to three levels of use cases. Bases on the use case scheme of (Cockburn, 1997) the levels kite-level, sea-level and fish-level are recognized. These levels represent respectively business use cases, a system use case describing a discrete interaction between a primary actor and the system, and a detailed system description in pseudo code. When a use case is executed, the result is noticeable for the involved actors (Hoogendoorn, 2003). Dietz (2003) says that the strong point of this method is that once the use cases are identified, the development of the application software goes smoothly. The weak point however is the identification of the use cases itself. Defining use cases is based on interviews with business experts. The descriptive use cases are probably not focussed on the essence of the organization, but they are too detailed or not detailed enough. The interviewee has indirect a large role in the created use cases. The combination of scenario inheritance, multi-role scenarios, and ambiguous levels of details are complex to measure with the patterns for service granularity of Steghuis (2006).

3.2.3. UML Activity Diagrams

Activity diagrams are a technique to describe procedural logic, business process, and work flow (Fowler, 2008). The activity diagram allows whoever is doing the process to choose the order in which to do things. In other words, the diagram merely states the essential sequencing rules that have to be followed. This is important for business modeling because processes often occur in parallel. The nodes on an activity diagram are called actions, not activities. Strictly, an activity refers to a sequence of actions, so the diagram shows an activity that is made up of actions (Fowler, 2008). Actions can be decomposed into sub activities, which can be implemented in methods on classes. This is already fine grained and could be generated into software component. Other aspects of activity diagrams lie in the area of process modeling. This makes activity diagrams an important technique to represent behavioral logic. The orchestration of actions formulates the business execution. This means that activity diagrams are to be used for orchestration of services, but it gives no direction in how those actions need to be created. The unit of functionality an action, or service, already must have been defined and solved before business process orchestration.

3.2.4. Petri nets

Petri nets can be used to model dynamic systems with a static structure (Weske, 2007). Petri nets are, like DEMO, abstract models and based on formal semantics (Aalst & Hee, 2008), (Weske, 2007). The advantages of formal semantics are its clear and unambiguous description and the ability to reason about processes (Aalst & Hee, 2008). In this way, reasoning about processes shows certain effects of the process execution. In essence, classical Petri nets consist of places, transition and directed arcs connecting places and transitions (Weske, 2007). Petri nets provide a stepwise process execution (Aalst & Hee, 2008). Transitions have input and output places. The input places of a transition are the places at the sources of its incoming arcs. Accordingly, a transition's output places are located at the end of its outgoing arcs. Transitions are the active components of Petri nets. By triggering a transition the process changes from the current state into the new state (Aalst & Hee, 2008). Transitions are similar to events or services. By triggering an event the system state changes. This can be compared to the function of services. Triggering a service changes the current state of a process. The dynamics of the Petri net system is modeled by tokens that reside on places. The tokens may change their positions according to firing rules. The current distribution of the tokens among the places determines the state of the Petri net and, thus, of the system modeled by it (Weske, 2007). Like activity diagrams, Petri nets are to be used for process execution of tokens via transitions. A Petri net in itself does not prescribe and gives no direction in defining the unit of functionality a transition should carry. The unit of functionality a transition already must have been defined and solved before business process definition.

3.2.5. BPMN

The intent of Business Process Modeling Notation (BPMN) in business process modeling is very similar to the intent of UML for object-oriented design and analysis (Weske, 2007). BPMN is a graphical representation for specifying business processes in a workflow. The set of ancestors of BPMN include not only graph-based and Petri net based process modeling languages, but also UML activity diagrams and event-driven process chains.

Weske (2007) says that while these modeling languages focus on different levels of abstraction, ranging from a business level to a more technical level, the BPMN aims at supporting the complete range of abstraction levels, including business levels and software technology levels. The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analyst that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes (Weske, 2007). In BPMN activities represent units of work performed during business processes. An activity can be atomic or non-atomic (compound). Weske (2007) states that atomic means that, for modeling, the internal structure of the activity is not relevant. Atomic activities are also called tasks. BPMN distinguishes several types of tasks. There are service tasks, which is implemented by a piece of software, either using a web service interface or an application programming interface to a software system. There are user tasks representing traditional workflow tasks that involve user interaction. Another type of task is a script task, which is a task that uses some scripting language expression in order to be performed. Yet another type is the manual task, which is performed without the support of software systems. Finally, there are reference tasks supported. Reference tasks provide a means to reuse tasks in different business processes. A reference task can link to a specific task that has been defined beforehand. Despite that BPMN provides a modeling notation for tasks or services, it gives no direction about the unit of functionality a service must contain. Finding the right service granularity is not supported by BPMN and therefore should be defined before modeling business processes.

3.2.6. BPEL4WS

BPEL4WS is a formal language specification used for composition, orchestration, and coordination of web services. It provides a rich vocabulary for expressing the behavior of business processes (Juric, Mathew, & Poornachandra, 2006). BPEL4WS is the most appropriate technology for SOA realization. Services, as a unit of functionality, are accessed through their interface. Each interface defines a set of operations. In order to define business services, the focus lies on correct granulation of operations. SOA services are best modeled with coarse granulation (Juric, Mathew, & Poornachandra, 2006). This concludes that BPEL4WS subscribes the statement to get granularity of services right. It even subscribes the viewpoint that coarse granulation for web services fits SOA at best, but the language itself does not provide a clear guideline of how to get those business services right.

3.2.7. Summary and conclusion

This main paragraph provided an overview of six modeling techniques that structures process modeling and addresses the topic of service granularity. Although about all models follow the top-down strategy, only one is directly linked to the essential business activities. The formal semantics of DEMO guides business analysts to focus on the essential activities only, which also is directive in identification of supportive services. Requirements elicitation is the weakest link in all other methods (Dietz, 2003). To define activities, following a top-down approach, (Dietz, 2003) says that the following requirements must be supported by the model:

- It should make a clear and well-founded distinction between the *essential* business actions and informational actions. The latter are exactly things that have to be reconsidered when developing an information system. For example, requesting a supplier to deliver articles is essential, but computing the amount of articles is informational.
- It should have the right granularity or level of detail. “Right” means in this respect: finding the actions that are *atomic* from the business point of view. They may be composite only in their implementations. For example, the request to a supplier is atomic from the business point of view, but to perform a request by postal mail, a number of non-essential actions have to be taken like mailing the order form, transporting it and deliver it to the supplier.
- It should be *comprehensive* or *complete*, i.e. it should contain everything that is necessary and it should not contain anything that is irrelevant. This requirement is possibly the most hard to satisfy since it is common practice in most organizations to perform several kinds of coordination acts tacitly, according to the rule “no news is good news”.

Besides the above mentioned requirements the different techniques are assesed on the following criteria as well. The assessment is based on theoretical desk research, which has been summarized per technique in this paragraph.

- Top-down approach.
- Formal semantics.
- Coherent, this means that the technique constitutes a logical and truly integral whole.
- Service-oriented.
- Objective, this means that the technique is based on real facts and not influenced by personal beliefs or feelings.
- Granularity, this means that the technique is directive in one or more service granularity determinants.

Table 4 gives an overview of the assessed criteria per technique. The colored cells represents meeting the criterium by a technique.

Table 4: *Overview of assessed criteria per modeling technique*

	Techniques		Activity			
Criteria	DEMO	Use Cases	Diagrams	Petri nets	BPMN	BPEL4WS
Top-down						
Formal semantics						
Essential						
Coherent						
Comprehensive						
Service-oriented						
Objective						
Granularity						

The assessment of criteria points that DEMO is the right choice that most likely meets the objectives of this research at best. DEMO enables defining a service model in a structured way that is directly linked to the essence of the organization without mentioning anything about process orchestration or implementation on the informational level.

3.3. Getting granularity right

The preceding paragraphs addressed some topics in the available literature that is related to service granularity. The literature study started with providing an overview of the main business drivers for SOA. The various definitions of SOA, that are used in this thesis, all points at the advantages of enabling flexibility in business processes and the possibility to reuse software services within a heterogeneous environment. These advantages must not become a goal on its own, but always be supportive to the business strategy.

The link between SOA and services created the foundation for paragraph 2.2. That paragraph illustrated the concepts of services and the concepts of granularity and how these two terms are related to each other. To put the concepts in the right perspective this research project adopts the definitions of Lankhorst (2005) and Terlouw & Albani (2010), saying respectively “A service is a unit of functionality that some entity makes available to its environment, and which has some value for certain entities in the environment” and “A service is a universal pattern of coordination and production acts, performed by the executor of a transaction for the benefit of its initiator, in the order as stated in the standard pattern of a transaction”. The combination of both definitions addresses the interaction of a service invocation as well as the intention of the service.

The research of Steghuis (2006) pointed at several determinants of service granularity per service type. The determinants are listed in table 3 on page 31. This classification of determinants for service granularity is of great importance in this thesis. It is an anchor to test optimum service granularity with. All the determinants play a central role in deciding about service granularity. When this thesis tries to find an optimum this means that the determinants must be evaluated as well.

Several strategies to determine services and service granularity can be applied. Paragraph 3.1 addressed four strategies; top-down, bottom-up, meet-in-the-middle, and middle-out. Although every strategy has its own strengths and weaknesses, from a business point of view the top-down strategy would be preferable to the other strategies (Terlouw, Terlouw, & Slinger, 2009). The top-down strategy also is applied in this thesis’ case studies. To support the analysis first top-down strategy, six different models are evaluated in paragraph 3.2. Considering the evaluation of the six modeling techniques, the DEMO technique is the best fit. The formal semantics of DEMO guides business analysts to focus on the essential activities only, which also is directive in identification of supportive services.

The topics addressed in the literature study contribute to the body of knowledge of service granularity. Already several times has been said that granularity of services must be optimized or must be right, but until now, this term has not been elaborated on. What does optimum mean?

The Cambridge University Press (2010) translates optimum in “most likely to bring success or advantage”. This means that it is not a certainty that success or advantage is achieved. This relative verb also has a subjective nature. What is perceived as success or advantage is different for almost everybody. To find an optimum in service granularity also refers to the best fit of services for a certain organization following a specific strategy and aiming at certain organizational goals. Dietz (2003) also refers to link optimum granularity with the business point of view. Weske (2007) states that the choice of a suitable service granularity depends on the particular usage scenario and on the properties of the application systems to integrate and the composite applications to develop. Dow, Ravesteyn, & Versendaal (2008) say that coarse granularity of services is relative to the level of problem being addressed and defining the optimal level is not as simple as counting the number of interfaces that a service has.

Optimum in this research project means achieving an informational service model that is recognizable for the business, is directive for IT specialists in systems development, supports flexible business process execution and effectively shows service reusability. Following the DEMO top-down approach results into essential business transactions. Grounded in the organization theorem, the essential business transactions are supported by the informational function of the organization, meaning the infological layer. The information services, within this layer, could be implemented with IT, meaning that a software system could support the information requests of the business. When all the essential business activities are known and well defined, and the information requirements are clear, then services can be deduced from the total information requests. The resulting services can be structured in a service model, which, if expressed in the DEMO infological model, is directly linked to the essential business transactions. The other way around, a complete service model is directive for IT specialists, because it is totally transparent what kind of services must be created to support the business requirements. From this point of view, an optimized services model is able to support all essential business transactions in a recognizable way (which is achieved by linking the informational services to the business transactions) and is able to support the IT specialists by being directive in the necessary services that must be created, when services are assigned to be implemented in software systems.

Before the case studies are introduced (that are underlying of this research project) a short introduction of the DEMO methodology is provided in the next chapter. The introduction in the methodology is required to get a better understanding of the case studies results and conclusions.

4. INTRODUCTION TO THE ONTOLOGY OF THE INFORMATIONAL ORGANIZATION

This chapter introduces the ontology of the informational organization, in preparation to answer sub question 6 “*What is the DEMO informational construction model?*”. Answering this sub question contributes to the understanding of the central role DEMO plays in this thesis and provides the underlying theory for the created models in the case studies. The informational organization is perceived as an intellect system that processes data into information that can be interpreted by its semantic meaning by human beings. The construction and implementation models of the informational organization are part of the software development process (De Jong & Dietz, 2010) and support the basic foundation for this research in finding optimum service granularity. In paragraph 4.2 the information organization is addressed in context with the layered enterprise, to get a clear understanding of the role of the informational organization. This role is further explained by a summary of the PSI-theory in paragraph 4.3 on which DEMO is founded. The four axioms and the organization theorem explain the layered integration of homogeneous systems. After the theory of DEMO the aspects models that are part of the methodology are illustrated. These aspect models represent the construction of organizations and are discussed in paragraph 4.4. Paragraph 4.5 briefly summarizes this chapter.

4.1. Why DEMO?

According to paragraph 3.2.7, the DEMO modelling language is chosen because it derives abstract models representing only essential activities, it is based on formal semantics, and it is service-oriented. This counts for all layers of the organization (De Jong, 2008). The formal semantics of the methodology forces an analyst to focus only on the essential activities of the organization. The resulting DEMO model is representing the organization in its essence. Starting from this model ensures the analyst that essential business transactions must be supported within the different organizational layers. The layered dependencies are expressed in De Jong (2009) where, for example, the informational layer of the organization supports the business layer of the organization. The transactions that are modelled on the different layers of the organization are to be considered as services of that layer (Wiersma, 2009b). Because all transactions are synonym with services, and the informational layer is primary supportive to its business layer this means that the informational layer can be modelled top-down only, starting in the business layer, and all layers of the organization that are modelled in DEMO are service-oriented.

When knowing the essence of the organization, the organization is better capable of defining the functional requirements for the informational layer. This means that services on the information layer are primarily defined supporting the essential business services. This methodology gives a clear guideline in defining the service model within the information layer of an organization. The determinants and the four patterns of service granularity can be tested against the transactions in the DEMO model.

4.2. Information organization

This paragraph addresses the three levels of enterprises and how these levels are related to each other. The paragraph explains that the three levels apply to all enterprises and how to differentiate between the three basic levels.

Activities in enterprises can be categorized into three basic levels (Dietz, 2006a). First there are activities related to the primary function of an enterprise. These activities concern the production of the material or immaterial goods and represent the core business of the enterprise. The level where these essential activities are performed is called the ontological layer. To carry out the activities at the ontological level there are activities required concerning collecting and providing information (Hoogervorst, 2009). This information can be of all kinds, but is always supportive to the activities at the ontological level. For example if someone wants to buy something he checks his bank credit first. This information supports him in deciding to buy or not. Dealing with addressing and handling the content of information is called the infological layer. The third level is about activities that support the infological layer, and concerns the form of information. The form of information can be expressed in data. In other words the transmission, transformation, or storage of data are datalogical activities that supports the infological layer.

It is plausible that all layers are about realizing some requested results (Hoogervorst, 2009). In order to realize a result on the ontological level, some information might be required that in turn must be gathered from a database. This means that realizing one result might imply that multiple results must be realized on the supportive layers. Although different actors request for a result per layer this does not mean necessarily that these actors are different human beings. It is still possible that a human being must produce a fact on the ontological layer and also request information from a database. For the sake of clear distinction of responsibility areas it is important to distinguish the various actor roles (Hoogervorst, 2009).

The three basic levels of an enterprise apply to all organizations that produce material or immaterial goods only. In case an organization produces information, it means that producing information is its core business and therefore ontological activities. During the production of information, which is an ontological activity for that organization, some other information might be required. For example an asset manager produces a figure that represents the net asset value of the portfolio daily. To produce this figure the asset manager needs to have information about the stock quotes, the currency quotes, the transaction history, the legislation etc.. The same holds for an organization whose primary activity concerns the production of documents. Figure 13 shows the three activity levels in perspective (Hoogervorst, 2009). The figure shows clearly that every activity level itself concerns the ontological layer, the infological layer, and the datalogical layer.

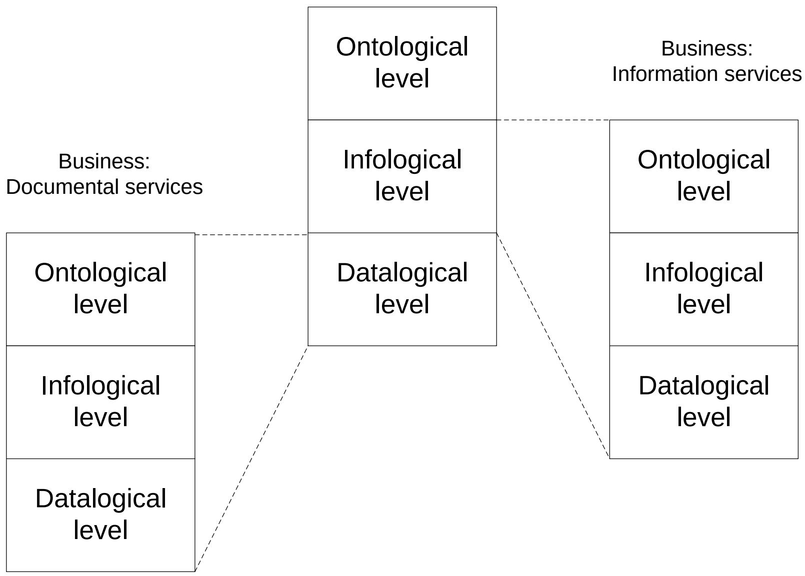


Figure 13: Three activity levels in perspective (Hoogervorst, 2009)

This paragraph discussed the three activity levels of enterprises and showed that every activity level itself concerns the three activity levels. This recursive behavior shows that the foundation of the DEMO theory applies for every layer. The next paragraph zooms into this foundation and addresses the four axioms DEMO is build upon.

4.3. PSI-theory

The underlying theory of DEMO is the PSI-theory (Dietz, 2006a), (Dietz, 2006b). The theory regards organizations as social systems and sees IT systems as support for social actors in performing communication-related activities and production-related activities (Terlouw & Albani, 2010). The theory supports the ontological view of organizations that abstracts from all implementation and realization issues (De Jong & Dietz, 2010). The PSI-theory originates in the scientific fields of Language Philosophy, in particular the Language Action Perspective (LAP) (Flores & Ludlow, 1980), (Goldkuhl & Lyytinen, 1982), and the Systematic Ontology (Bunge, 1979). It focuses on the use of language to achieve agreement and mutual understanding (Weigand, 2003). By applying the PSI-theory one can disentangle the essential knowledge of the construction and the operation of the organization of an enterprise (Terlouw & Albani, 2010). In the underlying theory of DEMO (Dietz, 2006a), an organization is conceived as the layered nesting of three aspect organizations: the B-organization (from Business), the I-organization (from Intellect), and the D-organization (from Document) (De Jong & Dietz, 2010). Every aspect organization consists of four axioms and one theorem. These axioms are the operation axiom, the transaction axiom, the composition axiom, and the distinction axiom, and the organization theorem (Dietz, 2006a). The remainder of this paragraph elaborates on the axioms and the organization theorem.

4.3.1. The operation axiom

This paragraph discusses the first axiom on which DEMO is founded, the operation axiom. The operation axiom is about the different acts actor roles perform. Every act has a definite result that contributes to bringing about the function of the organization or complies with commitments towards each other.

The operation axiom states that the operation of an enterprise is constituted by the activities of actor roles, which are elementary chunks of authority and responsibility, fulfilled by subjects (Dietz, 2006a). The operation axiom is summarized in:

"The actor roles perform two kinds of acts: production acts and coordination acts. These acts have definite results: production facts and coordination facts respectively. By performing production acts, subjects contribute to bringing about the function of the organization. This means that subjects contribute to bringing about the goods and/or services that are delivered to the environment of the enterprise. By performing coordination acts, actors enter into and comply with commitments towards each other regarding the performance of production acts."

The operation axiom distinguishes between production acts and coordination acts. The acts have effect in the production world and in the coordination world respectively. The mentioned worlds are part of a system wherein the acts are effective. A system may be said to have a definite composition, a definite environment, and a definite structure (Bunge, 1979). Regarding the ontological system notion is also said that a system has a definite production (Dietz, 2006a). The state of the production world reflects the effects of the production acts that are performed by the elements of the system. The state of the coordination world reflects the influences among the elements by means of their structural relationships.

A coordination act is defined by its proposition and its intention (Dietz, 2006a). The proposition consists of the production fact. An example of a production fact is: "Invoice #201 has been paid on February 27th, 2010". In the proposition act the performer proclaims the fact and the associated time the intention is about. The fact is the production fact and the time attribute refers to the time at which the fact is actually the case. The intention represents the purpose of the performer. Examples of intentions are: request, promise, or decline. The effect of performing a coordination act is that both the performer and the addressee of the act get involved in a commitment regarding the referred production act (De Jong & Dietz, 2010).

The standard notation of a coordination act is graphically formulated in figure 14 by Dietz (2006a). It concerns the request by the subject A to the subject B to pay the invoice on time.

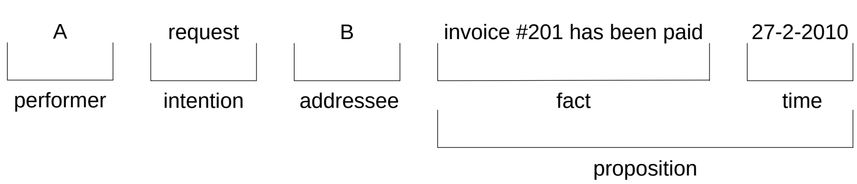


Figure 14: Standard notation of a coordination act (Dietz, 2006a)

Coordination acts are always, either directly or indirectly, about production acts (Dietz, 2006a). A production act is either material or immaterial. Examples of material acts are manufacturing and storage of goods and transportation acts, or computing and deducing information acts. Examples of immaterial acts are the judgment by a court to condemn someone, granting an insurance claim, or selling goods.

Figure 15 exhibits the operation axiom graphically. The symbols are important in the way of modeling DEMO and are defined in Dietz (2006a). The following description is given by Dietz (2006a) to explain the symbols used in the notation. “The symbol for coordination is the disc, the symbol for actor roles is the box, and the symbol for the production is the diamond. The plain arrow from the actors box to the coordination world disc expresses that actors perform coordination acts. The dashed arrow from the coordination world disc to the actors box expresses that actors take account of the state of the coordination world when being active. Likewise, the plain arrow from the actors box to the production world diamond expresses that actors perform production acts, and the dashed arrow from the production world diamond to the actors box expresses that actors take account of the state of the production world when active.”

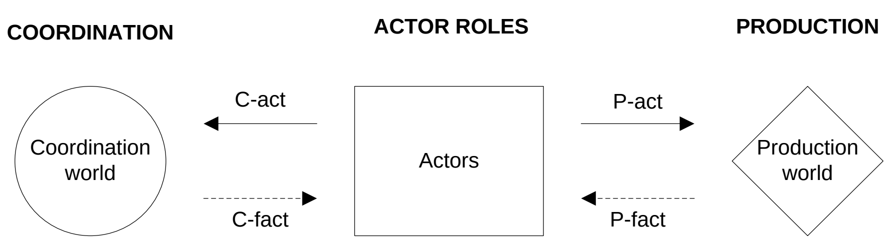


Figure 15: Graphical representation of the operation axiom (Dietz, 2006a)

This paragraph elaborated on the operation axiom. The operation axiom distinguishes between production acts and coordination acts. The acts have effect in the production world and in the coordination world respectively. Coordination acts are always, either directly or indirectly, about production acts. A production act is either material or immaterial.

4.3.2. The transaction axiom

This paragraph introduces the second axiom on which DEMO is founded, the transaction axiom. The transaction axiom is about the universal pattern of coordination acts and production acts. This universal pattern will be explained including the transitions between states of the pattern.

The transaction axiom further looks into the coordination acts. It appears that coordination acts and production acts occur as steps in a generic coordination pattern (Dietz, 2006c). This pattern is called a transaction. Transactions always involve two actor roles and are aimed at achieving a particular result. The transaction axiom states (Dietz, 2006a):

“Coordination acts and production acts always occur in particular patterns. These patterns are paths through one universal pattern, called transaction. The result of carrying through a transaction is the creation of a production fact.”

Four basic coordination activities can be identified: the *request* (to realize or produce something), the *promise* (to honor the request), the *statement* (that the requested is produced), and the *acceptance* (of the produced item) (Dietz, 2006a). A transaction pattern is thus a series of activities performed by two actor roles (Hoogervorst, 2009).

The basic transaction pattern in Figure 16 reads as follows. A certain actor (initiator) has a need for a specific result. This result might concern obtaining a material item for example, or getting approval for something. In achieving the desired result the initiator formulates a request. The other partaking actor (executer) receives a requested result. The executer decides if the desired result can be produced. If so, the executer promises to deliver the result, otherwise the executor declines the request. If the request for the desired result has been promised, then the executer produces the result and states that the result has been produced. The initiator actor observes the stated result and accepts the result. Only if the agreement is reached about the result will the production fact come into existence. The formal way of modeling the transaction pattern is shown in the right part of the figure.

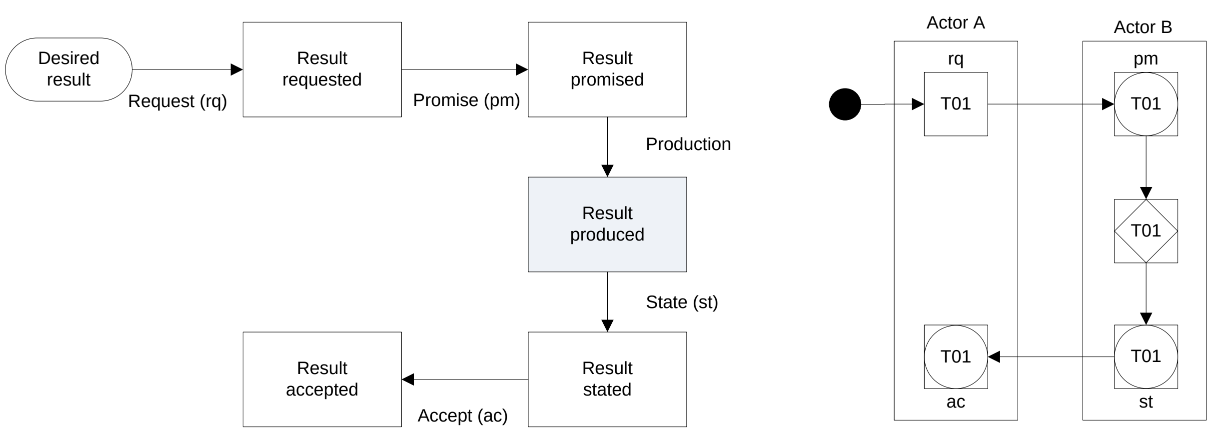


Figure 16: Basic Transction Pattern (Hoogervorst, 2009)

An example of the basic transaction pattern looks as follows:

1. Person A *requests* person B to *develop a software program*
2. Person B *promises* person A to *develop a software program*
3. <actual delivery of the developed software program>
4. Person B *states* to person A that the *software program has been developed*
5. Person A *accepts* from person B that the *software program has been developed* (the software program conforms the expectations)

The transaction pattern shows two conversation phases and one execution phase. The first conversation phase is called the *order phase* and is defined by the “request” and the “promise” coordination activities. The *execution phase* is where the material or immaterial production takes place.

The second conversation phase is called the *result phase* and is defined by the “state” and “accept” coordination activities (Hoogervorst, 2009).

The transaction pattern in Figure 16 presumes the normal execution of a transaction. However in reality a request can be denied or the produced can be refused. Both the initiator and the executer may dissent in two of the states; (i) the requested state and (ii) the stated state (Terlouw & Albani, 2010). In the first case the executor may, instead of promising, respond to a request by declining it. The initiator who made the initial request can refrain from further action and quit or maintain the request. Ultimately, either the promise or the quit state will end the order phase. In the second case the initiator may, instead of accepting, respond to a statement by rejecting it. The executer might agree with the reject and stop or the executer tries to gain acceptance. In the end, the result phase ends with either the acceptance or the stop state. By allowing these acts, a transaction can end up in a discussion state. Dietz (2006a) says that in this situation the actors must sit together, discuss the situation at hand, and negotiate about how to go further. When the basic transaction pattern is extended with these two dissent patterns, the standard transaction pattern is perceived. The standard transaction pattern is shown in figure 17.

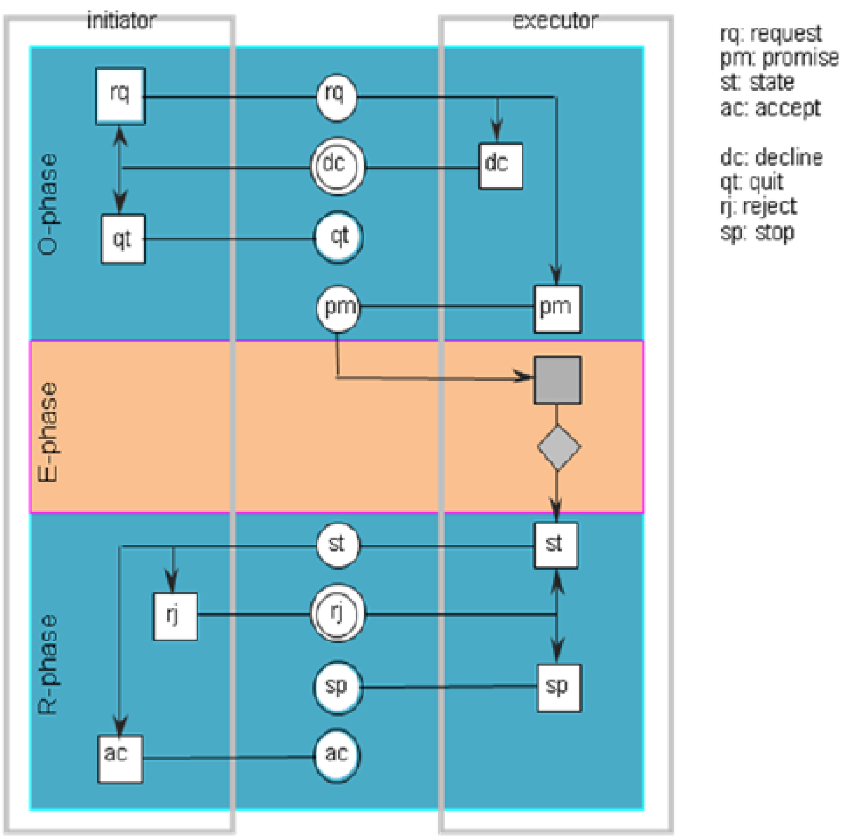


Figure 17: Standard Transaction Pattern (Dietz, 2006a)

The second extension to the basic transaction pattern consist of adding four cancellation patterns, one for each main transaction step (request, promise, state, accept) (Dietz, 2006c). Cancellations can occur when an initiator or an executor of a transaction wants to revoke an act. This means that a cancellation can only be performed if the coordination act to be cancelled exists (Dietz, 2006a). A cancellation coordination act will always be evaluated by the other partaking actor. The actor who receives the cancellation can either respond by allowing or refusing the cancellation. When the cancellation is allowed, the transaction will end in one of the end states or discussion states of the standard transaction pattern. In case the cancellation is refused, then the cancellation terminates. This means that the original coordination act remains the case.

When the cancellation patterns are added to the standard transaction pattern, the complete transaction pattern is created. The complete transaction pattern is considered to be a *socionomic law*: every transaction in every kind of organization is some path through this pattern (Dietz, 2006a).

This paragraph discussed the universal pattern of coordination acts and production acts that is called the transaction axiom. The main states of the transaction axiom were mentioned: request, promise, state, and accept. If this pattern is extended with discussion states, then the standard transaction pattern obtained. Besides the standard transaction pattern, four cancellation patterns exist. The combination of both is considered to be a socionomic law.

4.3.3. The composition axiom

This paragraph introduces the composition axiom. The composition axiom is the third axiom on which DEMO is founded. The composition axiom addresses the ability to nest transactions in a structure.

The composition axiom states that every transaction is enclosed in some other transaction, or is a customer transaction of the organization under consideration, or is a self-activation transaction (Dietz, 2006a). The description of the composition axiom states the nesting of transactions. This means that a transaction can start another transaction which can start another transaction itself etc.. The end-level of process detailing is reached when production activities of a transaction have to do with “atomic” tasks that make further detailing impossible or unfruitful (Hoogervorst, 2009). The nesting of transactions creates a composition of production facts that are produced. The nested production facts then are semi manufactured products that could be part of a larger production fact. For example the manufacturing of a car consist of the manufacturing of the dashboard, the steering wheel, the seats, the bodywork, the engine, etc.. Piece by piece, all production facts are successful results of transactions, but as a whole they are the successful result of the transaction to manufacture a car.

When transactions are nested, the process should be aware of the dependencies between the active transactions. In case of the car example, the production fact to manufacture the car can only be executed when all dependent transactions are finished successfully. This dependency introduces wait moments in active transactions. The dependencies between transactions are basically synchronous process executions. The way of modeling a wait dependency in the transaction pattern is a dashed arrow pointing at the act that waits for the successful end state of the origination act the dashed arrow starts from. The structure of enclosing a transaction is shown in figure 18.

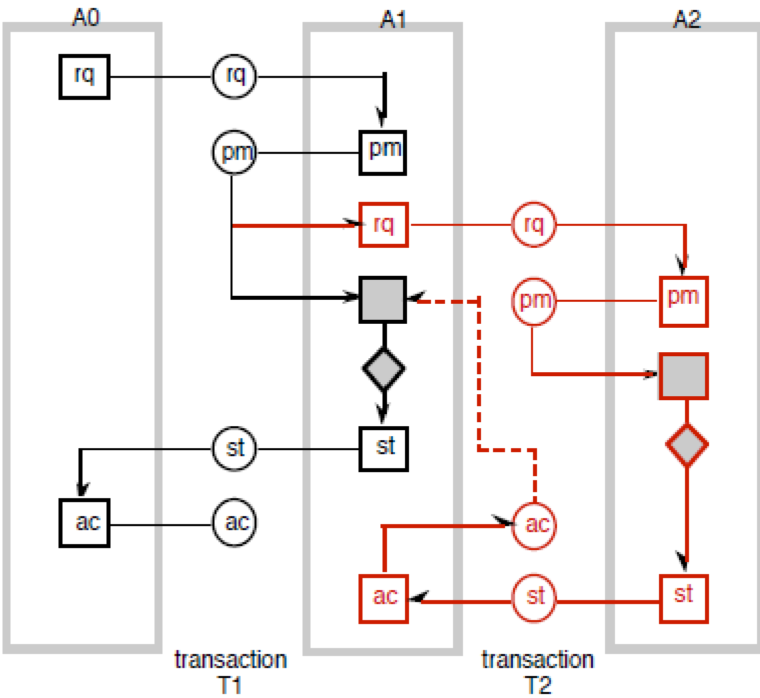


Figure 18: The structure of enclosing a transaction (Dietz, 2006a)

If a transaction starts from an initiator who is not part of the system, then the transaction is a customer transaction. This means that the transactions within the system of the customer are not part of the organization under consideration. In this case the transaction is triggered by external activation.

Another way to initiate a transaction is called self-activation. The self-activation structure is the generic solution for periodic activities, as for all control activities (Dietz, 2006a). Self-activation is initiated when two acts are performed from the state requested. One is the promise act, performed by the executor. The other act is the request for a next transaction, performed by the initiator. The way of modeling a self-activated transaction is show in figure 19.

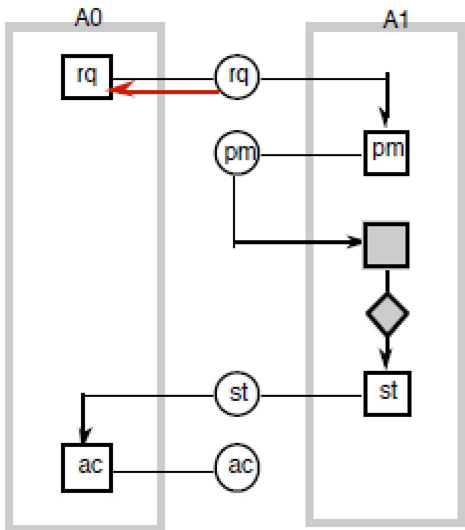


Figure 19: The structure of self-activation (Dietz, 2006a)

The composition axiom provides the basis for a well-founded definition of the notion of business process, which states that a *business process* is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment (*external activation*) or a request by an internal actor role to itself (*self-activation*). Every transaction type is represented by the complete transaction pattern (Dietz, 2006a).

This paragraph discussed the composition axiom. Two examples of the composition axiom are described. The first example was about the structure of enclosing a transaction within another transaction and the corresponding wait moments. The second example showed the self activation ability of transactions. The composition axiom provides the basis for business processes.

4.3.4. The distinction axiom

This paragraph discusses the distinction axiom on which DEMO is founded. The distinction axiom is about the different abilities of human beings that are involved in the activities they perform to realize a result.

The distinction axiom is concerned with the different abilities of a human being that are involved in the activities they perform (Terlouw & Albani, 2010). The abilities of human beings play an integrating role in constituting an enterprise. The axiom states (Dietz, 2006a):

"Three distinct human abilities play a role in the performance of coordination acts and production acts: the performa, informa, and forma abilities."

The definition of the distinction axiom states that the three distinct human abilities play a role in the performance of coordination acts and in the performance of production acts. First the relevance for the coordination acts is discussed, and second the relevance for the production acts. The process of performing a coordination act is shown in Figure 20 (Dietz, 2006a). The performa ability states that new information and knowledge can be created through communication between the performer and the addressee. Dietz (2006a) says that in order to have the coordination act from the performer to the addressee successfully performed, the performer must expose his commitment in a performative act, addressed to the addressee, and the addressee has to evoke in her the corresponding commitment to respond adequately. This act is part of the performative exchange between the performer and the addressee. For successful coordination there must be mutual social understanding of the coordination acts.

The informa ability concerns the content aspects of communication and information. In order to communicate the performer should formulate information in a way that the addressee can interpret. This means that the performer and the addressee should semantically be in an agreement with each other and share the same thoughts. This is also called intellectual understanding (Terlouw & Albani, 2010).

The forma ability concerns the significational understanding of communication and information. In order to communicate the performer should formulate a thought in a particular sentence or code scheme in some language and utter it in some form, like writing or speaking. The addressee can only perceive the informative act if they both share the same formative exchange. This means that the addressee only can perceive the informative act when she speaks the same language or reads the same words in a specific language.

With the notion of the coordination part of the distinction axiom a performer can only express his thoughts to the addressee by shaping from the performa ability into the informa ability and into the forma ability. All three distinctive levels are of great importance to perform a coordination act. So, when executing a coordination act, both actors must shape themselves into the different abilities several times.

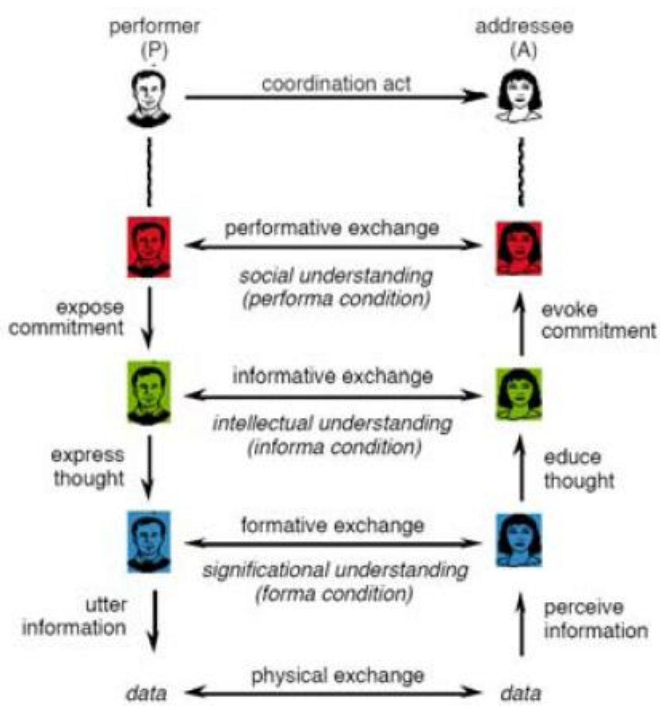


Figure 20: The process of performing a coordination act (Dietz, 2006a)

For the production acts the same distinction counts. The performa ability concerns the ability to create new original things, like the production of new material goods, or making decisions or judging about something in a context. These production acts represent the core of the business activities and are therefore called ontological transactions. These transactions are the essential activities of an organization. The ontological transactions (B-transactions) are performed by business actors or B-actors for short.

The informa ability concerns the ability to reason, compute, derive, or reproduce remembered knowledge etc. (Dietz, 2006a). The production act of one of the informative abilities is called an infological transaction. The infological transactions (I-transactions) are performed by intellectual actors or I-actors for short.

The forma ability is the ability to conduct documental activities, such as storing, retrieving, transmitting etc. (De Jong & Dietz, 2010). For all forma activities the information does not count, only the format of the data. Data formats can be digitally, documents, voice recordings, pictures etc.. The production act of one of the forma abilities is called a datalogical transaction. The datalogical transactions (D-transactions) are performed by documental actors or D-actors for short.

This paragraph explained the understanding of coordination acts between performa, informa, and forma actors. Mutual understanding between actors can only be obtained when the actors both share the same coordinative exchange. These distinct human abilities concern the distinction axiom.

4.3.5. The organization theorem

The four axioms that were discussed in the previous paragraphs provide the basis for the organization theorem. This paragraph discusses the organization theorem. This theorem provides a concise, comprehensive, coherent, and consistent notion of the enterprise, such that the (white-box) model of this notion of the enterprise may rightly be called an ontological model of an enterprise (Dietz, 2006a). It states the following (Dietz, 2006a):

"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, the I-organization, and the D-organization."

The relationships among them are that the D-organization supports the I-organization, and the I-organization supports the B-organization. The integration is established through the cohesive unity of the human being (Dietz, 2006a). Due to the social integration of the human being in the theorem, the three homogeneous systems are similar as far as coordination is concerned. They differ only in the kind of production. The production in the B-organization is ontological, the production in the I-organization is infological, and the production in the D-organization is datalogical. Figure 21 shows the layered representation of the organization theorem.

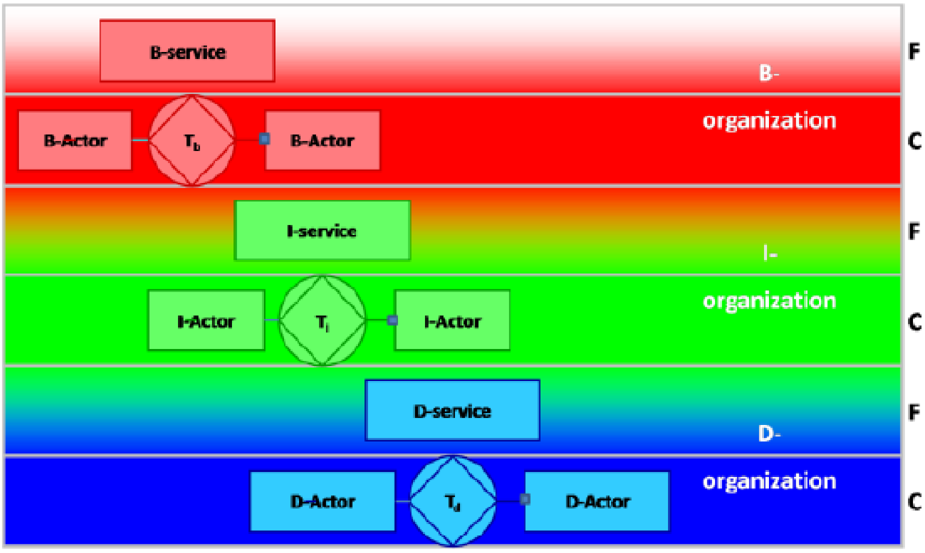


Figure 21: The layered integration of an enterprise (De Jong & Dietz, 2010)

All three layers are called aspects systems of the total organization of the enterprise (De Jong & Dietz, 2010). Every aspect system consists of a function view and a construction view.

The function view of a system is described in terms of the user of the system. It can be represented by a service the layer provides. For example the function view of an asset manager is the ability to trade on the stock exchange and beat benchmarks to gain performance. Terms of the user of the system are in this case the equity to be traded, the quote of the equity on a market, the amount to be bought, but also the sentiment in the market and the ability to predict the future to gain performance. The construction view is described in terms of components and elements and their interactions. These components and elements are not aware of the function aspects. The components and elements in the construction of the asset manager are for example a bank account, a relationship with brokers, a trading system, and an administration system. (Dietz, 2006a) says when engineering an organization it is the craftsmanship of the constructional designer of the system that bridges the mental gap between the function and the construction, such that the operation of the construction brings about the desired functional behavior of the system.

4.4. Aspect models

This paragraph illustrates the four aspects models that the DEMO methodology distinguishes. These models expresses the ontological knowledge of (the organization of) an enterprise (Dietz, 2006a). Each model discusses a different perspective of this ontological knowledge. Dietz (2006a) places the models in a triangular shape to represent their mutual relationships. See Figure 22. The top of the triangle symbolizes the most concise model of an enterprise, because there is nothing above the top. The DEMO methodology utilizes the following models: the construction model (CM) (the construction model is also known as the organizational construction model that is abbreviated as OCD), the process model (PM), the state model (SM) and the action model (AM). The following paragraphs briefly discuss the different aspects models.

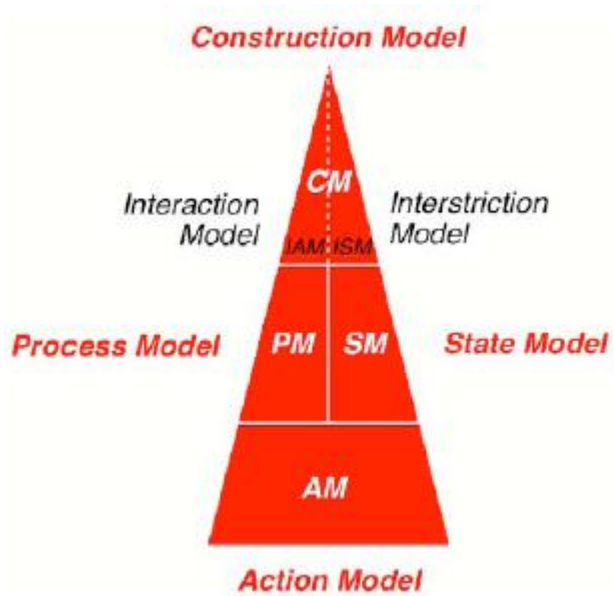


Figure 22: Enterprise aspect models of DEMO (Dietz, 2006a)

4.4.1. Construction model

The construction model is made up of the interaction model and the interstriction model.

4.4.1.1. INTERACTION MODEL

The interaction model specifies the enterprise boundaries (system boundaries), the enterprise transactions, the associated actor roles and the environment of the enterprise (Mulder, 2006). Describing an organization in an interaction model presents a concise overview of the business functions, the customers and executors, including communication that is required in operating the business functions. Within the interaction model, the result of every transaction is specified precisely (Hoogervorst, 2009). The specification of the product fact is summarized in a transaction-result table (TRT). The precise information pertinent to production facts are thus likewise defined (Hoogervorst, 2009). The TRT of Figure 23 is presented in Table 5.

An example interaction model is presented in Figure 23.

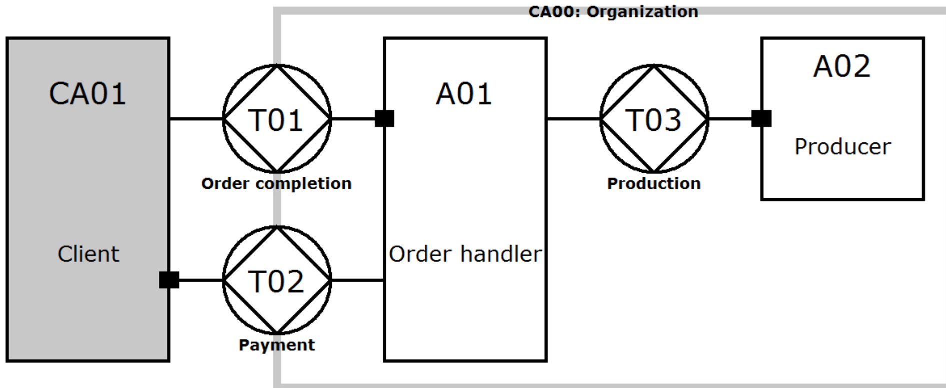


Figure 23: Interaction model (Hoogervorst, 2009)

Table 5: *Transaction-result table (Hoogervorst, 2009)*

Transaction	Result
T01 client order completion	R01 client order *CO* is completed
T02 client order payment	R02 client order *CO* is paid
T03 Producer order completion	R03 producer order *PO* is completed

The actor roles "order handler" and "producer" are called *elementary actor* roles, since they execute one transaction type. When more transactions are executed, the actor role is identified as a *composite actor* role, symbolized in grey. Because generally no information about the nature of external actors is available, they are symbolized as composite actors (Hoogervorst, 2009). The connecting lines between transactions and actors indicate the relationship. In the actor-transaction-actor connecting line, the line end with a black dot indicates the actor who has an executing responsibility over the transaction. Since an actor can execute precisely one transaction means that the other side of the connecting line has a requesting role.

The interaction model is a "timeless" description of the essential business functions (Mulder, 2006). This means that the aspect of time, or the sequence of transactions, is not an issue factually in the model of Figure 23. So the execution of T03 could succeed T02 (payment first), but the reverse is also possible (payment after order completion). In all case the interaction model remains the same (Hoogervorst, 2009).

4.4.1.2. INTERSTRICTION MODEL

The interstriction model provides an overview of the information that is required by actors in the starting, executing and finishing phase of a transaction at the essential level (Mulder, 2006). This overview is created by modeling per actor role which information sources are accessible. In the interstriction model the transaction symbol (see paragraph 4.3.1 and 4.3.2) is interpreted as the combination of a production bank and coordination bank. The accessibility of an actor and the bank is represented by a dashed line. The information links between actors and banks restrict the nature of the interaction to the information exchanged. That is why the model is called the interstriction model (Hoogervorst, 2009). See Figure 24 for an example of interstriction. In the example the external production bank PB01 with “terms of order processing” restricts the freedom of the actor “order handler” to produce transaction T01.

The interstrictions in the model are abstracted from their implementation. This means that the information link does not mention how the actor gets its information and also not mention if the actor must pull for the information or if the actor gets the information pushed (Mulder, 2006).

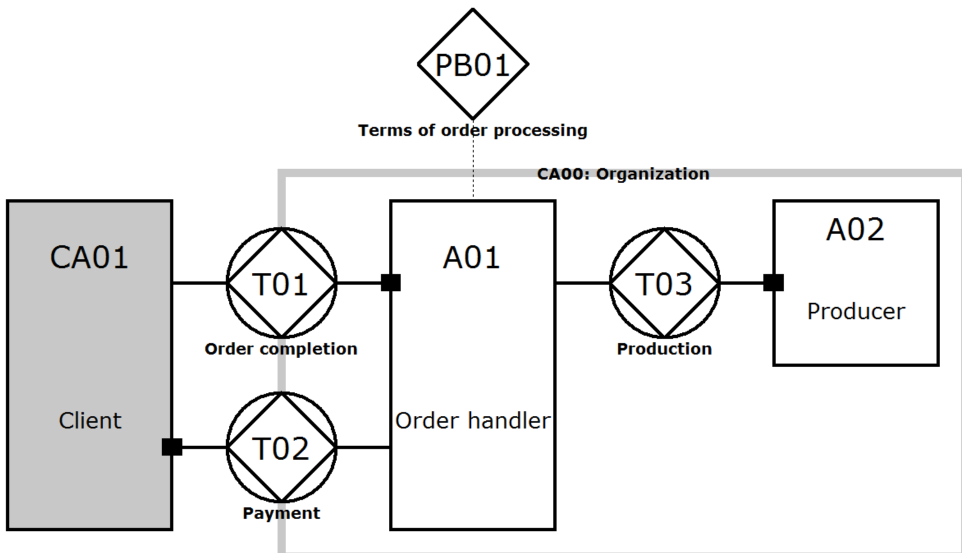


Figure 24: Interstriction model (Hoogervorst, 2009)

4.4.2. Process model

The process model is a global description of the time sequential relations and business rules between transactions and the different phases of coordination from which a transaction exists (Mulder, 2006). The process model captures two basic types of relations: causal and conditional relations. A causal relation between two transactions (phases) means that one transaction (phase) causes another transaction (phase) to start. A conditional relation means that finishing a certain transaction (phase) provides a condition for the finishing of another transaction (phase) (Mulder, 2006). Furthermore the conditionality is utilized to indicate whether a transaction is started given some validated context. Figure 25 (Hoogervorst, 2009) presents an example of the process model that is based on the interaction model in Figure 23.

Unlike the interaction model, the sequence of actions is identified in the process model. Completion of T01 necessitates the transaction T02 and T03. Hence there are causal links between T01/pm and T02/rq and T03/rq. The implied payment request becomes formal when the production of the producer (T03) is accepted by the order handler. This waiting condition (conditional link) is indicated by the dotted arrow. After the payment transaction is completed, transaction T01 is ultimately completed, as is similarly indicated by the waiting condition (Hoogervorst, 2009).

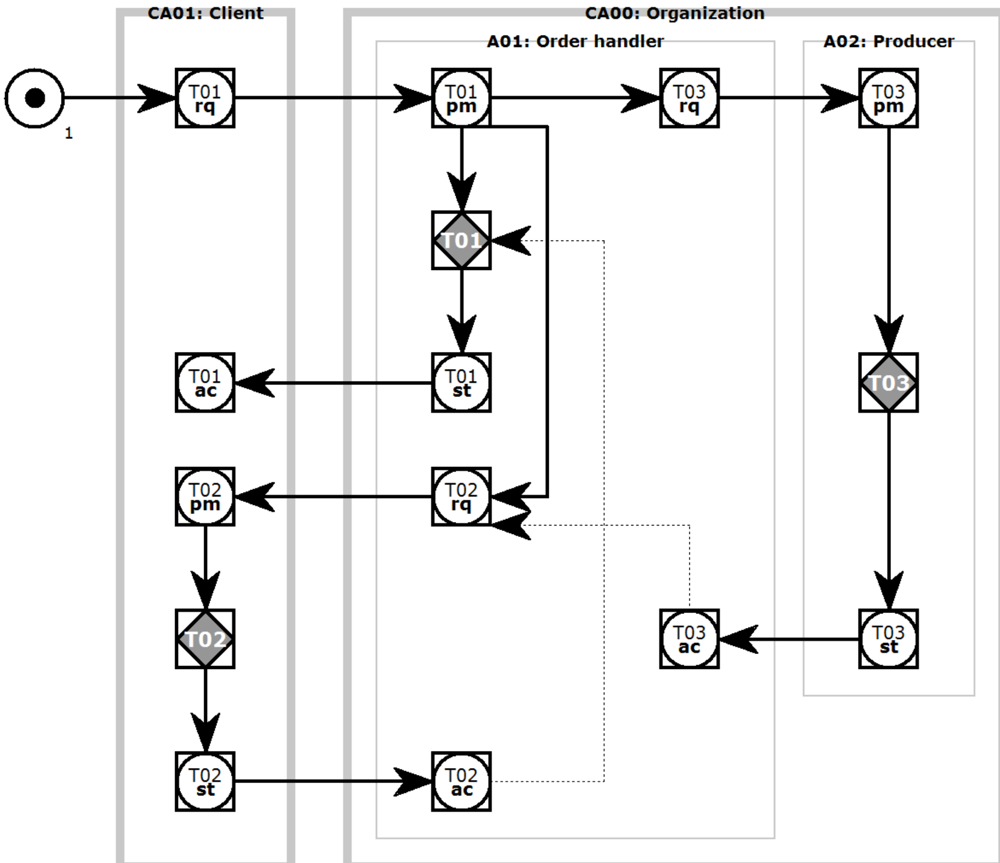


Figure 25: Process model (Hoogervorst, 2009)

4.4.3. State model

The *enterprise state* was defined as the totality of coordination and production facts at (or created up to) a certain moment in time. The totality of lawful states the enterprise can be in was identified as the *state space*. Within the enterprise ontology methodology, the state model is restricted to the production facts of the state space, since production has to do with the very purpose of the enterprise and its possible transactions. Production is about the realization of a material or immaterial fact, such a preparing a meal or assessing the value of a house. These facts concern so-called "objects". Facts say something about objects: that a meal is prepared, or that an assessment is completed. Objects are concrete or abstract things like the ones mentioned, and are an element of the respective object class. The state model specifies the state space of production facts by depicting the production facts pertinent to objects in the object class, and by showing the logical relationships between the object classes. So the model shows what possible production facts are associated with the respective objects (Hoogervorst, 2009).

Mulder (2006) concludes that the interstriction model and the state model are closely related. The interstriction model gives an overview of where and what kind of information is utilized by actor roles. The state model gives an overview of the how the information is structured. The theory and graphical notation of "Object-role modeling" is used within the DEMO methodology (Dietz, 2006a). See Figure 26 for an example of the state model that is related to the interstriction model of Figure 24 on page 68.

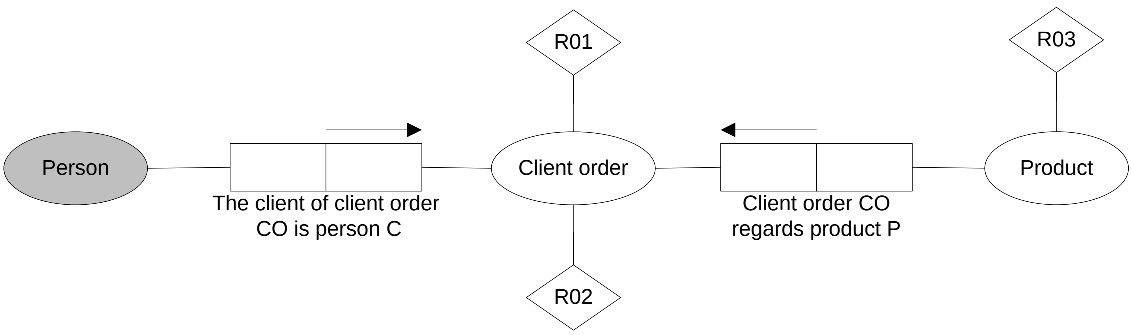


Figure 26: State model (Hoogervorst, 2009)

4.4.4. Action model

The action model is the most detailed and comprehensive aspect model (Dietz, 2006a). The model contains a procedural description of all actions for all actor roles and all transaction phases. Furthermore the action model describes the cohesion between the actions (Mulder, 2006). Just like the process model the cohesion between actions is a result of causal and conditional relations (Mulder, 2006). The action model is not concerned with production work instructions, but only with the handling of coordination facts. Coordination activities address the occurrence of a coordination fact, are performed by an actor role, and are guided by so-called *action rules*. So generally, there are action rules for the occurrence of a request, promise or decline, statement, and acceptance or reject, as well as for the associated cancellations (Hoogervorst, 2009). See paragraph 4.3.2 for the transaction axiom and Figure 27 for an example of action rules. The totality of action rules for guiding the coordination activities of the various transactions is called the *action model*. This model is important since the action rules define the process execution. Because of the explicit definition of the coordination activities, the precise definition of the required information is also established (Hoogervorst, 2009).

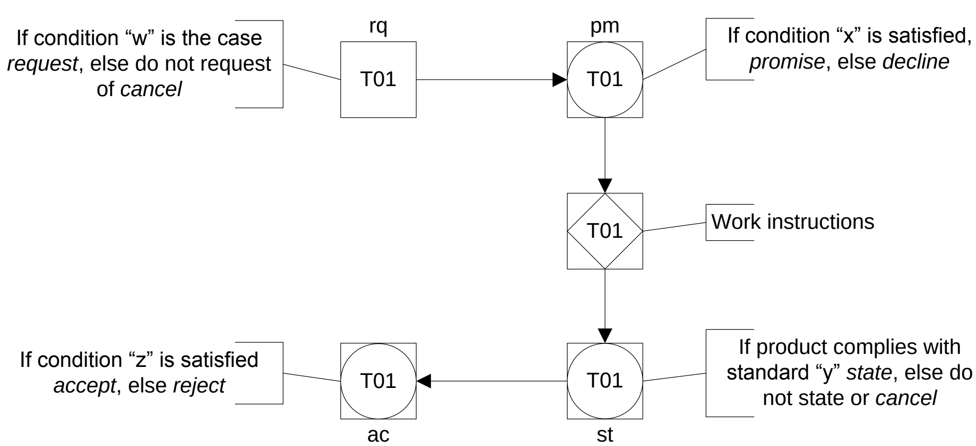


Figure 27: Action rules (Hoogervorst, 2009)

4.5. Summary

This chapter introduced the ontology of the informational organization. The information organization illustrated the three activity levels of enterprises and showed that every activity level itself concerns the three activity levels. This recursive behavior shows that the foundation of the DEMO theory applies for every layer.

The underlying theory of DEMO is the PSI-theory. The PSI-theory consists of four axioms and an organization theorem, which are the foundation of DEMO. The four axioms are listed below. The organization theorem is summarized separately.

- The operation axiom distinguishes between production acts and coordination acts. The acts have effect in the production world and in the coordination world respectively. Coordination acts are always, either directly or indirectly, about production acts. A production act is either material or immaterial.
- The transaction axiom is the universal pattern of coordination acts and production acts. The main states of the transaction axiom are: request, promise, state, and accept. If this pattern is extended with discussion states, then the standard transaction pattern is obtained. Besides the standard transaction pattern, four cancellation patterns exist. The combination of both is considered to be a socionomic law.
- The composition axiom is about the structure of enclosing a transaction within another transaction and the corresponding wait moments. Another aspect of the composition axiom is the self activation ability of transactions. The composition axiom provides the basis for business processes.
- The distinction axiom concerns the understanding of coordination acts between performa, informa, and forma actors. Mutual understanding between actors can only be obtained when the actors both share the same coordinative exchange. These distinct human abilities concern the distinction axiom.

The organization theorem is explained by the three homogeneous systems and the function and construction relationship between then. The primary conclusion is the supporting role of the function of an aspect system toward the construction of an aspect system, wherein the function of the D-organization supports the construction of the I-organization, and the function of the I-organization supports the construction of the construction of the B-organization.

The service-centered point of view of every homogeneous system's function correlates with the service type classification of Steghuis (2006). From a service-centered angle the B-organization provides business services, the I-organization provides informational services and the D-organization provides document or data services. The business services, information system services and software services of Steghuis (2006), see table 3 on page 31, are respectively the business services, informational services and data services.

The axioms underlying the DEMO methodology are expressed in four aspect models: construction model, process model, state model and action model. The main characteristics per aspect model are listed below.

- The construction model consists of the interaction model and the interstriction model. The construction model specifies the construction of the organization.
- The process model contains, for every transaction type in the construction model, the specific transaction pattern of the transaction type. It shows the sequential order of activities from the transaction and relationships between transactions.
- The state model specifies the state space of the production: the object classes and fact types, the result types, and the ontological coexistence rules.
- The action model specifies the action rules that serve as guidelines for the actors in dealing with their transaction type.

5. EXPLORATIVE CASE STUDIES

This chapter introduces the two explorative case studies that have been conducted in the spring of 2009. One case study has been conducted at Pretium Telecom, a Dutch telecom company that has its seat in Haarlem. The second case study has been conducted at the HU University of Applied Sciences in Utrecht, the Netherlands. This case study was about a fictitious value chain. Both case studies share a common approach in enterprise architecture, using DEMO and the principles of SOA. This common foundation makes both case studies applicable for this research. After the introduction of the case studies, paragraph 5.2 provides a thorough introduction in constructing the information organization. The joined knowledge with chapter 4 and this paragraph gives an answer to sub question 6 "*What is the DEMO informational construction model*". This understanding is prerequisite before the results of the explorative case studies can be examined in paragraph 5.3. The chapter concludes by summarizing the results and findings of the two explorative case studies in paragraph 5.4. To complete this chapter the full article about the explorative case studies that has been submitted to the International Information Management Association Conference in Utrecht, the Netherlands is available in appendix Q.

5.1. Case study introduction

The case study conducted at Pretium Telecom took about one month in the spring of 2009. The organization provides telecom services for about one hundred and fifty thousand customers in the Netherlands. The company employs forty five people who are responsible for providing the telecom service, of which twelve employees are located in the in-house IT department. The occasion of the case study was the planned introduction of a new telecom service. To support the new service the company wanted to know if the new service would fit within the current organization and could be supported by the current IT systems. To advice the company an overview of the essential business activities was made and per activity the information needs that are required to perform the activity were analyzed. The essential business activities were modeled with DEMO.

The second case study conducted at the HU University of Applied Sciences took about six weeks in the spring of 2009. The occasion of the case study was the final group assignment of the Business Process Management course. The assignment was to create the enterprise architecture and the integrated IT system of a fictitious value chain for a company which is assembling and selling bikes. We called the case study "Iron Horse", to clearly express the primary essence of the business. To meet the objectives of the assignment we started by modeling the essential business activities. Those activities were expressed with DEMO. Per activity the information requirements have been analyzed. Following this strategy resulted in an information model that was completely covering the business essential activities and was directive to software system developers.

5.2. Construction of the information organization

Chapter 4 introduced the four axioms and the organization theorem on which DEMO is founded. The organization theorem addressed the layered integration of an enterprise, showing an ontological layer, an infological layer, and a datalogical layer. Every layer has a function view and a construction view, representing the use of the system and how the system is build respectively. This paragraph is about the construction of the information organization, meaning the construction of the infological layer. The construction view of the infological layer shows how the information needs of the ontological layer are constructed. According to the DEMO way of thinking, information must be understood as supporting B-actors in their coordination acts. To be more specific, there is no information without communication, since information is produced only for the purpose of communicating (Dietz, 2001). In context of information utilization by B-actors, the transformation of data into information is knowledge (De Jong, 2008). Knowledge can be distinguished in explicit knowledge or information and implicit knowledge or tacit knowledge (Polanyi, 1966), (Nonaka & Takeuchi, 1991), (Weggeman, 2003). Explicit knowledge refers to knowledge that can be externalized in terms of some representation, like words in a language or symbols representing an informational model. Tacit knowledge cannot easily be represented on a medium, but must be learned by experience like bicycling. The construction model of the information organization is a description of a system (De Jong & Dietz, 2010). System descriptions are essentially forms of explicit knowledge pertaining to an existing or future system (Lankhorst, 2005). This means that the construction of the information organization supports explicit knowledge only.

When B-actors produce new facts, these facts are archived in facts banks. A fact bank is the transaction itself, which contains all the facts that are produced by that particular transaction. An I-actor understands these production facts as elementary building blocks for creating information products in order to support B-actors (De Jong & Dietz, 2010). This means that the infological layer is of great importance for the business in performing its business acts. Therefore the construction of the infological layer must be optimized to provide a service level that fits like a tailored suite to the B-actors. Optimizing the construction of the infological layer is also mentioned by De Jong & Dietz (2010). They state that in practice a derived fact can be calculated by several I-actors, each with their own responsibility and authorities. This is understood as the granularity issue of the construction of the infological organization. The main research question of this thesis is about assigning the granularity question with the construction view of the infological layer. An optimized infological layer supports the business layer at best, but is also directive to link supporting IT-applications (De Jong & Dietz, 2010).

This paragraph introduces the construction of the information organization. The infological layer has been placed between the ontological layer and the data layer and its primary function is to support the business actor’s acts. The function of the infological layer is established by its construction, which only facilitates explicit knowledge. The following paragraphs discuss the way of working to create the infological model, starting from the ontological model.

5.2.1. Why the informational construction diagram?

The informational construction model is the intermediate layer between the business layer and the data layer, from an organisational theorem point of view. This layer has data as input and information as output. The informational layer creates information that is used in business operations. Therefore the informational layer can be seen as the services layer on top of data that needs to be programmed by software engineers and are specified by business analysts. The informational layer is the layer that needs to be (partly) implemented in software systems. Therefore on the informational layer the issue for getting the service granularity right is the most visible.

5.2.2. Infological analysis

The introduction of this paragraph pointed at the purpose of the infological model. Its goal is to support the business actors in their coordination acts. All the coordinative acts are represented by a single transaction between two actors. Both actors have information needs before performing the act (De Jong & Dietz, 2010), (Op 't Land, Proper, Waage, Cloo, & Steghuis, 2009). The actors need to know if their act will be valid, considering the context. This means that an actor has a specific information need that follows the transaction pattern of the transaction axiom and is embedded as sequence of information usage by its coordination acts in the process structure diagram. Accessing information follows the distinction axiom, where a business actor can shape into an informative actor to request the necessary information. This paragraph addresses the information needs of a business actor by applying the process of information requirements analysis per coordinative act. This way of modeling is explained next.

According to Shishkov & Dietz (2003), the B-organization provides a starting point for the definition of functional requirements of information systems. The process of requirements analysis is trying to figure out what the actor needs in performing coordination acts (Fowler, 2008). The requirements are gathered via several ways like conducting interviews with actors who are really performing transactions. Regarding the requirements analysis Dietz (2006a) says that the transaction pattern is very useful as the starting point. The process model is created based on the transaction pattern and contains all the information an actor needs.

To structure the process of requirements analysis that follows the transaction pattern De Jong (2008) developed a table that contains situational information and operational information per agenda, see Table 6. All columns of the table are briefly discussed first. The agenda is equal with a coordination fact of a transaction, which is discussed in the transaction axiom paragraph 4.3.2. The act column describes the act to be performed. It is a short description of what is going to happen. De Jong (2008) describes that situational information is considered to be the service that delivers data to inform an actor about the next coordination act to take. The situational information can be derived from the action model of DEMO. This column will be elaborated further on. The operational information concerns the services that deliver the data to support the execution of the production act in the business organization. This is considered to be the information needed to process the data, like a procedure or an operational list with data. The last column of the table states the coordination acts to be performed next, when the agenda is processed.

Table 6: *Infological analysis table (De Jong, 2008)*

Agenda	Act	Situational information	Operational information	Data to process

To elaborate on the description of the situational information column, De Jong (2008) says that it can be derived from the action model of DEMO. Regarding to Dietz (2006a) the action model of an organization consists of a set of action rules. An action rule serves as a guideline for an actor in dealing with its agenda and influences the actor's acts. This means that situational information is synonym for action rules.

5.2.3. Intellect system

This paragraph provides a theoretical foundation on creating the infological construction diagram. The construction of the information organization could be perceived as an ontological notion which addresses the construction and operation of an aspect system of an enterprise. This is supported by the organization theorem, which has been discussed in paragraph 4.3.5, that states that the organization of an enterprise is a heterogeneous system which is constituted as the layered integration of three homogeneous systems; the B-organization, the I-organization, and the D-organization.

The B-organization is defined as in the category of a social system. In Dietz (2003), a social system is defined as a system of which elements are social individuals or subjects who perform two kinds of acts: production acts (P-acts) and coordination acts (C-acts), and thus produces P-facts and C-facts, respectively.

The I-organization is not perceived as a social system since C-acts are not considered essential in informative exchange (Sandhyaduhita, 2009). The I-organization concerns only the production aspects, the P-acts. The actors in this kind of organization are subsequently considered production actors only. Accordingly, the actor roles in the I-organization are defined as elementary chunks of responsibility and authority, fulfilled by subjects in performing P-acts and producing P-facts respectively (De Jong & Dietz, 2010).

Since the P-act in the I-organization corresponds with the informa human ability, refer the distinction axiom in paragraph 4.3.4, the subject fulfilling the I-actor role is called Intellect-actor (I-actor). Therefore the construction of the information organization can also be called the intellect system.

This paragraph provided the theoretical foundation for the construction of the information organization. It pointed at the elementary chunks of responsibility and authority of I-actors within a non-social system. The information organization concerns the P-acts only and is called an intellect system.

5.2.4. Production acts and facts

This paragraph discusses the utilization of the production acts and production facts of an informational organization.

The production acts of the I-organization are infological acts which are characterized by the informa ability to reason, compute, derive, etc., as well as to reproduce remembered knowledge (Terlouw & Albani, 2010). Since the I-organization is defined as a realization of the B-organization, the knowledge or information produced by the I-organization are about the production world of the B-organization. Consequently, the P-facts of the I-organization can be seen as the results of production acts of the informa ability of an actor (Sandhyaduhita, 2009).

5.2.5. Infological modeling notation

The I-organization will be modeled following the guidelines and notation of an Interaction Diagram (IAM). To model the I-transaction of the IAM of the I-organization, the transaction symbols of the B-organization will be used. However, the concepts implied by the symbols about the banks and the coordination act/fact will be ignored. The complete legend for the I-organization is given in appendix B.

5.3. Examining the case study results

This paragraph briefly illustrates the research process of the two explorative case studies. In accordance with the structured case method of Carroll & Swatman (2000) the steps of the applied research process are explained.

5.3.1. Conceptual framework

In paragraph 5.2.2 is stated that the B-organization provides a starting point for the definition of functional requirements of information systems (Shishkov & Dietz, 2003). Therefore conceptual framework in both case studies was the Organization Construction Diagram of DEMO. This diagram captures all the essential activities of the business and is the organizational layer that must be supported by the informational layer.

5.3.2. Plan

Both case studies aimed at identifying an optimum informational service model by utilizing DEMO and the process of creating the informational construction diagram. The structured case method approach was used to compare the results afterwards, because both case studies were executed separately from each other. By using the same frame of reference the comparison has been made possible.

There was no specific criterion which must be met to select the case studies. The only prerequisite was the possibility to apply DEMO and the case study scope was limited.

5.3.3. Collect data

Both case studies started with creating the organizational construction diagram (OCD). This diagram is used as the conceptual framework, but was not available in both cases. At Pretium Telecom an open interview[3] has been conducted first. Using this interview technique provided a global overview of the main objectives of the business. The interviewees were the Chief Information Officer and the lead Technical Architect. The goal of the interview was to get a picture of the business activities, so an OCD could be created. In the period after the interview the OCD was designed. The resulting diagram has been discussed twice with the interviewees before they agreed on the OCD. This approval was the starting point for the case study.

The information gathering process for the assignment at the HU University of Applied Sciences started with a description of the fictitious organization on paper. This information in combination with an open interview with the lecturer, who was playing the role of business owner in the fictitious case, provided enough information to create the OCD. The resulting model has been verified during the course.

5.3.4. Analyze

In both cases the OCD has been created and verified with the business. This model was the starting point to apply the informational analysis to get an informational construction diagram. The applied methodology followed the description of the infological analysis in paragraph 5.2.2. This process was a practical execution of a purely theoretical process. In order to achieve an optimum informational services model, which supports services on the aspects of reusability and flexibility in business processes, the determinants of service granularity (see Table 3 on page 31) were tested. All the determinants of service granularity per service type were tested against the DEMO methodology. Testing the service granularity determinants with the concepts of DEMO meant conducting literature research about the specific service granularity determinant and reviewing the theoretical behavior of it with the concepts of DEMO. The DEMO concepts are explained in chapter 4.

Per service granularity determinant and per service type the match with DEMO has been made. This resulted in overviews which service granularity determinants were supported by DEMO and which were not. See paragraph 5.4.

5.3.5. Reflect

The explorative case studies connected determinants of service granularity with DEMO. This resulted in an overview of service granularity determinants that are and are not supported by DEMO. Besides, the explorative case studies provided a conceptual framework to apply informational analysis within a DEMO way of working. The explorative case studies showed that following just the theoretical approach is not always sufficient to get a clear picture on the total information requirements. This conceptual model is further explained in paragraph 5.4 and will be the basis for the improving case study.

[3] The conducted interviews at Pretium Telecom are not part of this research project. Accordingly no transcriptions are integrated in this research project.

The findings of the exploratory case studies are published in (Wiersma, 2009a) and (Wiersma, 2009b) and are about to be published in (Wiersma & Ravesteijn, 2010). The full article that has been submitted to the International Information Management Association Conference is available in appendix Q.

5.4. Findings

The explorative case studies showed that DEMO is supporting some of the determinants of service granularity on the business layer as well as on the informational layer. Regarding the service granularity determinants on the data layer, the conclusion is that DEMO is not supportive, with respect to the organizational theorem.

A part of the main research question of this thesis is about defining the right granularity for services using DEMO's informational construction diagram. Therefore the findings of the explorative case studies regarding to the informational analyses are subject for further analyses. From the explorative case studies two things can be learned. The first aspect is the supportiveness of DEMO regarding the determinants of service granularity. For reasons of completeness the DEMO support on service granularity on the informational layer of the organization is shown in Table 7.

Table 7: *DEMO supporting information service granularity determinants*

Information System Service	DEMO support
Functionality	Supported by the fact that every information service is primary supporting the construction of the business layer.
Flexibility in Business processes	The flexibility of the business process is determined at the ontological level. The informational layer exists by supporting the business process and provides information towards that business process. How the information is being interpreted is not relevant within the informational layer.
Cognitive and Structural Complexity	The informational layer itself does not say anything about complexity of the implementation of this kind of services. The informational model does provide guidelines for implementation.
Reusability	Reusability of information services becomes directly clear in the informational construction diagram when multiple initiating actors request process executing of an executing actor.
Composability	The informational model provides a clear overview in the essential information transactions that are required in supporting the business layer. Information analysis will show

Information System Service	DEMO support
	when an information object must be composed from other information objects.
Reusability of Legacy	See reusability.
Sourcing	The infological model can make proposals supporting decision making on sourcing of informational actors within or outside the organizational borders. The infological model expresses this by creating new integration services that are associated with the external information provider in case the information provider is placed outside of the organization.
Genericity	Due to modeling essential information services only, a generic and concise information service can be designed.
Context-independence	Every information services is responsible for collecting its own data. The information service is not dependent on information from other information services, but maximum limited by the executing order within the business process.
Performance	The infological model is independent of any form of implementation. Therefore the infological model cannot provide a guideline for performance.

The second finding is a conceptual model how to perform informational analysis from a DEMO perspective. This conceptual model is extracted from existing literature, e.g. (De Jong & Dietz, 2010), (De Jong, 2009). In order to use this conceptual model as a frame of reference for an optimized informational services model when applying informational analysis from a DEMO perspective, the conceptual model must be tested. Testing the conceptual model was the goal of the improving case study, which will be introduced in the next chapters. The conceptual model from the explorative case studies is shown in Figure 28.

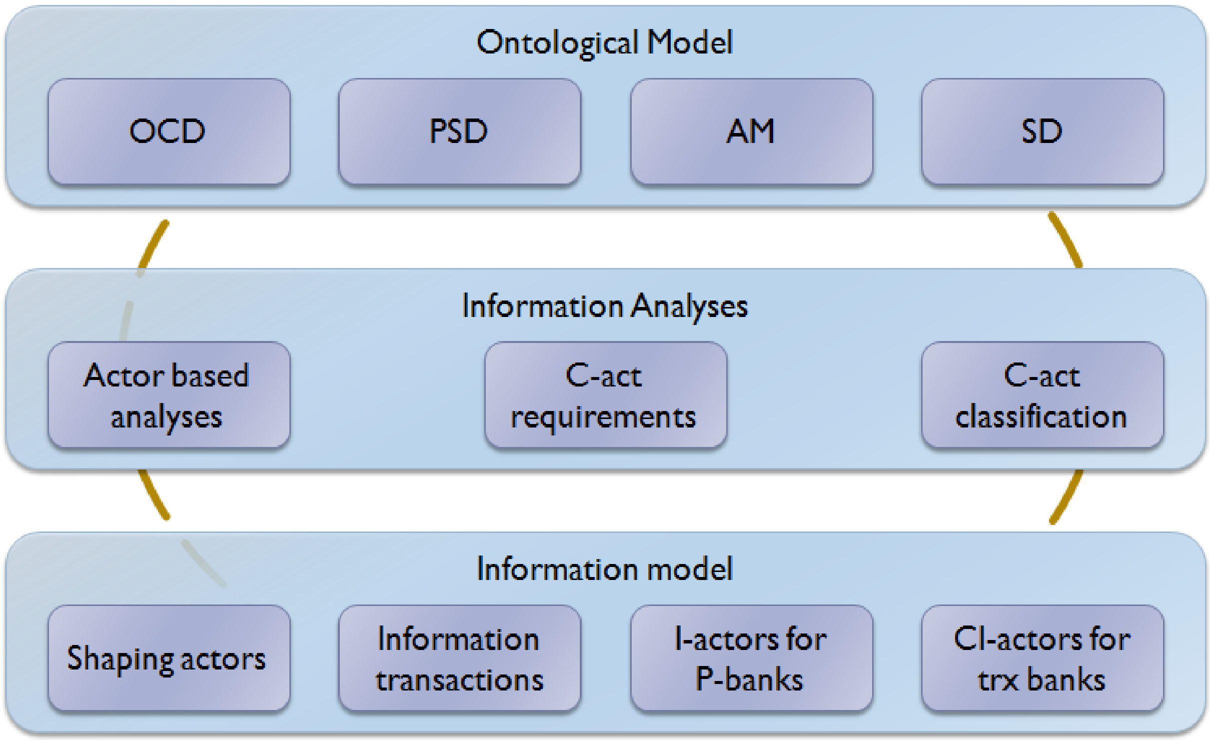


Figure 28: Conceptual model for information analysis

The conceptual model in Figure 28 expresses two states, the ontological model and the information model, and the transition to be performed to get from the starting state into the goal state, which is the information analysis. The start state, when modeling with DEMO, is the ontological model. The four terms within the ontological model area are the four aspects models of DEMO, OCD for Organizational Construction Diagram, PSD for Process Structure Diagram, AM for Action Model, and SD for State Diagram. These four aspect models express the ontological view of the business organization. This research's objective is to model an optimized service model, which is the information model area. The characteristics of the information model area are projected, which are shaping actors, information transactions, I-actors for Production (P)-banks and composite informa (CI)-actors for transaction banks. These four characteristics are explained below.

- Shaping actors; this is the behavior of the business actors who shape into their informa ability to perform infological transactions. As stated by De Jong & Dietz (2010) an informa can only execute information transactions and a performa can only execute business transactions. This also turns out from the distinction axiom in paragraph 4.3.4. When an actor shapes, it become possible for that actor to invoke transactions on the organizational level to which it shaped. The shaping ability of actors is the linking relationship between the different layers of the organization.
- Information transactions; these are transactions with a clear purpose. As stated by Dietz (2006a), the informa ability concerns the ability to reason, compute, derive, or reproduce remembered knowledge etc. The information transactions are the information facts that are produced when an informa performs one of its abilities.

- I-actors for P-banks; this means that for every external production bank in the OCD there must exist and informa to read from the external production bank. Because a production bank is a data collection that has been created outside of the scope of the organization under consideration, the production bank can be read but never be modified directly. The informa is responsible for gathering the information from the external production bank and provide the information to the business actors who require that information.
- CI-actors for trx banks; this means that for every transaction bank in the OCD there is a composite informa (CI) that handles the information flow from the forma towards the performa and vice verse. Because a transaction in the OCD contains all the p-facts of that transaction, those facts must be stored at the data layer or be retrieved from the data layer. Routing these p-facts is an information flow that is managed by the informa's. Because of the close relationship with the data layer it is seen as a best practice to model the informa's, who are responsible for this information flow, as a composite.

The transitional area of the conceptual model illustrates the behavioral part of the information analysis that must be performed. The objective of this area is about the process of implementing an information system in the end. These are the activities that must be performed by the analyst to create an informational model that has been extracted from the ontological model. These activities are actor based analyses, c-act requirements and c-act classification. The activities are explained below.

- Actor based analyses; this means that the starting point of the information analyses is always the actor. The information analysis approach is therefore always centered on the information requirements that an actor has. In other words, what are the shaping possibilities that a business actor must have to perform its essential business activity?
- C-act requirements; the coordination act requirements are based on the process structure diagram. The process of requirements analysis is trying to figure out what the actor needs in performing coordination acts (Fowler, 2008). Regarding the requirements analysis Dietz (2006a) says that the transaction pattern is very useful as the starting point. The process model is created based on the transaction pattern and contains all the information an actor needs. To structure the requirements of the c-acts Table 6 of De Jong (2008) on page 76 is used.
- C-act classifications; the classification of coordination acts illustrate the different types of services that exists. According to Terlouw & Albani (2010) there exist human services that are executed by human beings, and there exist IT services which are services executed by IT systems. Those two types of services exist per aspect type of the organization. This means that the services exist on the business layer, the information layer, and the data layer. Aalst & Hee (2008) defines an extra type of services. They state that services are not primarily executed by human beings or IT systems, but the combination exists as well. These so-called semi-automatic services consist of a human being service execution part as well as an IT-systems execution part. In classifying coordination acts the information analysis approach follows the definition of Aalst & Hee (2008). All the coordination acts are classified if it is an automated, semi-automated, or a manual act. This classification has a directive nature for software system developers.

To indicate that the process of designing the information model is iterative, the yellow/brown colored dashed line in Figure 28 is made partly visible. This circle expresses the iterative nature of the process.

The conceptual model itself has its foundation in the generic system development process model of (Dietz, 2006a). The generic system development process denotes the system to be designed as the “object system” (OS). This system delivers its function to certain elements of the environment. The system that requires the function of the OS is considered the “Using System” (US). For designing the function of the OS (object system function) properly, the construction of the US – its ontology – must be known (Hoogervorst, 2009). Both the function and construction design are guided by their respective architecture. The generic system development process is shown in Figure 29. The process within the conceptual model is to get the picture of the object system function as clear as possible. Just like the generic system development process, the conceptual model starts from the ontological model of the US. The objective is to create the informational model, which is the OS of the informational layer. The OS of the informational layer itself also has an ontology, which is the construction of the services and other software objects. This means that the transitional area of the conceptual model is covering the functional design part and partly the constructional design of the generic system development process. Hoogervorst (2009) says that the generic system development process portrays the function and the construction design as sequential phases, but that most likely some iterations will take place because constructional issues might have an effect on the function design. The explorative case studies substantiate this statement. Therefore the dashed circle in the conceptual design, which expresses an iterative process, has been added.

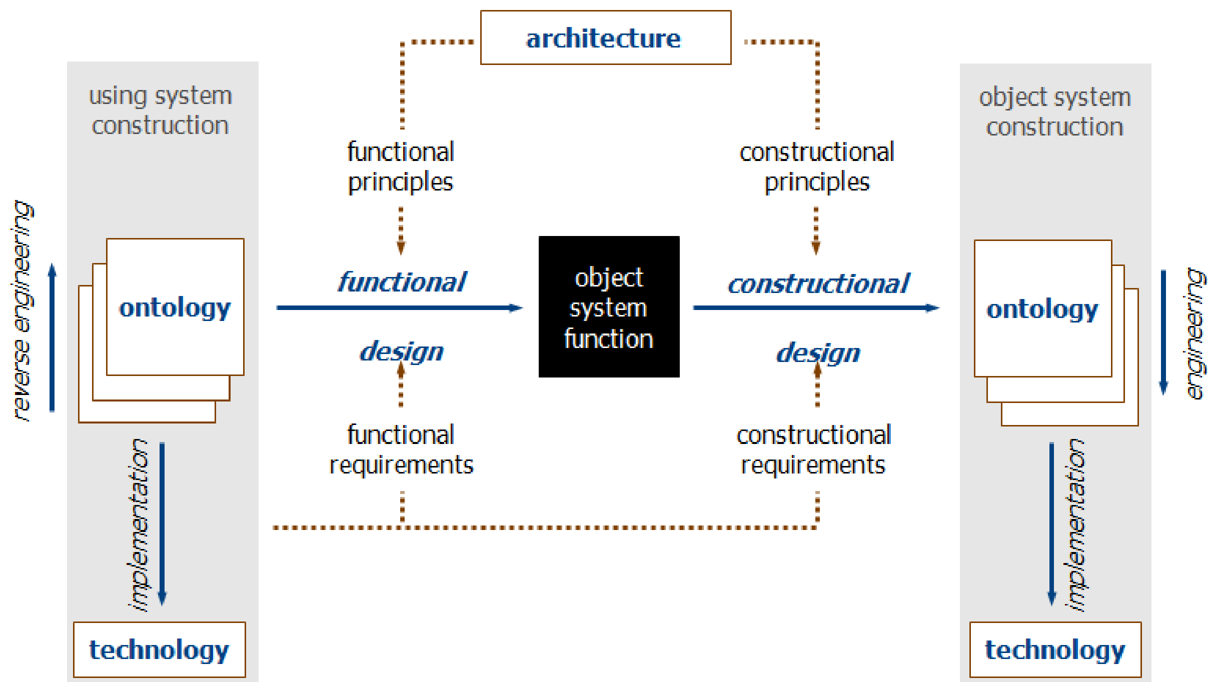


Figure 29: Generic System Development Process (Dietz, 2006a)

5.5. Summary

This chapter illustrated the explorative case studies that have been conducted. The main objective of the explorative case studies was trying to find an optimum in service granularity by utilizing DEMO's informational construction modeling. The findings were that DEMO does meet some of the determinants of service granularity as stated by Steghuis (2006) and that constructing the informational model is an iterative process, starting top-down from the ontological model and applying informational analysis activities.

The findings in the explorative case studies were based on the field studies and extensive literature review. In the improving case study, which is illustrated in the next chapter, the findings are challenged and ultimately validated by an expert panel review.

6. IMPROVING CASE STUDY

This chapter discusses the improving case study that has been conducted at Alpha in the winter and spring of 2010. The objective of the improving case study was to validate the conceptual model from the explorative case studies in finding an optimum in service granularity using DEMO's informational construction diagram. This chapter first introduces the case study in paragraph 6.1. After the introduction the research approach is explained in paragraph 6.2. The applied research approach followed the action research cycle of Baskerville & Wood-Harper (1996). Every phase of the action research cycle is described in the following subparagraphs. Because we conducted three iterations in total, three action research cycles are discussed. The first iteration starts in paragraph 6.2.2.3.1, the second iteration starts in paragraph 6.2.2.3.2 and the third iteration starts in paragraph 6.2.2.3.3. The action taking phase, that captured all three iterations, is evaluated in paragraph 6.2.2.4 and the lessons learned are illustrated in paragraph 6.2.2.5. The chapter ends with a brief summary in paragraph 6.3.

6.1. Case study introduction

This paragraph introduces the company and the case study. The improving case study considered a real life project. For confidentiality reasons the company name including the names of the attendees and the name of the projects are scrambled. All names or theme's that are mentioned in the case description are fictitious. This thesis refers to the company by naming it “Alpha”.

The case is a planning and coordination project for a national government organization which is responsible for the coordination and operation in international projects. This organization works under extreme pressure when it executes its core business processes. Information management is crucial while planning and coordinating all kind of activities. All activities have to be aligned using the most recent available information that exists. Therefore information management is a key factor for successful project execution.

The case study project is started to structure the requests for information access at Alpha. The complexity of the requests requires structured and precise coordination between several hierarchical layers. Therefore DEMO has been chosen as preferred methodology to support the projects analysis. The project is in the area of access control and user management to control the usage of networks.

The business architects of Alpha discovered a generic process that counts for all the planning and coordination projects that the organization executes. This process is designed at conceptual level in DEMO. This means that the Organizational Construction Model and the Transaction Pattern Diagram[4] has been designed, including a Transaction Result Table. The Organizational Construction Diagram is represented in the most abstract level of the used methodology. At this level a composite actor execute high level business transactions. For one composite actor a drill down is created. At that level an elementary actor execute elementary business transactions.

[4] DEMO version 3 equivalent of Process Structure Diagram in DEMO version 2.

All business architects involved in the project are basically trained in the methodology of DEMO in the last year. They are all DEMO certified. After certification several projects were started. This project is seen as kind of a pilot study for the rest of the organization. If the project is successful and the used methodology fits within the organization, then the methodology will be integrated in the current systems analysis methodology.

The reason for the project is the repetitive occurrence of comparative business processes in a different context. Due to changing contexts it makes it hard to generalize business process that abstracts from the context. The DEMO methodology does abstract from context and implementation and focuses on essential activities only. This leads into a business model that was generic for the planning and coordination activities. When the model was validated with the business by the business architects, the business concluded that the model was representing the activities that take place when the business process runs.

The project is guided by several principles:

- Every planning and coordination process consists of three phases which are preparation, execution, and closing.
- There is a formalized hierarchy in control and decision making.
- The generic process is actor and responsibility based.
- The generic process is a representation of actors who are performing business transactions.

These principles aligned with the current way of working in information management projects and are aligned with the current system development method.

The reason for the improving case study was twofold. First, the organization was facing difficulties in translating the abstract models of DEMO into something concrete and understandable for system developers. Basically this issue is addressed within the applied system development method of the organization. The organization uses an adjusted version of the V-Model (Brook, 1986). The first two phases, Business Case and Requirements, are described on the conceptual level. The company uses an Organization Concept Description document for this purpose. The DEMO models that are mentioned before are part of this document. The next phase in the V-Model is System Specification. In this phase all the previously collected and described requirements are passed through the system developers. The organization experienced that the gap between the conceptual models and requirements is too large for the system developers to create a software design for. This gap causes that the system developers take too much assumptions when developing software, which results in software systems that are poorly supporting the projected business transactions. Based on their experiences the organization is looking for a method that is directive in translating the conceptual model into software. This directive approach must result into fewer assumptions made by system developers and counter wise contributing to the enterprise architecture.

Second, the service-oriented approach of DEMO related to the project triggered the question to define a service model. The main research questions in this thesis were a fit on both topics. The relationship between the senior business architect and me resulted into the improving case study to associate some of the project goals with the goals of this research.

6.2. Research approach

For the improving case study the action research approach is used. Action research is used more often in collaborative modeling research as it requires modelers (researchers) to make an active intervention in a group of participants, to study the effect (Morton, Ackermann, & Belton, 2003). As modeling requires highly advanced skills, it is difficult to train others to make this intervention to observe this effect (Barjis, Kolfschoten, & Verbraeck, 2009).

The action research cycle from Baskerville & Wood-Harper (1996) is applied: diagnosing (1), action planning (2), action taking (3), evaluating (4), and specifying learning (5). In the diagnosing stage, the key problems that require the enterprise to change or improve are identified. In the action planning step, the intervention is designed. The researchers and business process owners envision an approach to change the situation. In the action taking step, the actual intervention is made. The evaluation reflects whether the change had the desired and theoretically predicted effect by assessing the process in an expert panel review. Finally the business process owners and researchers reflect on what have been learned from the intervention.

The underlying subparagraphs discuss the research approach in detail, following the structure of the action research cycle.

6.2.1. Participants of the collaborative session

The collaborative modeling approach has been applied for conducting the modeling of the construction of the informational layer of Alpha was carried out as follows.

In total eight half day sessions has been organized in the period between February 9th, 2010 and May 12th, 2010 with the following participating roles:

- Two senior operational business architects with extensive knowledge of the operations and daily routines in the enterprise.
- One senior technical solution architect with extensive knowledge of the IT environment and tools and techniques that are used at Alpha in system development.
- Modeler – that role was fulfilled by the researcher.
- Process coach – the business owner and team manager of the operational business architects.
- Facilitator – the business owner and team manager of the operation business architects.

The role of recorder and gatekeeper were fulfilled by the modeler. All together there were maximum five participants during the sessions.

6.2.2. The action research cycle

This paragraph discusses all the steps in the action research cycle. In the diagnosing part the scope and boundaries of the case study are discussed, including the mutually benefits Alpha aimed at. During the action planning phase the case study's approach has been elaborated, including the corresponding time schedule and request for resources. In the action taking phase the case study is executed. The case study execution took three iterations and every cycle basically followed the action research cycle as well. All three iterations are illustrated in this sub paragraph. Due to time and resource constraints there has been agreed on the maximum of three iterations. In the evaluation phase of the action research cycle the case study's results are evaluated by conducting an expert panel session. Finally, the learning phase mentions the lessons learned from the case study.

6.2.2.1. DIAGNOSING

In the current way of working, Alpha needed an approach to translate the abstract business models of DEMO into a more specific representation of what a system developer must create. This representation must be aligned with the business purposes and therefore should not be created from a technology perspective. A presentation was given by a business architect of company Alpha. In the three hour session the context of the project was explained and the fit with the case study of the thesis was explored. The objective of the case study from the research point of view is to improve the conceptual model from Figure 28 in finding an optimum in service granularity following a formalized methodology. The resulting product will be an improved conceptual model which is applicable for designing an informational service model that is optimum for the organization, and therefore aligned with the business purposes. The match was found when the purpose of the research for the thesis was explained. The case study must identify an optimum service model, which is directive for system developers and still aligned with the business purposes.

After the presentation of the project by the business architect the first essentials of the case study were explained. The research approach was discussed and new appointments were made.

6.2.2.2. ACTION PLANNING

The research process has been explained in a presentation in which addressed the background, the problem statement, the main research question of the research project, and the conceptual model for informational analysis. See Figure 28. The point of contact within Alpha organized a meeting room that included all presentation facilities, like a beamer, a whiteboard, and computers. In total five participants joined the presentation. All participants are within the same department of Alpha and are addressed as business architect. The presentation took about one and a half hour and started at one o'clock in the afternoon. Besides the already mentioned topics, the presentation also captured an explanation of the service granularity topic. The granularity framework of Steghuis (2006) was explained. The notion of service, the informational layer of the organization, the methodology, the case study approach, and the proposed schedule was addressed as well. Especially the granularity framework resulted into in-depth discussions. Some of the discussion topics were about functionality of services, flexibility of business processes, and service reusability. The participants were discussing about the relevance of the topics for Alpha and how the service granularity framework aspects must be interpreted. Apparently all the aspects of the framework sounded familiar to the participants, which mean that the topic of service granularity also exists for the business architects of Alpha.

After the presentation and a coffee break the group continued in explaining the project of the case study. This was planned because not all participants of the first presentation were familiar with the project. The project was explained in the new way of working and the new concepts of DEMO version 3. Because some differences exist between DEMO version 2 and version 3 the group needed to agree on the version of DEMO that will be used during the case study. The group agreed on using DEMO version 2 for the case study because of the support by Xemod[5] for modeling. Another argument was the understandability of the differences between the Process Structure Diagram and Transaction Pattern Diagram in DEMO version 3 compared to the Process Structure Diagram in DEMO version 2. Most of the participants of the group are not educated in DEMO version 3 and therefore DEMO version 2 has been chosen. This agreement will not influence the results of the case study because of the focus on information objects. The information objects are linked to the arrows in the Process Structure Diagram, which represents the direction of execution of the coordination acts. Basically the arrow represents the agenda or coordination fact for the actor. Therefore the essence of the information analysis for the basic communication pattern will not change when the presentation format of the Process Structure Model changes.

Regarding modeling the informational layer the following has been agreed upon. Despite the proposed way of modeling to connect the business layer with the informational layer of the organization by De Jong & Dietz (2010), the case study uses a slightly different way of modeling. De Jong & Dietz (2010) describes the possibility of a performa actor to call an information transaction directly, without shaping the performa actor into an informa actor. Figure 30 expresses the possibility as described by De Jong & Dietz (2010). Just a part of the total picture is presented here.

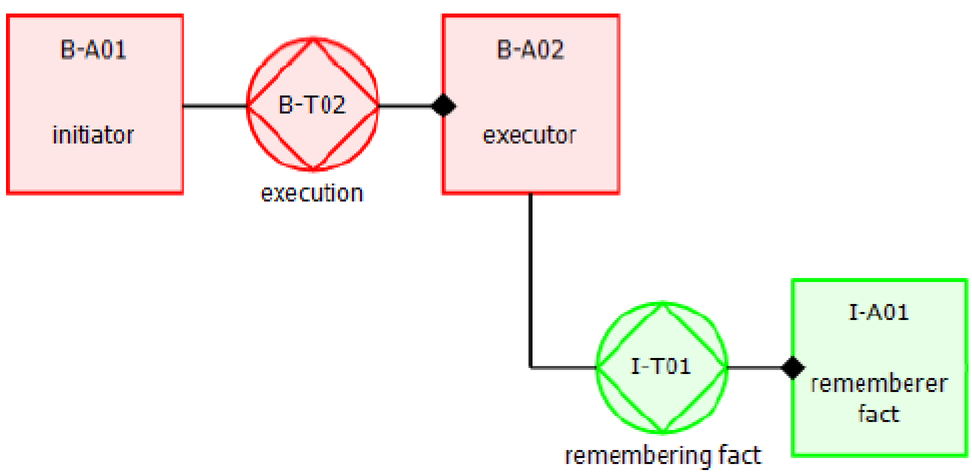


Figure 30: Performa invokes infological transaction (De Jong & Dietz, 2010)

This way of modeling suggests that performa actor "B-A02" starts an informational transaction with informa actor "I-A01". In this way it is not clear that the performa actor and the informa actor are the same,

[5] Xemod is a software package of Xprise. See www.xprise.com for more information.

but shaped from the ontological layer into the infological layer. Therefore this way of modeling is not precise and not based on the formal semantics of DEMO. The distinction axiom, as discussed by Dietz (2006a), says that in order to have a coordination act between two actors A and P successfully performed, actor P must expose his commitment in a performative act, addressed to actor A, and that actor A has to evoke in her the corresponding commitment to respond adequately. This act is part of the performative exchange between actor P and actor A. According to the theory of Habermas (1981), which is grounded in the methodology of DEMO, Dietz (2006a) says that the only way for actor P to expose its commitment and to make it knowable to actor A, is to express it through its informa ability, followed by the inducement in the mind of actor A of an equivalent thought, by means of its informa ability. Actor P does so in an informative exchange with actor A. Such an exchange consists of informative acts, which aim at achieving the intellectual understanding of the coordination act between actor P and actor A (Dietz, 2006a). The distinction axiom of DEMO states that intellectual understanding can only be reached by actors in its informa ability. Therefore it is not possible for an actor in its performa ability to start an informative exchange directly.

The way of modeling within the case study is explained by an example that is expressed in Figure 31.

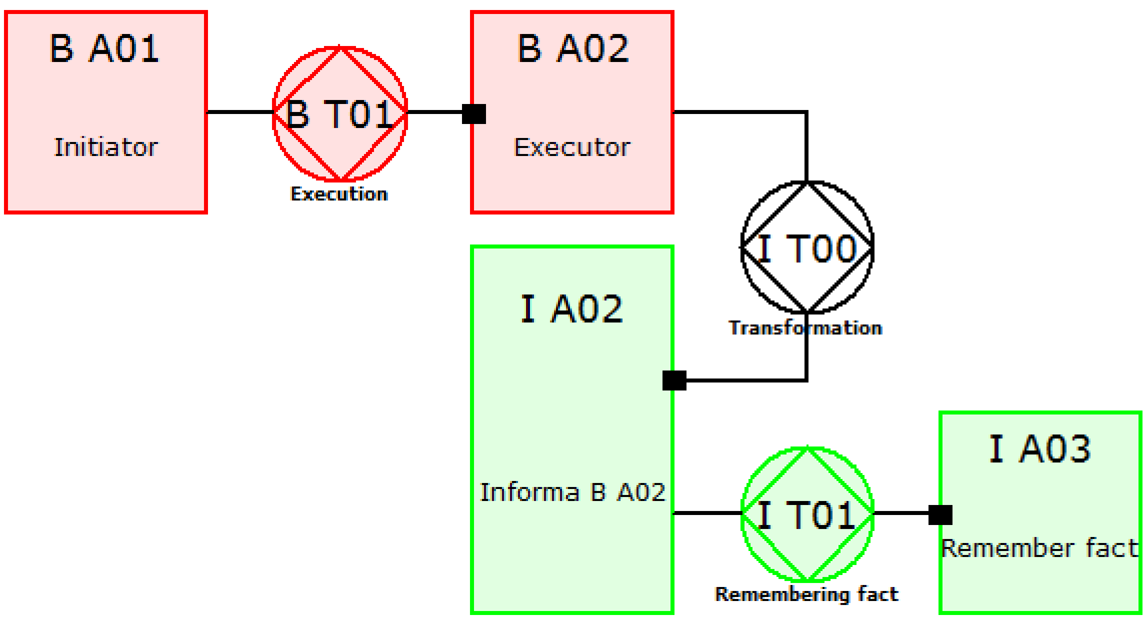


Figure 31: Performa transforming into informa

The figure shows the same example as in Figure 30 of De Jong & Dietz (2010), but it expresses the shaping of actor “A02” from its performa ability (“B A02”) into its informa ability (“I A02”). Informative transaction “I T00” exists for the purpose to request for information from the informa ability of actor “A02” by the performa ability of actor “A02”. This way of modeling separates the construction of the B-organization form the construction of the I-organization. The transaction “I T00” shows the connection between the construction of the B-organization and the function of the I-organization. The case study adopts the way of modeling from Figure 31, which makes transaction “I T00” obsolete, because it is embedded in the way of modeling.

Regarding the action taking process we agreed about that the case study will be executed at the desks of the business architects. The rooms where the business architects are situated all have white boards and internet connections available. Also a round table is available for group discussions. Besides the mentioned benefits the business architects also have direct access to supporting information sources in case questions pop up that can only be answered by requesting additional information sources, like documents or calling colleagues.

Both the presentations about the case study’s objectives and the agreements made about modeling provided a sound foundation to start the informational analysis.

6.2.2.3. ACTION TAKING

Based on a description of business processes we picked one main business actor in the organization. The business actor was modeled as a composite actor in the business model, which was already expressed in DEMO. See appendix C for the business model of Alpha at the start of the case study. The selected composite actor fulfills a central role in the organization of Alpha. Therefore we all agreed to model this composite actor because of its variety and central role in the organization, and at the same time define the scope of the case study. We selected composite actor “CA700 - SC”.

The following of this paragraph illustrates the three iterations that have been performed at Alpha. The iterations followed the action research approach as well. Therefore the following sub paragraphs are divided by the phases of the action research cycle.

6.2.2.3.1. First iteration

Diagnosing

The improving case studies goal was to get an approach that translates the abstract business models of DEMO into a more specific representation of what a system developer must create. This representation must be aligned with the business purposes and therefore should not be created from a technology perspective. The applied methodology from the exploring case studies was a fit to meet the first objectives of Alpha.

Action planning

The objective of the first iteration was to apply the conceptual model in Figure 28 on page 81. In doing so the researcher guaranteed that the infological model is created following the methodological approach of the explorative case studies, which has been described in paragraph 5.2.2. Therefore the researcher explained the conceptual model to the participants of the improving case study and I created the units of work that must be performed. The units of work were the information analysis phase and the creation of the infological model. The information analysis phase was guided by the concepts of conceptual model, which are; actor based analysis, gathering c-act requirements, and the classification of c-acts. The creation of the information model was guided by the concepts; shaping actors, information transactions, I-actors for production banks, and composite information actors for transaction banks.

The results of the first action taking phase must be evaluated and concluded to improve the conceptual model. The improved conceptual model will be utilized in the following iterations.

Action taking

The business architect, who was also my point of contact, started the information analysis for the case study. During the infological analysis sessions there was a lot of discussion. It appeared that the organizational construction model was not complete yet. While speaking about the required information it became clear that some interstrictions were not modeled, that boundaries of the department of consideration were not well structured, some transactions were missing, and some ontological defined actors and transactions appeared to be fully infological. The discussion was very fruitful. All participants agreed on the proposed changes and everybody was fully cooperative. The meaning of interstrictions was illustrated in paragraph 4.4.1.

Before we continued with the informational analysis we agreed first to make a new version of the organization construction diagram (OCD) and process structure diagram (PSD) of actor “CA700”. These models would help us verify the scope of the case study and would be the foundation of the whole case study. The senior technical solution architect requested to make a state diagram (SD) as well. This model shares a common understanding of the structure of information objects. The OCD, the PSD, and the SD of actor CA700 in its environment are illustrated in appendix D.

The researcher provided the participants the template for the information analysis. The template has been created with respect to the infological analysis that has been discussed in paragraph 5.2.2. This way of modeling made sure that the results of the infological analysis could be compared with the explorative case studies, because of the shared underlying structure. The used template is presented in appendix E. Directly after starting the infological analysis the researcher noticed that the modeling method is not self explaining. The operational business architect was not familiar with this way of modeling. Therefore the researcher explained the infological analysis method again in other words, supported by more examples, and the researcher provided him with some hints in the direction of information analysis, like using the Transaction Result Table and the interstriction arrows of the Organizational Construction Diagram (see paragraph 4.4.1). For confidentially reasons the Transaction Result Table is not included in this thesis.

Under supervision of the researcher we created the infological model. Because the participants, excluding the researcher, were not familiar with the way of modeling the infological model, the modeler created a first version of the infological model. The created infological model is presented in appendix F.

Evaluation

The first impression of the senior technical solution architect was "*the model seems like an unstructured mess, which is not useful for making software designs. The things that are missing are the transitions in correlated information flows*". The technical solution architect did find the state diagram very helpful to define the data structures. This reaction was understandable. When someone is looking from the viewpoint of technology, the state of the data is important. This means how the data is organized in a database structure. The impressions of the senior operational business architects on the information model were "*complex*", and "*all the benefits of the ontological model are withdrawn in the infological model*", and "*Do not understand what it means*". It became clear that the infological model is not easy to interpret, even not when the participants are DEMO professionals. One of the participants concluded that the characteristics and rules of the ontological model are also valid in the infological model. This participant could look through the complexity of the model and pointed at the strong aspects of it. Because the infological model follows the principles of DEMO, the same results can be realized between actors, transactions, responsibilities etc. The only thing is that the infological model is quite large compared to the ontological model. This substantiates the expression of one of the participants. The complexity is partly caused by the relative old modeling tool that we used. The used version of the tool does not support bridging functionality when two lines cross each other. Due to this lack of functionality, it does not become clear to the participants which relations are valid between an actor and transaction(s). Due to the large amount of informative exchange, which follows the results from the extended information use table, results into huge amount of relationships between informas and transactions in the infological model.

The infological model shows the functional model of information. It defines which informative transactions occur between actors, who share the same intellectual understanding. The informative actors also apply to the generic socionom patterns, like the performative actors do. After some discussion the infological model became clear for the participants and a basic approach has been agreed upon how optimize the functional infological model.

While we were discussing the infological analysis approach and the infological model we concluded the following.

1. The term "situational information", which is part of the template the researcher provided, is not clear. The term is not self exploratory and the operational business architects encountered difficulties to interpret the term with a respectful meaning.

2. When information must be provided to an informa that exists out of the scope, and the information is provided by an automated system, then a transaction that provides the information is created for the purpose of external links. We decided for this way of modeling to support the interstriction in the OCD. Interstriction visualizes an information requirement between an actor and the transaction that contains certain production facts. The information transaction that is embedded in the scope is therefore always of a type that gathers production facts that are required outside of the scope.
3. The utilization of information actors is considered a proper way of modeling, because there exist an informa who is primarily responsible for the information that is required by the performa.
4. The composite actors that are responsible for retrieving information from and storing information in transaction banks must be aligned with the state diagram instead of a one-to-one mapping from the organizational construction diagram. Storing and retrieving information functions as a gateway to the data logical layer of the organization. Therefore the informational layer should take notice of the data logical structures and must transform the information in such a way that it can be stored or retrieved. This means that one composite informa per business transaction for retrieving or storing production facts is not efficient and does not reflect the actual gateways to the data logical layer.

To elaborate on the description of the situational information column, De Jong (2008) says that it can be derived from the action model of DEMO. Regarding to Dietz (2006a) the action model of an organization consists of a set of action rules. An action rule serves as a guideline for an actor in dealing with its agenda and influences the actor's acts. This means that situational information is synonym for action rules.

Dietz (2006a) mentions the similarity with business rules, but for the sake of ambiguity he would not intertwine these concepts. He says that action rules are like procedures that must be executed to achieve a particular result, but that business rules are not having a uniform definition. In order to replace the term action rules by business rules, this thesis adopts the business rules definition of Steinke & Nikolette (2003), which is very similar with the description of action rules:

A business rule is a statement that aims to influence or guide behavior and information in an organization.

Business rules occur in different structures. Wagner (2005) distinguishes five structures. These structures are:

- Integrity rules express constraints.
- Derivation rules express conditions that result in conclusions.
- Reaction rules, also known as Event-Condition-Action rules, alternative-action rules, or post-conditions, specify a trigger that activates the evaluation of the rule, a condition that is evaluated, and a subsequent condition is met.
- Production rules, also known as condition, action rules, are similar to reaction rules, but do not specify a particular circumstance in which the evaluation takes place.
- Transformation rules restrict the state changes of objects.

Regarding the description of action rules, the adopted definition of business rules in this research also is about influencing acts in an organization. Looking at the formal description of the action model language in Dietz (2009) the action rules follow the syntax of reaction rules, meaning that an event is tested (the agendum) on a certain condition that results in some action. Due to the similarity of action rules and the reaction rule structure of business rules, the term situational information in Table 6 on page 76 is replaced by the term business rules. Hoogervorst (2009) also says that coordination activities of the transactions are guided by business rules. This results in an adjusted table layout, which is provided below in Table 8. The table carries the extended information use table (eIUT) label. Herewith it refers to the Information Use Table (IUT), which is a cross model between the Process Model and the State model. The IUT shows which elements of the State model are used in performing each of the steps in the Process Model. The eIUT goes beyond just referring to object types and fact types, but addresses the guidelines and utilization of the object types and fact types in performing each agendum in the Process Model.

Table 8: *Extended Information Use Table*

Agenda	Act	Business Rules	Operational information	Data to process

To get an understanding of the utilization of the eIUT a small example will be discussed now. Consider the ontological model and its process model in Figure 32. The example is based on the library case in Dietz (2006a). Besides the two models the result type for transaction B-T01 is: *R01 – membership M has been started*.

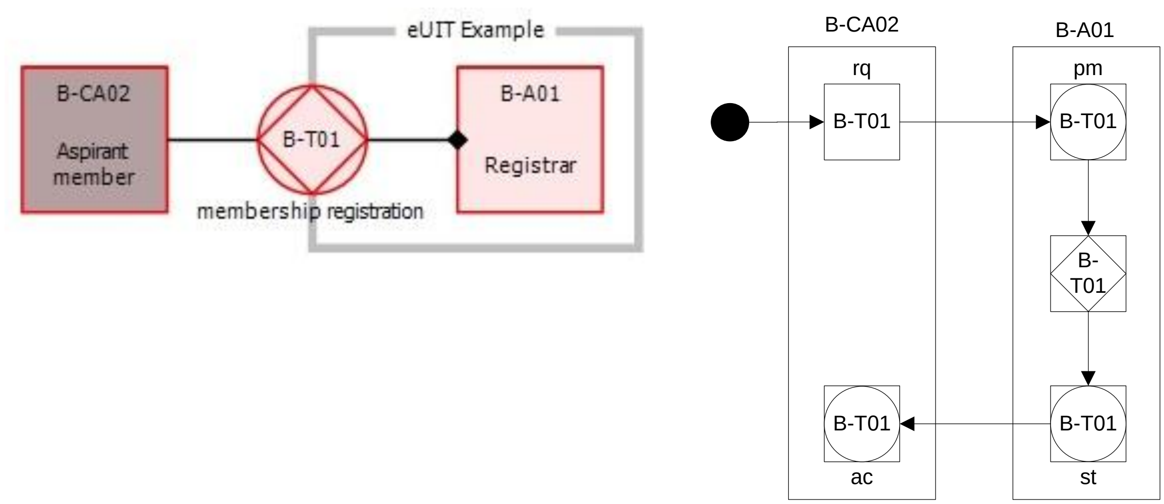


Figure 32: Organization Construction Diagram and Process Structure Diagram

To start the information analysis the eIUT is created for all elementary actors within the boundary of the organization. Elementary actors are subject of analysis, because those actors process information within the scope of the system under consideration. The information requirements of external composite actors are analyzed separately.

Derived from the process model the eIUT can be created. Table 9 shows the initial table setup. The label of the table is eIUT – B-A01 Registrar, referencing to the elementary actor B-A01.

Table 9: *eIUT – B-A01 Registrar*

Agenda	Act	Business Rules	Operational information	Data to process
b-t01/rq	Validate the registration application of the aspirant member.			b-t01/pm
b-t01/pm	Apply registration of the aspirant member.			b-t01 b-t01/st

To continue with the information analysis, the business rules and the operational information must be gathered. Interviews with representatives of the business, which follows, can be structured with help of the eIUT. Because all the activities of the process model are already known, the analyst can ask directive questions about the process. One should note that not the process model structure only is directive for interviewing business representatives. Take the transaction result table and the interstrictions from the Organizational Construction Diagram also into account. How to structure interviews and how the eIUT can be helpful in that process is beyond the scope of this thesis. More information about structuring interviews can be found in Silverman (2005).

For the sake of the example the business rules and the operational information are partly copied from Dietz (2006a) and are fictitiously made up. The complete result of the information analysis process is shown in Table 10.

Table 10: *eIUT – B-A01 Registrar (complete information analysis)*

Agenda	Act	Business Rules	Operational information	Data to process
b-t01/rq	Validate the registration application of the aspirant member.	Age >= 18? Valid legal identification? Annual fee known?		b-t01/pm
b-t01/pm	Apply registration of the aspirant member.		Surname First name Middle initials City of residence Street name House number	b-t01 b-t01/st

Agenda	Act	Business Rules	Operational information	Data to process
			Postal code Gender Date of birth Starting date of membership Annual fee	

As one can see the business rules apply only for the “promise“ C-act (see the “Data to process column”) and the operational information apply only for the “execution” and “state” C-acts. This is just a coincidence and not a rule. In case multiple processes are connected asynchronously with each other, then some business rules will be valid at the execution step.

The four results that were concluded impacts the conceptual model in Figure 28 on page 81. The information analysis must split the C-act requirements into specifying business rules for C-acts separately from specifying operational information for P-acts. This split explicitly shows the utilization of business rules in communication and the utilization of information in production. The information model also specifies information transactions for external links. This supports interstriction between system scopes. Further the information model introduces the information manager concept. The information manager is an informa that functions like an information gateway for the performa who is shaped into its informa ability. Therefore an information manager informa is linked one-to-one with a performa who is shaped in its informa ability. Finally, the information model recognizes the clustering of transaction banks instead of defining a composite informa per transaction bank. The clustering implies the transformation of information into a format that can be accepted by data logical gateways.

The improved conceptual model is shown in Figure 33.

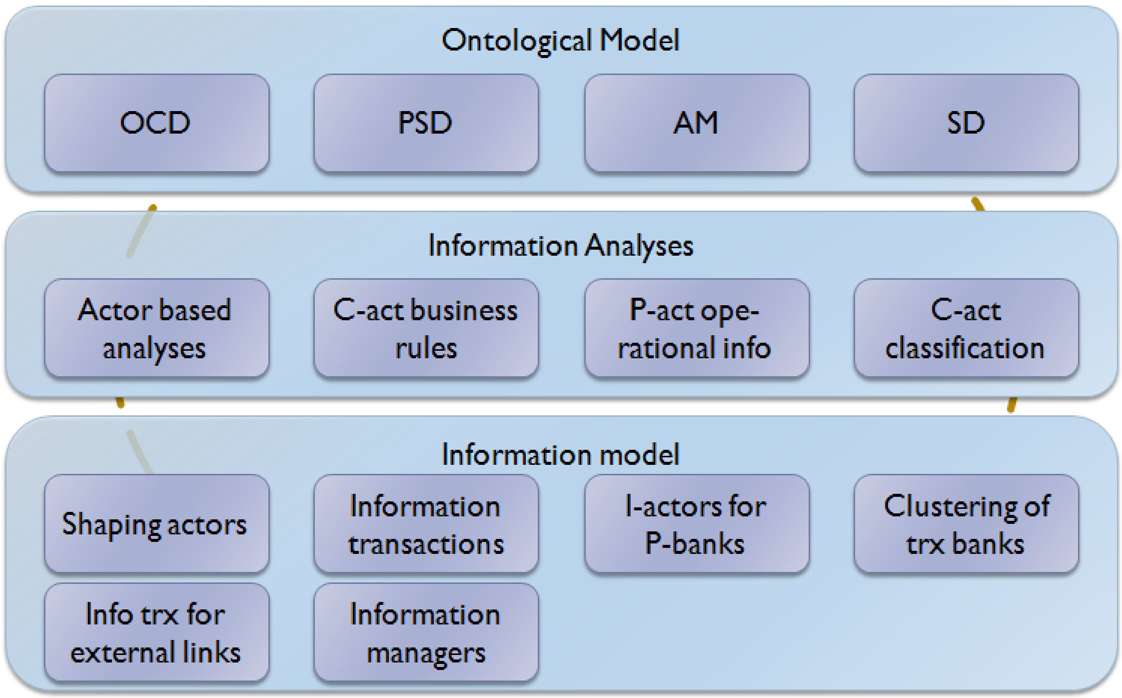


Figure 33: Improved conceptual model after the first iteration

Learning

Below we mention the key lessons learned of the first iteration from the joint sessions with Alpha.

- The sessions showed that information analysis, starting from an OCD, requires in depth knowledge of the business transactions and the business processes. The workshop approach is good and well structured. The only thing that has to be remembered is the time spent on a subject. Some workshops took about 4 (four) hours and resulted in discussing 6 transactions and some rework to be done.
- The sessions proved that optimizing the information model is fruitful in a workshop setting. There are more requirements to be evaluated besides following the proposed methodology to shape from the performa level into the informa level.
- The state model is of great importance in defining the information model and also the system to system integration must be clear to identify the proper integration transactions.
- The information transactions are still defined at a high level. This results into an unclear definition of the unit of functionality that an information transaction contains.

6.2.2.3.2. Second iteration

Diagnosing

The evaluation phase and the learning phase from the first iteration leaves room for further improvement of the conceptual model. Despite that the information model expresses the information requirements per performa, the information model is still too abstract for a system developer. On the other hand, the information transactions are strictly guided by the information analysis without considering granularity.

Action planning

The objective of this cycle is to analyze the created information model from a technological perspective and a granularity perspective. This way of looking at the information model is similar to the middle-out strategy as discussed in paragraph 3.1.4. Just like the first iteration we conducted several workshops in which we looked at the information transactions from a different angle. This perspective visualized other aspects of transactions that have not been seen before. In this phase it is possible to reconsider the information model from different angles and without a strict sequence, because the reference architecture of the information model is a stable factor. The reference architecture is the OCD, which has been thoroughly discussed in the first iteration.

In order to support the middle-out strategy we evaluated the service granularity determinants on the information model. In accordance with the explorative case studies we have seen that the determinants of service granularity can be evaluated against DEMO transactions. When discussing the transactions we looked at the technological side of the transaction. There was a large contribution projected for the senior technical solution architect in this iteration.

Action taking

The action taking phase started with explaining the service granularity determinants. The presentation that has been performed was a brief recap from the presentation that was held during the action planning phase at the beginning of the improving case study at Alpha.

While performing the workshops the participants came to the conclusion that the current information model is not meeting the aspects of granularity yet, and it is not directive in software development for a technical architect as well.

To structure the discussion about service granularity we made a grid of the infological transactions from Figure 42 (see appendix F on page 144) and the service granularity determinants. Based on the theoretical explanation of the determinants of service granularity (see paragraph 2.2.2 and appendix A) and relying on the fundamental experience in business and information technology of the participants the grid has been established. The grid is illustrated in appendix G.

In parallel with creating the grid we started to discuss about the influences that granularity determinants imply on software development. In this discussion we followed the business services granularity scheme, as has been illustrated in 3.1.5. All the transactions were evaluated and the conclusions and findings of these discussions are presented in appendix H.

The discussion about service granularity took several sessions. This concludes as well that the topic is complex and amendable to various opinions. In the time between the sessions we adjusted the models based on our findings and performed further desk research on raised questions that could not be answered at that time. This iterative way of working with DEMO, which has been presented in Dietz (2009), has been proved in practice by us. The discussions lead to changes in all the models that we have created before. This meant that the OCD, the PSD, the SM, and the infological construction model changed.

In the joint session with the senior operational business architects, the senior technical solution architect and the other participating roles, we redesigned, discussed and validated the OCD, the PSD, the SM and the infological construction model. This thesis presents the latest versions of the models only. All the intermediate versions that were created in between of our sessions are not relevant for the objective of this thesis and are therefore not included. The mentioned most recent models are found in appendix I.

Evaluation

One of the findings during creation of the service granularity grid was the lack of clarity in the functionality that one information transaction captures. Guided by the ability of deriving use cases from business transactions, as described in Dietz (2003), we collaboratively agreed upon to define kite-level use case per information transaction. We decided to apply use cases because of the common understanding of the methodology and the state of the methodology to be a well known standard in information system analysis. The decision of defining kite-level use cases has been made upon the common frame of reference of the senior business architects and the senior technical solution architect. Both roles share knowledge about kite-level use cases. For the business architect, he or she needs to specify the function of the use case at a high level. And for the technical solution architect, this level is already directive for system development without blocking the creativity in the craftsmanship of the information technology specialists. The technical solution architect can correlate with the kite-level use cases and specify the use cases in-depth by creating sea-level or fish-level use cases. The description of the kite-level use cases are centered on the production act and production fact, which the information transaction represents. The levels of use cases have been illustrated in paragraph 3.2.2.

While we were discussing the unit of functionality that must be captured by a use case, we automatically discussed other service granularity aspects. The unit of functionality also triggered the discussion of the complexity of implementing the transaction, the ability to reuse the functionality that is provided by the transaction and the conceived flexibility in the business process when the transaction is implemented.

The template that we used to define kite-level use cases is an adjusted version of the template provided in Fowler (2008). The template is illustrated in appendix J.

Validation of the granularity aspects on transactions that contain information facts about captured information from production banks always shows the same behavior. We concluded to alter the way of modeling ontological production banks. A production bank reference on ontological level is required for information purposes. Because no actor within the boundary of the model is responsible for the transaction that feeds the production banks, this means that a production bank can be used for information purposes only. An actor refers to a production bank because the actor needs that kind of information while executing its responsibility. At infological level the information from a production bank is provided by an information provider actor who reads the production bank. This way of modeling is directly in an optimum form considering the service granularity aspects context independence, functionality, genericity, and reusability. This means that when a basis CRUD[6] (create, read, update, delete) is applied to the production bank provider, the transaction must always be implemented as a read transaction.

Another change in the basic infological model was made after discussing the composability of several production banks. In the ontological model three different production banks are mentioned, represented as infological transactions TI08, TI09, and TI10, see Figure 42 on page 146. From the use case description we learned that basically the information that must be provided from these banks is generic. Therefore it has been concluded, substantiated by the service granularity aspects of composability and genericity, that the information provider actors could be composed into a single information provider actor in the infological model. The way of using the information from the production banks is the same in all cases.

Driven by the reusability aspect of service granularity, the non-ownership of the production bank facts, and the state model, we decided to explicitly model the production banks outside of the infological boundary and connect the production bank provider actor to it by a transaction. These infological transactions are basically composite transactions that are combining several transactions into one that belongs to different informa actors. Some other transactions are defined to share information between systems. In this way an information system is defined following the DEMO system definition in Dietz (2006a) and De Jong & Dietz (2010). These kinds of transactions are identified only by knowing the

[6] CRUD are four basic functions of persistent storage. An external production bank is considered to be persistent storage. For more information about CRUD see: http://en.wikipedia.org/wiki/Create,_read,_update_and_delete

interstriction rules of an actor in another system. These transactions are modeled onto the boundary of the infological model. This way of modeling explicitly shows the infological transaction that must be supported by the system, and shows that an actor must be responsible for providing that web service within the infological model. In this way the interstriction at the ontological level between an actor and a production banks is represented in the infological model as a transaction to request the production bank provider to call the infological transaction from the external production bank. This way of modeling shows the utilization of information banks outside of the informational model. It becomes clear what kind of interfaces must be realized in interaction with third party information transactions.

Based on the state model we concluded that the composite infological actors that are responsible for the gateway to the datalogical data stores were not matching in the infological model. The reason why the model changed is substantiated by reusability of services and the links that must be supported between these data stores. The number of data stores to be used effectively has been taken from the state model. This resulted in less data stores to be modeled.

The four results that were concluded impacts the conceptual model in Figure 33 on page 98. The information analysis recognized the contribution of use cases to use as a frame of reference when discussing the service granularity aspects. Besides, the use cases contributed in the mutual understanding between operational business architects and technical solution architects. The information model also captures the validation of service granularity aspects. Explicitly validating the service granularity aspects opens up valuable discussions between the stakeholders. The third and the forth adjustment are about composing transactions or splitting services. Basically this functionality is an intrinsic part of DEMO that is elaborated in paragraph 4.3.3 while discussion the composition axiom.

Within the second iteration the benefit of clustering information transaction banks became applicable. Due to the combination of the state model, the information model and a senior technical solution architect we could optimize the utilization of transaction fact storage.

The improved conceptual model is shown in Figure 34.

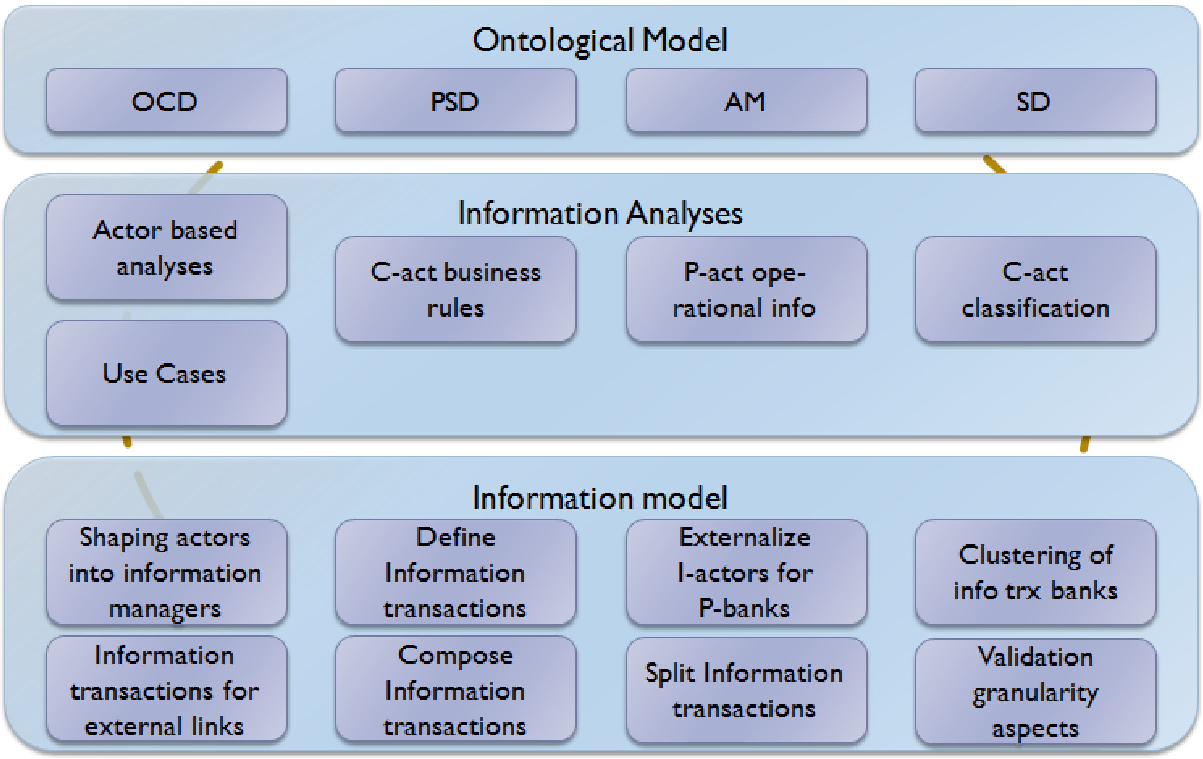


Figure 34: Improved conceptual model after the second iteration

Learning

Below we mention the key lessons learned of the second iteration from the joint sessions with Alpha.

- The granularity discussion revealed the way of modeling the DEMO infological model from different angles. To make the grid of infological transactions and service granularity more concrete we gave a functional personal interpretation to the grid by identifying kite-level use cases and we gave a technical personal interpretation to the grid by referencing to the state model
- By our opinion the gap of abstract modeling at the organizational level and concrete modeling at the systems level is shortened when use cases are defined that specify the unit of functionality that an information transaction contains.
- Kite-level use cases provides basic context of the transaction, which is helpful for the technical solution architect to continue from. During the discussion about service granularity aspects and the kite-level use cases, we frequently updated the basic infological model into a new version. This updated version then was the basis for discussion in the next round. This iterative way of modeling results in an optimized infological model that still is correlated with the ontological layer, but already leveled with service granularity aspects and therefore optimized in information usage.

- Just like the first iteration, during the second iteration it was also said several times that the infological model is visually complex and therefore complicated to implement. The participants wanted to know how a system would look like when the infological model is implemented. It is required to keep the infological model structured and less complex. Another aspect that must be considered before implementing the infological model is to get support among stakeholders.

6.2.2.3.3. Third iteration

Diagnosing

The evaluation phase and the learning phase from the second iteration leaves room for further improvement of the conceptual model. Although the second iteration gave more interpretation how to implement the infological model, there was still lack of understanding in the used vocabulary. The DEMO infological taxonomy did not match the interpretation of the taxonomy that is used in systems development. Another aspect that was still unclear was the purpose of the information managers. It was not clear how to implement this special kind of informa's.

Action planning

The objective of this cycle is to analyze the information model that is modeled during the second iteration, from a complementary perspective. This must result into a common understanding of the informational model. On the one hand the operational business architects must be able to correlate the information model to the business ontology, and on the other hand the technical solution architect must have a directive view on how to support the information model from within a system.

In order to test the feasibility of the objectives we arranged some more collaborative workshops. The goal of these workshops was to discuss about the information model from the second iteration and searching for a common taxonomy that is supported in literature as well.

Action taking

In the action taking phase of the third iteration we discussed the objectives of this iteration. After some discussion it became clear that the information transactions in the infological model could be interpreted as information services. This change in the taxonomy is substantiated by literature in Terlouw & Albani (2010). See paragraph 2.2.1 and 2.2.3. We collectively decided to translate information transactions into information services.

After changing the information transactions into information services we continued to discuss about the purpose of the information managers that are part of the infological model. We discussed this topic because it was not clear for the technical solution architect how to implement such kind of services. He said *"It looks like a personalized information portal in which an actor has to consume all the information instead of the information needed for his or hers current business operation"* and *"This does not seem to contribute to the flexibility in business processes granularity aspect."* With help of the manager of the operational business architects we came to the conclusion that providing information is not the correct responsibility. The responsibility to gather the required information actually lies with the business actor. The business actor must have the ability to shape in an informa actor to capture the information required for his or hers business operation. The manager of the operational business architect introduced transformations for this kind of behavior. A transformation represents the shaping of a business actor into its informa ability. In the former models of first iteration and the second iteration (see Figure 33 and Figure 34) this was expressed by a regular Information transaction and an information manager actor, but in the current way of modeling we changed it in a black transaction with a TF prefix. TF stands for transformation. The transformations are connecting the business actors outside of the boundary with information actors inside the boundary. An information actor only is responsible for one information object. Only those objects that are somehow related to each other are within the responsibility of an information actor. This means that a business actor can shape into one or more information actors. Based on the aspect of context independency the decision has been made what information requests must be supported by one information actor. Only information requests that are required in supporting a business actor are covered in one information actor. If the required information is utilized several times from different contexts, then the information actor is separately modeled. The transformation objects are linked to actors by a dashed arrow. A dashed arrow means an information linkage, like the interstriction linkage in the organizational construction diagram or the dashed lines in an actor bank diagram. This actor bank diagram shows that in fact all relations between actors by transactions are information links, but the solid line covers the dashed line in an OCD. See Dietz (2006a) page 206-207 for more information. The adjusted infological model is illustrated in appendix K.

While we were discussing the newly created transformations we concluded that the n..m relationship between business actors (performa's) and informa actors to shape into is visually complex. All the information links are crossings each other, which does not make the model clear. In order to structure the transformations one operational business architect came with the idea to create an authorization matrix for actor transformations. This authorization matrix shows all the business actors and the information actors. The resulting grid is an overview of business actors who can shape into an information actor. Whenever one cell in the grid is colored, then the corresponding business actor is able to shape into the information actor. The authorization matrix is illustrated in appendix L.

Evaluation

Once more the collaborative modeling sessions resulted in meeting the predefined objectives. Because of the collaboration between the operational business architects, the technical solution architect, the manager of the operational business architects and the modeler with his supporting theoretical foundation we were able to provide more clearance in interpreting the infological model.

We concluded that there is a better understanding among the participants when we are speaking in terms of services instead of transactions. Although DEMO transactions are not sound services, the terms are overlapping in purpose. In order to express our conclusion on using the term service, we adjusted the conceptual model from the second iteration. The term transaction has been replaced by the term service in the information model area of the conceptual model.

Due to the remarks that the actors who are representing the role of information manager were not understand from a technical point of view, we eliminated these kind of actors on a sound foundation. The arguments why we decided to eliminate the information managers have been discussed in the action taking phase already. To support the shaping ability of performa's into informa we introduced transformations. The transformations give a better expression of the shaping activity that takes place. In order to express the importance of transformations we added the transformations to the conceptual model and removed the shaping activity from the information model area.

By a form of reverse modeling we created an authorization matrix for actor transformations. Afterward we concluded that it would have been better if we created the authorization matrix before modeling the information model. The sequence of first analyzing which actor must transform into an informa is more related to the process of information analyses. Therefore the conceptual model is adjusted in this area as well with the activity to create an authorization matrix. Because of the iterative nature of this research and the creation of a new conceptual model we could not foresee this activity beforehand.

The improved conceptual model is shown in Figure 35.

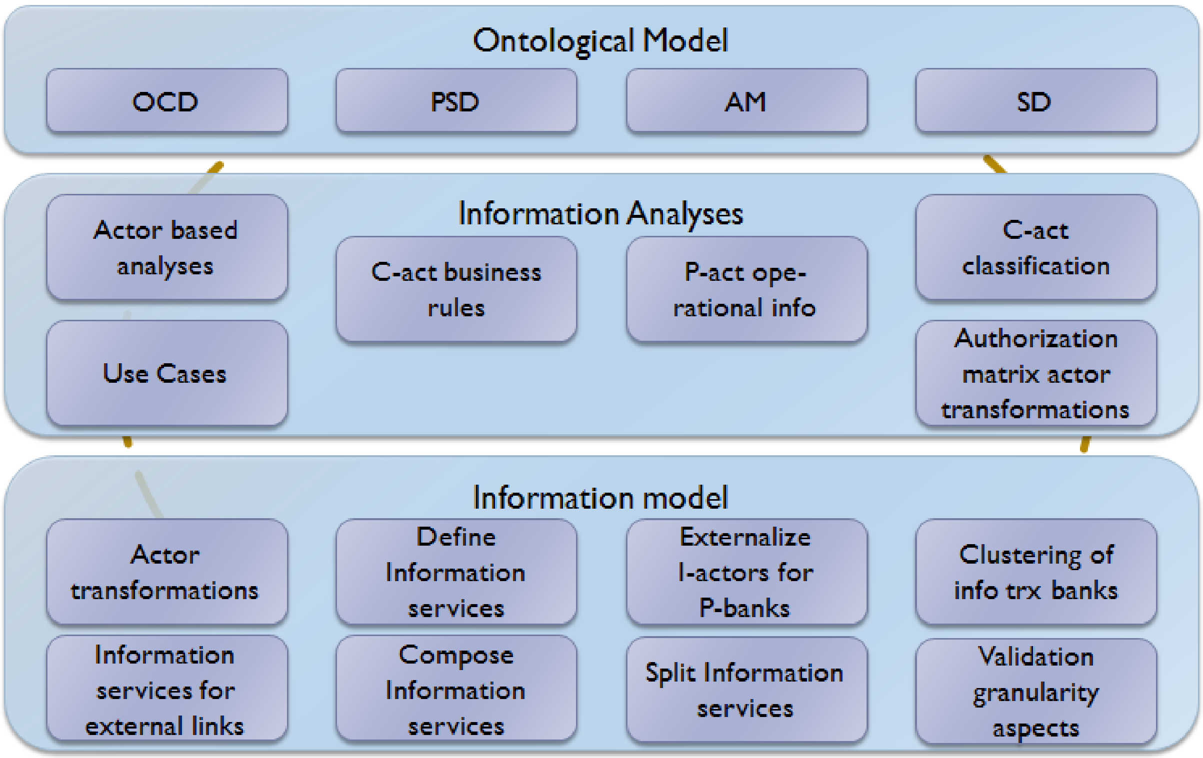


Figure 35: Improved conceptual model after the third iteration

Learning

Below we mention the key lessons learned of the third iteration from the joint sessions with Alpha.

- Searching for a common semantic understanding of the terms used in the process of information system design is prerequisite for success. Collaborative modeling, where several specialists with a different frame of reference participate, contributes to achieve that kind of common semantic understanding.
- Validation of the information model from different angles leads to a model that is understood by all participants and functions like a frame of reference for follow up activities.

6.2.2.4. EVALUATION

The action taking phase has been an intensive period in which a great enhancement has been made to the conceptual model that originated from the explorative case studies. The multi disciplinary team that participated in the collaborative discussion and modeling sessions of the improving case study has proven to be successful. Thanks to the contribution of the participants we created a conceptual model that originates in DEMO and guides the information analyses and information modeling towards an information model that is directly translated from the organizational business layer and is directive for system developers.

The decision to specify information usage based on business rules and operational information was contributing the information analyses process. Because we adjusted the formal DEMO method a little, we were able to clearly specify the information requirements based on the process structure diagram. The resulting extended Information Use Tables (eUIT) were the starting point of the information modeling phase.

The discussion about service granularity opened doors to look at the information model from different angles. The several service granularity aspects triggered the participants to discuss about the unit of functionality, reusability, complexity, flexibility, genericity etc. One effect of this discussion was the creation of kite-level use cases to specify the unit of functionality in more detail. The kite-level use cases were perceived to lower the level of abstractness, which provided the technical solution architect a better direction of the functionality that must be supported by technology. Another effect was the clustering of information actors that provided the gateway to the datalogical layer. We modeled the information actors in such a way that the information storage or retrieval was optimized to the underlying data structures. Again another effect was the introduction of integration services. By defining integration services it became clear how the information model operates within its environment. The integration services identified the interfaces that must be realized to capture information from outside the system or provide information to other systems.

The transformation ability of business actors (performa's) to shape into information actors (informa's) provided a different view on the ability of business actors to shape in their informa ability. The transformations showed that there exists a n..m relationship between performa's and informa's. To structure the complex relationship we developed an authorization matrix for actor transformations. The authorization matrix visualizes the complex transformations that actors can undergo.

During the action taking phase, which lasted from February 2010 until May 2010, we also struggled with the interpretation of the information model. Sometimes it was difficult for the participants to follow the process because of their little experience in information modeling using DEMO. When that happened we stopped our research and did some recap on DEMO, service granularity or the applied research method, depending on the need of that moment. With collaborative modeling it is important that all participants have complete knowledge about the process and all have access to the same information to reach the most valuable contribution of all participants.

At the end of the action taking phase we all concluded that we conducted in an innovative case study with an ambitious objective to find an optimum in service granularity. All the participants agreed that the final conceptual model is helpful in reaching an optimum, but it will always be the joined craftsmanship of the specialist which makes it a success.

6.2.2.5. LEARNING

Below we mention the key lessons learned from the joint sessions with Alpha.

- The scope of the system under consideration must be clear. We started our improving case study with already created ontological models of Alpha. These models were too large to analyze within the available amount of time. Therefore we started scoping the ontological model into a level that was contributing to business objectives as well as to perform a challenging scientific case study.
- The participants indicated that the modeling effort helped them to understand the resulting model. Seeing a model being built is easier to understand then trying to understand a completed model presented by an expert. Based on the resulting conceptual models the participants could identify gaps and improve the conceptual model, while keeping a clear and recognizable picture that captured shared understanding among the participants.
- The modeling language should be explained using an example. The participants were not familiar with the particular modeling language, and explaining it during the actual modeling effort could result in misunderstanding. To avoid this, it is useful to create a small example model, of 3-5 elements and relations. In this way the modeling language can be explained.
- Completeness and correctness of the model is more likely when the business owners are involved in the actual modeling effort. In confronting different perspectives of the organization and the architects, tradeoffs and different views are integrated or resolved. Further, when the model is created with all relevant stakeholders, completeness is more likely.

6.3. Summary

This chapter illustrated the improving case study that has been conducted at Alpha during the winter and spring of 2010. The objective of the improving case study was to test and improve the conceptual model that has been developed during explorative case studies in the past. Following the action research approach of Baskerville & Wood-Harper (1996) the conceptual model is tested and improved in three iterations. This approach, in which the result of the preceding phase is input for the next phase, is also known as the spiral towards understanding (Carroll & Swatman, 2000).

The first iteration introduced the concepts of business rules and operational information as part of the information analysis process. To structure those information requirements we used the extended Information Use Table (eUIT). Supported by the information requirements we created the first informational model. That model showed the information transactions reflecting the information requirements. The information transactions were requested by performa's who were shaped into their informa ability. Besides modeling the infological model we had to adjust the ontological model and the state model as well. These changes were required after several in depth discussions among the participants of the collaborative sessions.

The second iteration was about the service granularity determinants. Discussing service granularity resulted in looking at the infological model from different angles. Guided by the determinants the participants were discussing about unit of functionality, flexibility, reusability, complexity and so on. Due to the multi disciplinary team the granularity aspects were discussed from a business perspective as well as from a technological perspective. To get grip on the abstractness of information transactions we concluded to define kite-level use cases per information transaction. This provided a better understanding of the offered amount of functionality that an information transaction represents.

The third iteration concluded in using transformations to represent performa's who shaped into their informa ability. Due to the n..m relationship between performa's and informa's we showed that every performa can shape into the desired informa whenever the performa needs the information to process its operational business activity. On the other hand, every informa can be requested by several performa's. This means that the information that is provided by an informa can be reused by several performa's. To structure the complexity of this kind of relationship we introduced an authorization matrix for actor transformations. This grid provides a clear overview of the transformations that exist. The necessity of such an overview was driven by the fact that the infological model was complex to interpret due to the mix of crossing transformation lines.

The resulting conceptual model after iteration three is perceived to be helpful in reaching an information model that expresses an optimum in service granularity. The participants also concluded that it will always be the joined craftsmanship of the specialist which makes it a success.

7. VALIDATION

This chapter discusses the validation process of the conceptual model for finding an optimum in service granularity using DEMO's informational construction modeling, since the improving case study has lead to an adjusted conceptual model. In paragraph 7.1 the reason for validating the conceptual model including some theoretical foundation is given. We decided to validate the conceptual model by an expert panel. In paragraph 7.2 the planning phase preceding the expert panel session is illustrated. Performing the actual expert panel session is discussed in paragraph 7.3. The results of the validation session are presented in paragraph 7.4 and the underlying conclusions are discussed in paragraph 7.5.

7.1. Diagnosing

In order to mark the conceptual model with scientific relevance and make a contribution to the body of knowledge, the conceptual model that has been created during the collaborative research period must be validated. To test whether the results of the conducted improving case study are contributing to answer the main research question the researcher asked experts who are seen as authorities in the area of DEMO and SOA from a practitioner's point of view as well as from an academic point of view. In order to gain the maximum benefit of the validation process the researcher aimed for a joint session with experts and participants of the improving case study to discuss several propositions that reflect the conducted research process and research results.

Creating a joint session with experts and practitioners to validate the research process by discussing the propositions combines all the experiences.

The technique to discuss the propositions is similar with the focus group method. Bryman & Bell (2007) defines a focus group method as *"… a form of group interview in which: there are several participants (in addition to the moderator / facilitator); there is an emphasis in the questioning on a particular fairly tightly defined topic; and the accent is upon interaction within the group and the joint construction of meaning. As such, the focus group contains elements of two methods: the group interview, in which several people discuss a number of topics; and what has been called a focused interview, in which interviewees are selected because they 'are known to have been involved in a particular situation' and are asked about that involvement. The focused interview may be administered to individuals or to groups. Thus, the focus group method appends to the focused interview the element of interaction within groups as an area of interest and is more focused than the group interview."*

The benefit of conducting the focus group method is the ability to have the participants probe each other's reasons for holding a certain view. Especially because the improving case study reflects a specific process execution that has been chosen to follow, it is interesting to discuss the participant's opinion about the conducted case study. Another projected benefit of the focus group method by Bryman & Bell (2007) is the challenging part among the interviewees. The response to a proposition could trigger another interviewee, who shares a different opinion. Challenging each other's views generally results into consensus after a while by sharing a consistent common understanding.

A Group Decision Support System (GDSS) is utilized that facilitates the focused group method. A GDSS is designed to improve the efficiency and effectiveness of (distributed) group work by offering a variety of tools to assist the group in the structuring of activities, generating ideas, and improving group communications (Hengst, Adkins, Keeken, & Lim, 2005). By using a GDSS that is controlled by a computer the researcher could assess the propositions, facilitate the discussion and validate the opinions of every participant by asking them to make a decision that fits their opinion about the proposition best.

7.2. Action planning

In preparation of the GDSS session the researcher invited experts and practitioners in the field. The experts and the practitioners were invited by e-mail. The sent invitation is illustrated in appendix M. In total 8 (eight) specialists participated in the GDSS, of whom were three experts and five practitioners.

The experts who participated in the GDSS session were:

- Prof. Dr. ir. Jan Dietz (emeritus professor, Delft University of Technology)
- Ir. Joop de Jong (emeritus professor Extended Enterprise Studies, HU University of Applied Sciences, Utrecht)
- Ir. Linda Terlouw (Enterprise architect and Ph.D. student of Prof. Dr. ir. Jan Dietz)

For confidentially reasons the names of the practitioners of Alpha who participated in the GDSS cannot be specified. The roles of the practitioners who participated the GDSS session were:

- Two senior operational business architects
- Two senior technical solution architects
- One manager of the operational business architects department

The facilitators were René Wiersma (the researcher) for the content part and Prof. Dr. ing. J.B.F. Mulder MBA (HU University of Applied Sciences, Faculty Science & Engineering) for the facilitating part.

To conduct the GDSS session we arranged a meeting room with a beamer to give an introducing presentation and to project the propositions during the GDSS. Further the meeting room was equipped with a wireless network. The wireless network was used to connect the laptops of the participants to the GDSS host computer. In order to capture the GDSS discussions we digitally recorded the GDSS with a video camera. The set-up used for the GDSS is visualized in appendix N.

Aligned with the research questions and the conducted improving case study the researcher formulated 32 (thirty two) propositions that were split in 5 (five) main categories. The 5 (five) main categories were covering the following areas:

- Service granularity;
- The process of information analyses;
- The activities that are part of the conceptual model in Figure 35 on page 107;
- The contribution of DEMO in finding an optimum in service granularity;
- Expert panel discussion to challenge the participants in giving their opinion.

The formulated propositions per main category are listed in appendix O. All propositions are formulated in an open fashion to challenge the participants to give their opinion. During the GDSS the participants were asked to score their opinion on a Likert scale and optionally substantiate their opinion by giving extra explanation in a free form text field. The used Likert scale is as follows:

- Strongly agree (SA) = 5
- Agree (A) = 4
- Undecided (U) = 3
- Disagree (D) = 2
- Strongly disagree (SD) = 1
- Abstinence (AB) = 0

7.3. Action taking

The GDSS took place on May 12th, 2010 at Alpha. The GDSS started at one o'clock in the afternoon and lasted for about two hours and fifteen minutes. The session started with a presentation about the conducted improving case study. After a short introduction the basic assumptions that count for the improving case study were explained. These assumptions were the utilization of DEMO and the DEMO informational construction model, the research of De Jong & Dietz (2010) and the available OCD that Alpha already modelled. Furthermore the researcher explained the research approach and the collaborative modelling approach we conducted.

After the introduction in the presentation, the conceptual model that was created from the explorative case studies and the three iterations that we conducted during the improving case study was introduced. Per iteration were the changes explained compared with the conceptual model of the previous iteration and why those adjustments were made. This structure guided the participants through the improving case studies process and made them familiar with it. After explaining the final iteration the GDSS started.

Before we started with the GDSS the facilitator Hans Mulder checked if the supporting GDSS systems were working. After everybody's approval we started with the GDSS. The process went as follows. The facilitator presented the propositions on the projection screen. All the propositions for the first main category were shown at once. This gave me the opportunity to read the propositions out loud and give a little extra explanation about its purpose. When the propositions were explaining the researcher asked if the formulated propositions were clear. In case one or more propositions were not clear, then the researcher explained the proposition again in other words or by using examples. When everybody agreed that the propositions were clear, we started using the GDSS. All participants had to give their opinion about the propositions on a Likert scale. They had to choose the value on the Likert scale that meets their opinion at best. In case the participant felt like giving some extra information why he or she chose that value, then the participant could use the functionality to add additional information. This functionality was supported by the GDSS. When all participants had provided their opinion about the propositions, then the facilitator took over the system and presented the results. The results were presented visually in bar graphs. The graphs were showing the average rating, meaning the average choice that was provided based on the Likert scale, and the graph shows the variability, meaning the spread or variance among the answers. In case the variability was more than 40%, then the facilitator asked the participants about their opinions to trigger the discussion in case there were none or hardly any remarks given during the GDSS session. The just described process continued for all main categories.

7.4. Evaluation

Directly after the moment that the opinions of the participants of the GDSS were given the facilitator started the analyzing process, which is embedded in the GDSS. Per main category we discussed the findings in case the spread of the opinions was too large or the results of the ratings attracted attention. The findings of the GDSS are elaborated in appendix P and summarized in paragraph 7.5.

7.5. Learning

The objective of the validation of the improving case study was to validate whether the research results, the concepts on which the results are anchored and the process to achieve those results were shared among experts and practitioners. The main research question was how to define the right granularity for services such that the determinants of granularity are balanced with the proposed business benefits of flexibility and reusability using DEMO's informational construction modeling from a service-oriented perspective. The process of the improving case study has tested the several components of the research question. We challenged the granularity of services and by collaborative modeling we tried to find an optimum that follows the provided definition of optimum in paragraph 3.3. The improving case study showed that the right granularity of services is not achieved out of the box. We have seen that optimizing service granularity is a process of several iterations and that reaching an optimum is something subjective. The participants of the improving case study agreed that the end result is an optimum model of information services, because we achieved consensus about the results that could be substantiated whenever necessary and is directive for the technical solution architect to further design software on.

Granularity is an important concept in the research question. In accordance with many literature sources that are already mentioned before, the participants of the GDSS also agreed that service granularity is an important topic in information system design in every organization. Without exception, all the participants conclude that service granularity is a critical success factor in information system development. The participants of the GDSS concluded that service granularity is a universal topic in information system design, independent the type of organization. The participants recognized the benefit of an optimized service granularity in order to gain reusability in information objects or reduce the level of complexity, even if complexity is reduced by the utilization of information hiding via composition of services. An optimized granularity of services can also contribute to the flexibility of business processes whenever an information transaction reflects its granularity. If granularity is optimized in such a form, then the information model is just as flexible as the business. Flexibility is bound by the business process itself and the necessity of information. When the construction of the business changes the supporting information model must be adjusted as well.

Just like the conclusion that service granularity is considered to be a universal critical success factor, the participants of the GDSS also concluded that an optimum in service granularity can be standardized per branch when DEMO is applied. One participant explained that "within a right ontology, the information services are all the same. Independent of the type of organization". By applying DEMO the essential activities of the organization are by default in an optimized fashion. Although every organization has its own face, identity and strategic goals, when the construction of the organization is analyzed there is a lot in common among organizations. This underlines the probability that an optimum in service granularity can be standardized per branch.

One note that has been stated several times during the validation session is the complexity of service granularity in combination with the infological model of DEMO. It was hard to understand the infological model and the service granularity viewpoint that has been created made it even harder for the participants. This means that whenever service granularity is approached with DEMO, the participants of the collaborative modeling sessions must have a thorough knowledge of DEMO, of the infological layer and its way of modeling, and of service granularity.

To be able to answer the question of how to define the right granularity of services we validated the process of the improving case study. In optimizing the granularity of services the information analyses process must be directive in defining the coarseness of those services. This means that the information analyses process is responsible for gaining the benefits of an optimum service granularity. The participants recognized the directive nature of the informational analyses in information system development. A top-down strategy is considered to be a good approach where the business remains involved during the process. Collaborative modeling, where business competences and IT competences are united in the informational analyses process, was concluded to be successful. Fusing competences minimizes the gap that is perceived between business and IT.

Although collaborative modeling results in better optimized information models, the quality of the realized software remains dependent on the skill level of the software developer. One participant of the GDSS completed "one can still create awkward software, even in an environment where service granularity is optimized". This underlines the craftsmanship of the software developer in the process of implementation.

The iterative nature of the process, the top-down strategy, the activities that are part of the informational analyses phase and the activities that are part of the construction of the information model were all validated by the conceptual model. See Figure 35 on page 107. 69% of the participants agreed with the defined activities and they all recognized the contribution of the activities in relationship with the objective of the improving case study. The other 31% did not have an opinion about the conceptual model because they found themselves unfamiliar with the completed process of the improving case study.

The experts and practitioners of the GDSS collectively recommended a change in the conceptual model. The specific purpose of the activities “Compose information services” and “Split information services” in relation with the activity “Define information services” was not unambiguously defined. The latter two must be considered as a way of modeling information services within the activity of defining them. The recommendation of the participants of the GDSS resulted in an improved version of the conceptual model. See Figure 36 on page 117.

The last part of the main research question to be validated is the methodology. Based on the research question the improving case study applied DEMO. The participants of the GDSS agreed that the DEMO infological construction model can be seen as an information services model. Considering the concepts of services in paragraph 2.2.1 we conclude that the DEMO infological construction model adopts the service-oriented concepts. The objective in modeling the DEMO infological model was to create a model that is linked to the business organization but is also directive for services implementation. The participants of the GDSS did not support the proposition that the DEMO infological construction model is directive for implementation. In contrary they stated that the ontological foundation of DEMO abstracts from implementation. The ontological model describes the essence of the organization independent of the organizational layer one is modeling. However, the participants agreed on DEMO being applicable to model services in a fashion that it is optimized on service granularity in all organizations. It even makes reusability of services explicit by modeling. The participants share the opinion that the DEMO infological model is not suitable for an implementation in software. Apparently the DEMO infological model does not contain enough information to be directive for software development. This makes the infological model a logical model of the construction of information services.

Summarizing the conclusions of the GDSS is that DEMO is applicable in modeling the essence of the informational organization. The DEMO infological construction model supports the modeling of services in a fashion that services could be optimized to their environment. By extending the information services of DEMO with the activities that are stated in the conceptual model, the participants concluded that a coherent and relevant set of activities is provided that contributes to optimizing service granularity.

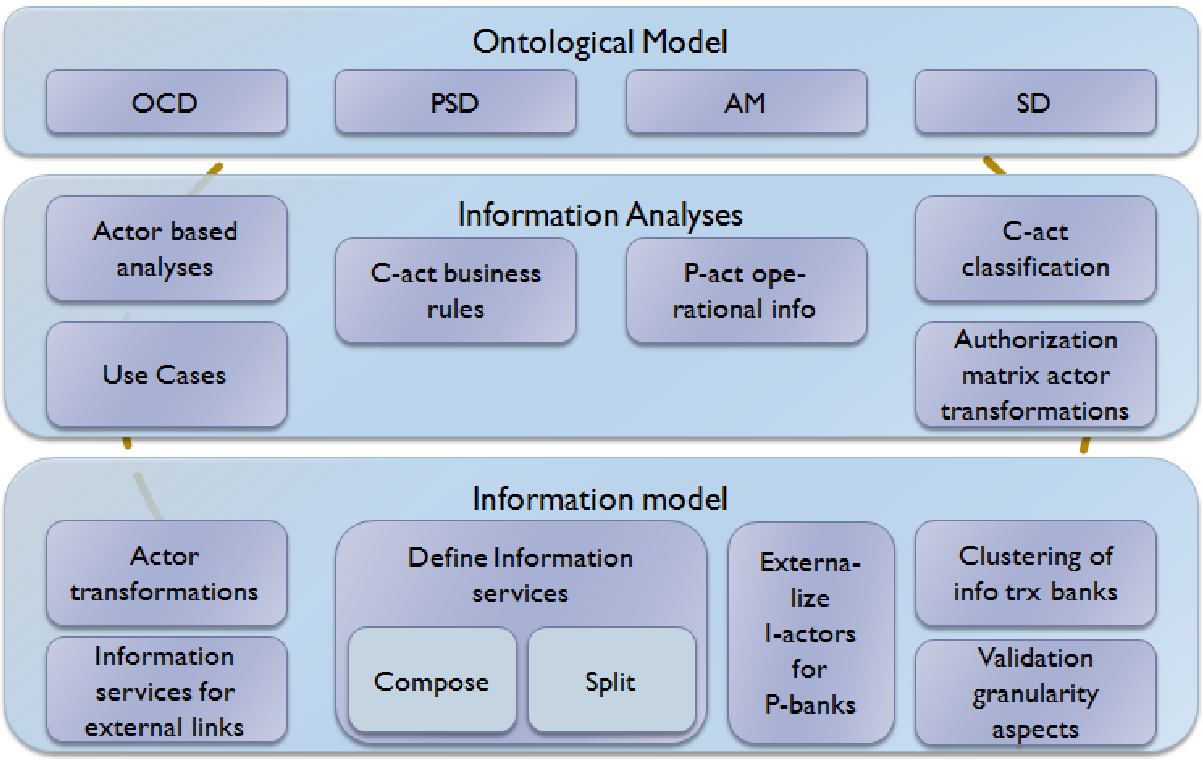


Figure 36: Improved conceptual model after the GDSS

8. RESEARCH CONCLUSIONS AND RECOMMENDATIONS

This chapter concludes the research. In order to answer the main research question this chapter firsts recalls the sub questions in paragraph 8.1. By answering the sub questions on the basis of findings derived from this research the way is paved for answering the main research question. At last recommendations are given for future research.

8.1. Answers to research questions

This paragraph recalls the sub questions and answers them accordingly. The sequential order of the sub questions is equal to the sequential order when presented in paragraph 1.2.

1. What are the main business drivers for service-oriented architectures?

The literature study conducted for this research project (see paragraph 2.1) shows that embedding a SOA in organizations is mostly driven from two pressures on IT departments. The first pressure is caused by the heterogeneous environment. All applications, systems and other IT solutions must interact together to support the business processes. This IT landscape with a variety of components is difficult to maintain without a standardized integration protocol. This means that IT departments must standardize how they implement systems integration and prevent other solutions. For example, the IT department standardizes the usage of a canonical data model to be the integration format and prohibits the utilization of application specific formats. The second pressure is caused by the constantly changing business. The IT environment must be adaptive and flexible to support the required changes.

SOA is able to release those pressures. The various definitions of SOA all point at the main advantages: flexibility in business processes and reusability of services within a heterogeneous environment. For example with SOA services can be orchestrated into a composition that supports the execution of a business process. The orchestration is possible due to a standardized protocol in systems integration. When the business process changes, it probably will be sufficient to adjust the orchestration of services. An easy to adjust orchestration benefits to the perception of flexibility in supporting business process. When the granularity of services is optimized it is most likely that the services can be reused in multiple orchestrations to support several business processes.

Instead of creating the most flexible and reusable SOA, one has to consider the main business strategy. The SOA must be supportive to the business strategy and not become a goal on its own.

2. What are the concepts and determinants of service granularity?

Literature research in paragraph 2.2.2 learned that most authors agree on the main determinant of granularity, which is the unit of modularity of services. Granularity is like an umbrella over several determinants. Previous research concluded the determinants that all together create the term granularity. The determinants are listed in Table 3 on page 31. Not all determinants are relevant in a particular situation. To distinguish between the determinants they are divided according to the three layers from the organizational theorem. The organizational theorem has been illustrated in paragraph 4.3.5. Depending on the organizational layer, determinants for granularity shift in relevance. In basis there exist two types of granularity of services. Fine grained services typically implement a single atomic operation and exchange limited amounts of data. Coarse grained services implement high-level business functions. A third type, called enterprise services refers to the ability of services compositions. Coarse grained services can be composed of more fine grained services. The composition of fine grained services into larger grained services can facilitate reuse and flexibility in orchestrating business processes. For example the cash withdrawal service, which a bank could provide by an ATM or a cashier behind a desk, can be considered as a coarse grained service. This service could be a composition of more fine grained services like checking the balance of the account, verifying the bank account number and personal access code, and withdrawal of an amount of the account.

3. What methods are available to define service granularity?

Literature research and the expert panel of the GDSS session concluded that there is no comprehensive design method for service granularity (see respectively chapter 3 and appendix P). In order to define service granularity one has to define the services first. There are several techniques for service identification known in the information technology science. The available techniques can be converted into one of the four following strategies.

1. Top-down: is an "analysis" first approach;
2. Bottom-up: is a progressive process of building services or assembling existing technologies to provide business solutions;
3. Meet-in-the-middle: advocates a service identification strategy using both a top-down and bottom-up approach, applied in an iterative fashion;
4. Middle-out: produces both higher-level business and information architecture and design artifacts, and working and deployed services.

Every strategy has its own strengths and weaknesses. From a business point of view the top-down strategy would be preferable to the other strategies. But applying strictly one strategy is usually not practical. This means that for example a top-down approach and a middle-out approach could be combined to benefit the strengths of both strategies in the service identification process.

When services are identified, then the right level of granularity needs to be determined. Based on literature research in paragraph 3.1.5 three levels of granularity are provided.

1. Distributed services: are the most fine-grained services and mostly implemented by a specific platform component model;
2. Business services: are coarse-grained services, providing information about a certain business object. In the service classification of Steghuis (2006), these services belong to the type of information system services;

3. Application services: are compositions of business services, providing a total solution for an essential business activity.

One should strive to find an optimum in business services (information system services, Steghuis (2006)), since that is the ideal unit of asset management and has the largest potential for software re-use. This has been the objective of the improving case study.

4. What models are used to express service granularity?

With respect to the main research question this research provided an overview of six different analysis techniques to express service granularity in. These models are chosen because of their applicability to follow a top-down strategy, their well known foundation in the information technology science, their underlying formal semantic model or their massive utilization in today's IT projects. The models are:

- Design & Engineering Methodology for Organizations (DEMO) (See paragraph 3.2.1)
- UML Use Cases (See paragraph 3.2.2)
- UML Activity Diagrams (See paragraph 3.2.3)
- Petri nets (See paragraph 3.2.4)
- Business Process Modelling Notation (BPMN) (See paragraph 3.2.5)
- Business Process Execution Language for Web Services (BPEL4WS) (See paragraph 3.2.6)

The six different models have been assessed. The assessment is based on theoretical desk research, which has been summarized in Table 4 on page 50. The criteria that were assessed are listed below.

- Top-down approach.
- Formal semantics.
- Coherent, this means that the technique constitutes a logical and truly integral whole.
- Service-oriented.
- Objective, this means that the technique is based on real facts and not influenced by personal beliefs or feelings.
- Granularity, this means that the technique is directive in one or more service granularity determinants.

This research applied DEMO because it is an eligible methodology that enables defining a service model in a structured way that is directly linked to the essence of the organization without mentioning anything about process orchestration on the level of the informational construction model. The DEMO Infological construction model enables to construct an information service model that encapsulates determinants of service granularity within the defined transactions.

5. What is a right service granularity?

The expert panel participants were not unequivocal in defining right in a right service granularity (see appendix P). Experts and practitioners explained a right granularity like “granularity that is attached to the B-organization is by default right, according to the applied methodology.” Another point of view was “choosing granularity based on software design and software maintenance principles”. Some experts referred to the iterative nature of the informational analyses process, which means that a right granularity can never be reached because there are always influencing changes within or outside the organization. One expert refers to an aligned unit of functionality that supports orchestration of services in a sense to be supportive to the business objectives and also some expert panel participants were not able to give a definition of a right granularity.

According to the participants of the improving case study an optimum model of information services is the end result which is agreed upon. In their opinion right was the achieved consensus about the results that could be substantiated whenever necessary and is directive for the technical solution architect to further design software.

From all these statements by experts, practitioners and participants can be concluded that finding a right service granularity is a process that is influenced by subjective behavior.

Literature research did not result in a clear definition as well. Only the Cambridge University Press dictionary (2010) translates optimum in “most likely to bring success or advantage”. This means that it is not a certainty that success or advantage is achieved. This relative verb underlines the subjective nature of optimum.

Within this research the verb right in a right service granularity is defined in paragraph 3.3.

6. What is the DEMO informational construction model?

The DEMO informational construction model is the intermediate layer between the business layer and the data layer, from an organizational theorem point of view (see paragraph 5.2). This layer has data as input and information as output. The informational layer creates information that is used in business operations. Therefore the informational layer can be seen as the services layer on top of data that needs to be programmed by software engineers and are specified by business analysts. The informational layer is the layer that needs to be (partly) implemented in software systems. Therefore on the informational layer the issue for getting the service granularity right is the most visible.

This means that the DEMO informational construction model shows the construction of the information services. These information services are provided to business actors who can judge or make decisions about the situation. For example, an information service provides the amount of money that someone has on its account. Based on the information that has been provided by the information service, the business actor can decide if the person who asks for money withdrawal could or could not receive the requested amount. The underlying data to provide the information service are all the savings and withdrawals on the specific account. This concludes that the construction of the information service is to perform a certain calculation over all the requested data and provide the result of the calculation to the business actor.

During the explorative case studies we created several informational models. We have experienced that creating such a model is an iterative process. This is also recommended by the new way of working with DEMO version 3. The activities in the process are the informational analyses, the corresponding modelling and the reflection with the ontological construction models. This all concludes that the DEMO informational construction model is not just a model, but also the process to create the model.

8.2. Discussion

This paragraph discusses the central research question. All findings, conclusions, interpretations and consequences are listed. The findings are the actual findings of the analysis. The conclusions are the conclusions that can be drawn from the findings. The interpretation is the interpretation of the findings and the conclusion, which may provide additional reasoning or foundations for the conclusion. The consequences illustrate the effects. The central research question from paragraph 1.2 is recalled first.

How to define the right granularity for services such that the determinants of granularity are balanced with the proposed business benefits of flexibility and reusability using DEMO's informational construction modeling from a service-oriented perspective?

In the information technology science the granularity of services is considered to be an important, but difficult to answer, question. The first finding is about the process to identify services.

Finding	The extensive literature study of this research as well as the conclusions from the improving case study shows the importance of the granularity question and that it is a general problem in organizations that cannot be answered by a commercial-of-the-shelf solution.
Conclusion	The explorative case studies as well as the improving case study showed that finding an optimum granularity in services is an iterative process that is concentrated on informational analysis, modelling the results and evaluating the model in the context of its environment. Although the improving case study is performed in a collaborative fashion only, literature and experts agree that such an approach on average gains better results. The mix of business competences and technology competences united in one process challenges out of the box solutions by looking from different viewpoints. The applied top-down strategy, which is directed by the DEMO methodology, is considered to be a good

	approach in the process of information analyses by experts and literature. The mix of competences and the iterative nature of the process allowed us to apply the middle-out strategy as well, to look at the model from a technology perspective. An expert completed "*…It is good to look from an IT perspective to see what's handy for those guys*", referring to the applied middle-out strategy. According to literature DEMO is an eligible methodology to (re)design organizations in an integrated way. According to the expert panel DEMO is applicable in modeling the essence of the informational organization. The DEMO infological construction model supports the modeling of services in a fashion that services could be optimized to support the essentials of the business at best. The participants of the expert panel agreed that the DEMO infological construction model can be seen as an information services model that adopts the service-oriented concepts.
Interpretation	The translation of the essential business transactions into an informational construction model that optimally provides the information requirements is an iterative process that is concentrated around informational analysis. Optimizing the granularity of services in the informational construction model is related to the ontology of the business and requires several disciplines. These disciplines model the services from a business perspective following a top-down strategy and also from a systems perspective following the middle-out strategy.
Consequences	The iterative nature of the informational analysis process could result in a never ending cycle. This could lead to an over-analyzed informational construction model that could not be implemented. In project environments these cycles must be bounded to balance the quality of the informational construction model with the other project objectives.

The second finding is about defining a right granularity of services.

Finding	The objective of finding an optimum in service granularity became the achievement of consensus.
Conclusion	According to the participants of the improving case study an optimum model of information services is the end result which is agreed upon. In their opinion an optimum or right services model was the achieved consensus about the results that could be substantiated whenever necessary and is directive for the technical solution architect to further design software on. The consensus was grounded on two pillars. One pillar was the organizational support, which was expressed in the degree of the informational construction model providing the informational services to support the essential business activities. The second pillar addressed the practical implementation of services that was concentrated on technical aspects. These technical aspects were linked to the

	service granularity determinants. All the determinants were evaluated on the ability of implementing the services. For example the degree of complexity of a service or the degree of reusability.
Interpretation	When an optimized granularity of services is reached when consensus is achieved among the participants of the information analysis process, then there could be no methodology that finds an optimum in services granularity just by applying a methodology. It is a subjective interpretation of what skilled participants of the information analysis process conclude together. The benefit of a methodology turns into a guideline that can be followed that triggers the participants of the information analysis process to act in a certain way.
Consequences	Finding an optimum in service granularity will always be influenced by the craftsmanship of the analyst. This makes that optimizing service granularity is no mathematical process and remains a complex topic when designing services.

The third finding is about the utilization of DEMO in finding an optimum service granularity.

Finding	The participants of the expert panel did not support the proposition that the DEMO infological construction model is directive for implementation. In contrary they stated that the ontological foundation of DEMO abstracts from implementation.
Conclusion	The ontological model describes the essence of the organization independent of the organizational layer one is modeling. However, the participants agreed on DEMO being applicable to model services in a fashion that it is optimized on service granularity in all organizations. It even makes reusability of services explicit by modeling. The participants share the opinion that the DEMO infological model is not suitable for an implementation in software. Apparently the DEMO infological model does not contain enough information to be directive for software development. This makes the infological model a logical representation of the construction of information services. The participants also concluded that the way of modeling the infological model in DEMO is complex and even harder to understand when the infological model is used to express service granularity.
Interpretation	DEMO is a good starting point to identify information services when a top-down service identification strategy is followed. The informational construction model expresses precisely the information services that are required by the business. Although these information services are required it is not plausible to conclude that these services are already optimized in service granularity. According to the service granularity determinants technical aspects are an important part in optimizing service granularity, but not all considerations can be reflected in the DEMO informational construction

	model. This concludes that the DEMO informational construction model is not suitable for software systems implementations. Whenever service granularity is approached with DEMO, the analysts participating in the informational analyses process must have a thorough understanding of DEMO, of the infological layer and its way of modelling, and of service granularity.
Consequences	The DEMO informational construction model remains an abstract model. Although the applied methodology in this research did provide a guideline to develop services, it was not sufficient to develop the services without further detailed design. This means that the DEMO informational construction model must be completed with at least another software design oriented methodology. The combination of both completing methodologies creates the work packages that can be developed in software.

The fourth finding is about the business benefits of an optimized service granularity.

Finding	An optimized granularity of services contributes to the promises of SOA.
Conclusion	The expert panel recognized the benefit of an optimized service granularity in order to gain reusability in information objects or reduce the level of complexity, even if complexity is reduced by the utilization of information hiding via composition of services. An optimized granularity of services can also contribute to the flexibility of business process whenever an information transaction reflects its granularity. If granularity is optimized in such a form, then the information model is just as flexible as the business. Flexibility is bound by the business process itself and the necessity of information. When the construction of the business changes the supporting information model must be adjusted as well. This makes optimizing the determinants of granularity a continuously balancing game.
Interpretation	The link of the informational construction model with the ontological construction model ensures that the essence of the business is supported. This link, which is provided by applying DEMO, is in favourite of supporting flexibility. This means that when the construction of the organization changes the information organization changes as well. This does not automatically leads to redesigning the complete information construction model, but it could lead to changed information requirements. The reusability of information services is expressed in the DEMO informational construction model when more than one requesting actor request production of an information facts. This way of modelling directly expresses reusability.
Consequences	Although the promises of SOA can be expressed in the DEMO informational construction model, the model itself can quickly be too crowded. An overcrowded model reduces the

	applicability of the model.

The fifth finding is about the conceptual model that has been created in the improving case study.

Finding	The conceptual model provides activities that are relevant to define a right service granularity.
Conclusion	By extending the information services model in DEMO with the activities that are stated in the conceptual model (see Figure 36), the participants of the expert panel concluded that a coherent and relevant set of activities is provided that contributes to optimizing service granularity.
Interpretation	The conceptual model is founded on a coherent and comprehensive methodology and provides a guideline how service granularity of information services can be optimized.
Consequences	The conceptual model is the result of this research, but it has not been validated in other organizations or branches yet. To conclude if the conceptual model provides a guideline to optimize service granularity and expresses all the necessary activities it has to be tested in other organizations or branches first.

8.3. Recommendations for future research

This paragraph recommends for future research topics for students or practitioners. Future research is recommended on the validation of implementing the conceptual model in several organizations. An improved conceptual model that is validated in several organizations can contribute to the general acceptance of the conceptual model and being the first methodological approach towards finding an optimum in service granularity that is founded on a coherent and comprehensive methodology. According to the expert panel session the probability has been recognized that an optimum in service granularity can be found per branch. This probability must be validated by scientific research and applying the conceptual model in different organizations that operate within the same branch.

In organizations there is an urge for practical experience in implementing optimized information services models. Considering this urge a recommendation for future research is to make the step from concept to concrete, where the conceptual model of this research can be the frame of reference. This research provided the process and the activities that are recommended to apply when the DEMO infological model requires implementation. Although the infological model itself is abstracted from implementation, the combination with the provided additional activities is directive in information systems development. The student's research objective is to start from the frame of reference and improve the conceptual model with activities that contribute to a smooth implementation.

BIBLIOGRAPHY

Aalst, W. van der., & Hee, K. van. (2008). *Workflow Management – Modellen, Methoden en Systemen* (3rd ed.). Den Haag: Academic Service, Sdu Uitgevers.

Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., & Holley, K. (2008). SOMA: A method for developing service-oriented solutions. *IBM Systems Journal , 47* (3), pp. 377-396.

Barjis, J., Kolfschoten, G.L., & Verbraeck, A. (2009). Collaborative Enterprise Modeling. In E. Proper, F. Harmsen, & J. Dietz (Ed.), *First NAF Academy Working Conference on Pratice-Driven Research on Enterprise Transformation PRET 2009, held at CAiSE 2009* (pp. 50-62). Amsterdam: Springer.

Baskerville, R.L., & Wood-Harper, A.T. (1996). A Critical Perspective on Action Research as a Method for Information Systems Research. *Journal of Information Technology* (11), 235-246.

BEA, IBM, & Oracle. (2005). *Service Component Architecture: Building Systems using Service Oriented Architecture*. Retrieved May 8, 2010, from A joint white paper by BEA, IBM and Oracle: http://xml.coverpages.org/ni2005-12-07-a.html

Brook, P.E. (1986). *Wikipedia*. Retrieved January 2010, from V-model: http://nl.wikipedia.org/wiki/V-model

Brooks, F.P. (1986). No Silver Bullet - Essence and Accident in Software Engineering. *Proceedings of the IFIP Tenth World Computing Conference*, (pp. 1069-1076).

Brown, A., Delbaere, M., Eeles, P., Johnston, S., & Weaver, R. (2005). Realizing service-oriented solutions with the IBM Rational Software Development Platform. *IBM Systems Journal , 44* (4), pp. 727-752.

Bryman, A., & Bell, E. (2007). *Business Research Methods* (second edition ed.). Oxford University Press.

Bunge, M.A. (1979). Treatise on Basic Philosophy. *A World of Systems , 4*.

Burbeck, S. (2000). The TAO of E-Business services - The Evolutions of Web Applications into Service-Oriented Components with Web Services. *Tech.rep., IBM Software Group* .

Capgemini. (2006). *Capgemini.* Retrieved April 18, 2010, from Architects Capgemini: http://architectes.capgemini.com/communauteDesArchitectes/laMethodologieIAF/b_Architecture_and_the_Integrated_Architecture_Framework.pdf

Carroll, J., & Swatman, P. (2000). Structured-case: A methodological framework for building theory in information systems research. *European Journal of Information Systems* .

Cockburn, A. (Sept-Oct 1997 and Nov-Dec 1997). Structuring Use Cases with Goals. *Journal of Object-Oriented Programming* .

Combine. (2003). *Component-Based Interoperable Enterprise System Development - An EU Framework 5 Project.* The Open Group.

Cox, D., & Kreger, H. (2005). Management of the service oriented-architecture life cycle. *IBM systems Journal , 44* (4).

Daft, R.L. (2007). *Understanding the Theory and Design of Organizations.* Ohio, USA: Thomson South-Western.

De Jong, J. (2008). *Desiging the ontology of the I/D-organization of the enterprise.* Univeristy of Technology Delft. Delft: Not yet published.

De Jong, J. (2009). Integration Aspects between the B/I/D Organizations of the Enterprise. In A. Albani (Ed.), *CIAO! 2009 and EOMAS 2009, LNBIP 34* (pp. 187-200). Springer-Verlag Berlin Heidelberg.

De Jong, J., & Dietz, J.L.G. (2010). Understanding Organization Realization. *6th International Workshop on Cooperation & Interoperability - Architecture & Ontology.* DESRIST 2010.

Dietz, J.L.G. (2001). DEMO: Towards a discipline of organisation engineering. *European Journal of Operational Research , 128*, 351-363.

Dietz, J.L.G. (2009, November 4). *DEMO-3_Way_of_Working.pdf.* Retrieved April 4, 2010, from DEMO Knowledge Centre: http://www.demo.nl

Dietz, J.L.G. (2003). Deriving Use Cases from Business Process Models. In *Conceptual Modeling* (pp. 131-143). Berlin / Heidelberg: Springer.

Dietz, J.L.G. (2006a). *Enterprise Ontology - Theory and Methodology.* Berlin Heidelberg: Springer-Verlag.

Dietz, J.L.G. (2006b). Enterprise ontology - understanding the essence of organizational operation. *Enterprise Information Systems VII* , 19-30.

Dietz, J.L.G. (2003). The atoms, molecules and fibers of organizations. *Data & Knowledge Engineering* (47), 301-325.

Dietz, J.L.G. (2006c, May). The Deep Structure of Business Processes. *Communications of the ACM , 49* (5), pp. 59-64.

Dow, M., Ravesteyn, P., & Versendaal, J. (2008). Assessing the Quality of Enterprise Services: A Model for Service Oriented Architecture Design. *Proceedings of ICEIS 2008 10th International Conference on Enterprise Information Systems.* Barcelona, Spain.

Endrei, M., Ang, J., Arsanjani, A., Chua, S., Comte, P., Krogdahl, P., et al. (2004). *Patterns: Service-Oriented Architecture and Web Services.* IBM Redbooks.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design.* Prentice Hall.

Erradi, A., Anand, S., & Kulkarni, N.N. (2006). Soaf: An architectural framework for service definition and realization. *IEEE SCC* (pp. 151–158). IEEE Computer Society.

Fenton, N., & Pfleeger, S. (1997). *Software metrics - a Rigorous & Practical Approach* (Second edition ed.). Boston: PWS Publishing Company.

Feuerlicht, G. (2006). Service granularity considerations based on data properties of interface parameters. *International Journal of Computer Systems science & Engineering , 21* (4), pp. 315-327.

Feuerlicht, G., & Meesathit, S. (2004). Design Method for Interoperable Web Services. *ICSOC2004.* New York: ACM Digital Library.

Feuerlicht, G., & Wijayaweera, A. (2007). Determinants of Service Reusability. In H. Fujita, & D. Pisanelli (Ed.), *New Trends in Software Methodologies, Tools and Techniques. 161*, pp. 467-474. Rome: IOS Press.

Flores, F., & Ludlow, J. (1980). Doing and speaking in the office. *Decision Support Systems, Issues and Challenges* , 95-118.

Foody, D. (2005, August 13). *Getting web service granularity right*. Retrieved December 2009, from SOA Zone: http://www.soa-zone.com/index.php?/archives/11-Getting-web-service-granularity-right.html#extended

Fowler, M. (2008). *UML Distilled : A Brief Guide to the Standard Object Modeling Language* (3rd Editiion ed.). Boston: Addison-Wesley.

Goldkuhl, G., & Lyytinen, K. (1982). A language action view of information systems. *Proceedings of the Third International Conference on Information Systems (ICIS'82)*, (pp. 13-29). Ann Arbor, MI, USA.

Group, The Open (2006, June 2). *The Open Group*. Retrieved May 8, 2010, from Service-Oriented Architecture: http://www.opengroup.org/projects/soa/doc.tpl?gdid=10632

Habermas, J. (1981). *Theorie des Kommunikatives Handelns* (Erster Band ed.). Frankfurt am Main: Suhrkamp Verlag.

Hengst, M. den., Adkins, M., Keeken, S. v., & Lim, A. (2005). *Which facilitation functions are most challenging: A global survey of facilitators.* Delft: Delft University of Technology.

Herzum, P., & Sims, O. (2000). *Business Component Factory.* Wiley.

Hevner, A.R., March, S.T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly , 28*.

Hoogendoorn, S. (2003, July 1). *Smart*. Retrieved April 18, 2010, from Beschrijven stappenplan use cases: http://smart.sanderhoogendoorn.org/TASK028.HTML

Hoogervorst, J.A.P. (2009). *Enterprise Governance and Enterprise Engineering.* Diemen, The Netherlands: Springer.

Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (1992). *Object-Oriented Software Engineering: A Use Case Driven Approach.* Addison-Wesley.

Juric, M.B., Mathew, B., & Poornachandra, S. (2006). *Business Process Execution Language for Web Services – An architect and developer's guide to orchestrating web services using BPEL4WS* (2nd ed.). Birmingham: Packt Publishing Ltd.

Keen, M., Bishop, S., Hopkins, A., Milinski, S., Nott, C., Robinson, R., et al. (2004). *Patterns: Implementing an SOA using an ESB.* IBM Redbook.

Klein, H.K., & Myers, M.D. (1999). A set of principles for conducting and evaluating interpretative field studies in information systems. *MIS Quarterly , 23* (1), 67–88.

Knoll, K., & Jarvenpaa, S. (1994). Information Technology Alignment or Fit in Highly Turbulent Environments: The Concept of Flexibility. *Proceedings of the 1994 Computer Personnel Research Conference on Reinventing*, (pp. 1-14). Alexandria, Virginia, United States.

Krafzig, D., Banke, K., & Slama, D. (2005). *Enterprise SOA - Service Oriented Architecture Best Practices.* New Jersey: Prentice Hall.

Lankhorst, M. e. (2005). *Enterprise Architecture at Work - Modelling, Communication, and Analysis.* Berlin Heidelberg: Springer-Verlag.

March, S.T., & Smith, G.F. (1995). Design and natural science research on information technology. *Decision Support Systems , 15* (4), 251-266.

Marks, E.A., & Bell, M. (2006). *Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology.* Wiley.

McGovern, J., Sims, O., Jain, A., & Little, M. (2006). *Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations.* Dordrecht: Springer.

Morton, A., Ackermann, F., & Belton, V. (2003). Technology-Driven and Model-Driven Approaches to Group Decision Support: Focus, Reseach Philosophy, and Key Concepts. *European Journal of Information Systems* (12), 110-126.

Mulder, J.B.F. (2006). *Rapid Enterprise Design.* PhD Thesis, Technical University Delft.

Nonaka, I., & Takeuchi, H. (1991, November-December). The Knowledge-Creating Company. *Harvard Business Review* , pp. 97-130.

Op 't Land, M., De Jong, J., & Goedvolk, H. (2008). *Enterprise Engineering Framework.*

Op 't Land, M., Proper, E., Waage, M., Cloo, J., & Steghuis, C. (2009). *Enterprise Architecture : Creating Value by Informed Governance.* Berlin Heidelberg: Spinger-Verlag.

Papazoglou, M. (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. *Fourth International Conference on Web Information Systems Engineering.* Roma, Italy: IEEE.

Papazoglou, M., & Van den Heuvel, W. (2006). Service-Oriented Design and Development Methodology. *International Journal of Web Engineering and Technology* .

Papazoglou, M.P., & Yang, J. (2002). Design Methodology for Web Services and Business Processes. *Proceedings of the 3rd VLDB-TES Workshop* (pp. 54-64). Hong Kong: Springer.

Polanyi, M. (1966). *The Tacid Dimension.* London: Routledge & Kegan Paul.

Press, C. U. (2010). *Cambridge Advanced Learner's Dictionary*. Retrieved May 30, 2010, from http://dictionary.cambridge.org

Renger, D.R.M., Kolfschoten, G.L., & de Vreede, G.J. (2008). *Patterns in Collaborative Modelling: A Literature Review. In: Climaco, J., Kersten, G.E., Costa, J.P. (eds.) Group Decision and Negotiation.* Faculdade de Economia da Universidade de Coimbra, Coimbra, Portugal.

Richardson, G.P., & Andersen, D.F. (1995). Teamwork in Group Model Building. *System Dynamics Review , 11* (2), pp. 113-137.

Robson, C. (2002). *Real World Research* (2nd edition ed.). Blackwell.

Rosen, M. (2007). A Methodology for Designing Service-Oriented Applications. Cutter Information LLC.

Rosen, M., Lublinsky, B., Smith, K.T., & Balcer, M.J. (2008). *Applied SOA: Service-Oriented Architecture and Design Strategies.* Wiley.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empir Software Eng* (14), 131-164.

Sandhyaduhita, P.I. (2009). *Extending the DEMO methodology for determining information systems requirements.* MSc Thesis, Delft University of Technology.

Schekkerman, J. (2004). *How to survive in the jungle of Enterprise Architecture Frameworks: Creating or choosing an Enterprise Architecture Framework.* Trafford.

Schmelzer, R. (2006, October 3). *Solving the service granularity challenge*. Retrieved May 23, 2010, from The SOA Magazine: http://searchsoa.techtarget.com/tip/0,289483,sid26_gci1172330_mem1,00.html

Shishkov, B., & Dietz, J.L.G. (2003). Deriving use cases from business processes: the advantages of DEMO. *The Fifth International Conference on Enterprise Information Systems.* Angers, France.

Silverman, D. (2005). *Doing Qualitative Research* (2nd Edition ed.). London: SAge Publications Ltd.

Sims, O. (2005, June). Developing the Architectural Framework for SOA – Part 2 – Service Granularity and Dependence Management. *CBDi Journal* .

Steghuis, C. (2006). *Service Granularity in SOA Projects: A Trade-off Analysis.* MSc Thesis, University of Twente, Business Information Technology.

Steinke, G., & Nikolette, C. (2003, 52). Business rules as the basis of an organization´s information systems. *Industrial Management and Data Systems* (1/2).

Stevens, M. (2002, May 21). *Multi-Grained Services*. Retrieved May 23, 2010, from Developer.com: http://www.developer.com/design/article.php/10925_1142661_1/Multi-Grained-Services.htm

Stirna, J., & Kirikova, M. (2008). Integrating Agile Modeling with Participative Enterprise Modeling. *The proceedings of the CAiSE workshop EMMSAD.*

Terlouw, J., Terlouw, L., & Slinger, J. (2009). An Assessment Method for Selecting an SOA Delivery Strategy: Determining Influencing Factors and Their Value Weights. *Proceedings of the Busital workshop.* Amsterdam, The Netherlands.

Terlouw, L., & Albani, A. (2010). *An Enterprise Ontology-Based Approach to Service Specification.* Technical Report, University of Technology, Delft.

Wagner, G. (2005). Rule modeling and markup. In M. Eisinger, & J. Maluszynski (Ed.), *Reasoning Web. 3564*, pp. 251-274. Malta: Springer.

Weggeman, M. (2003). *Kennismanagement: de praktijk.* Schiedam: Scriptum Management.

Weigand, H. (2003). The language/action perspective. *Data & Knowledge Engineering , 47* (3), 299-300.

Weske, M. (2007). *Business Process Management: Concepts, Languages, Architectures.* Berlin Heidelberg: Springer Verlag.

Wiersma, R.W. (2009a, June). De route naar optimale service granulariteit (1). *Business Process Magazine , 15* (4), pp. 41-45.

Wiersma, R.W. (2009b, September). De route naar optimale service granulariteit (2) - Verdieping. *Business Process Magazine , 15* (5), pp. 38-41.

Wiersma, R.W. (2009c, December 1). *Services in SOA*. Retrieved May 24, 2010, from Yenlo: http://www.yenlo.nl/renewiersma/2009/12/01/service-in-soa-2/

Wiersma, R.W., & Ravesteijn, P. (2010). A method for Defining Optimum Service Granularity. *International Information Management Association (IIMA).* Utrecht, the Netherlands.

Yang, J. (2003). Service-oriented computing: Web service componentization. *Communications of the ACM , 46* (10), pp. 35-40.

Yin, R.K. (2003). *Case study research: Methods and Design* (third ed.). Thousand Oaks: Sage Publications.

Zimmerman, O., Krogdahl, P., & Gee, C. (2004, June 2). *Elements of Service-Oriented Analysis and Design – An Interdisciplinary Modeling Approach for SOA Projects.* Retrieved December 2009, from IBM: http://www.ibm.com/developerworks/webservices/library/ws-soad1/index.html

APPENDIXES

A. Overview service granularity determinants

This appendix provides a brief overview of the service granularity determinants. The service granularity determinants are captured by Steghuis (2006). Per service granularity determinant a short description is given and, if relevant, corresponding literature is referenced. Table 11 lists the service granularity determinants in the first column. The second column gives the description of the determinant.

Table 11: *Service granularity determinant descriptions*

Service granularity determinant	Description
Functionality	Captures an intuitive notion of the amount of function contained in a delivered product or in a description of how the product is supposed to be. Functionality can be measured. I.e. function points. (Fenton & Pfleeger, 1997)
Flexibility in Business processes	The ability to change or be changed easily according to the situation. (Cambridge University Press, 2010) • Situations o Business process changes; o Design changes; o Development and maintenance improvements. • Types o Flexibility in functionality: variability in input conditions; o Flexibility in use: related to the outcome of a function; o Flexibility in modification: the ease to modify technology. (Knoll & Jarvenpaa, 1994)
Cognitive and Structural Complexity	A measure about the understandability of the service. Complexity refers to the number of parts and dependencies between them. • Types

Service granularity determinant	Description
	○ Accidental; relates to problems in the software itself, like code optimization issues; ○ Essential; is caused by the problem to be solved. The problem itself is related to essential activities of a process and remains mandatory until the process changes. (Brooks, 1986)
Reusability	The process of adapting a generalized component to various contexts of use. • Determinants ○ Maximum cohesion; ○ Minimum coupling.
Composability	Combining services to construct complex business processes.
Reusability of Legacy	See reusability.
Sourcing	Is the decision making process about outsourcing or insourcing part of the work. • Characteristics ○ Strive for minimum coupling to identify sourceable units; ○ Focus on secondary processes. • To simplify sourcing, the IT services have to fit in with the business services.
Genericity	The concept of making a service applicable to use in many different ways. • Characteristics ○ Dependent on the level of functionality; ○ Associated with reusability and composability. • The goal of making a generic service is to create a service with much functionality, but without unnecessary functionality. ○ Beware of goldplating.
Context-independence	Context-independence means that a service does not need to have knowledge about its surroundings.

Service granularity determinant	Description

	• Characteristics
	    o Loosely coupled;
	    o Self-contained.
	
	• Loosely coupled services are rather coarse grained. (Papazoglou M. , 2003)

Performance	Specifies how well a person, machine, etc. does a piece of work or an activity.
	
	• Characteristics
	    o Quality attribute;
	    o Constraint for service granularity and can influence business and IT services.

B. Legend modeling I-organization

Legend for the interaction model of the I-organization.

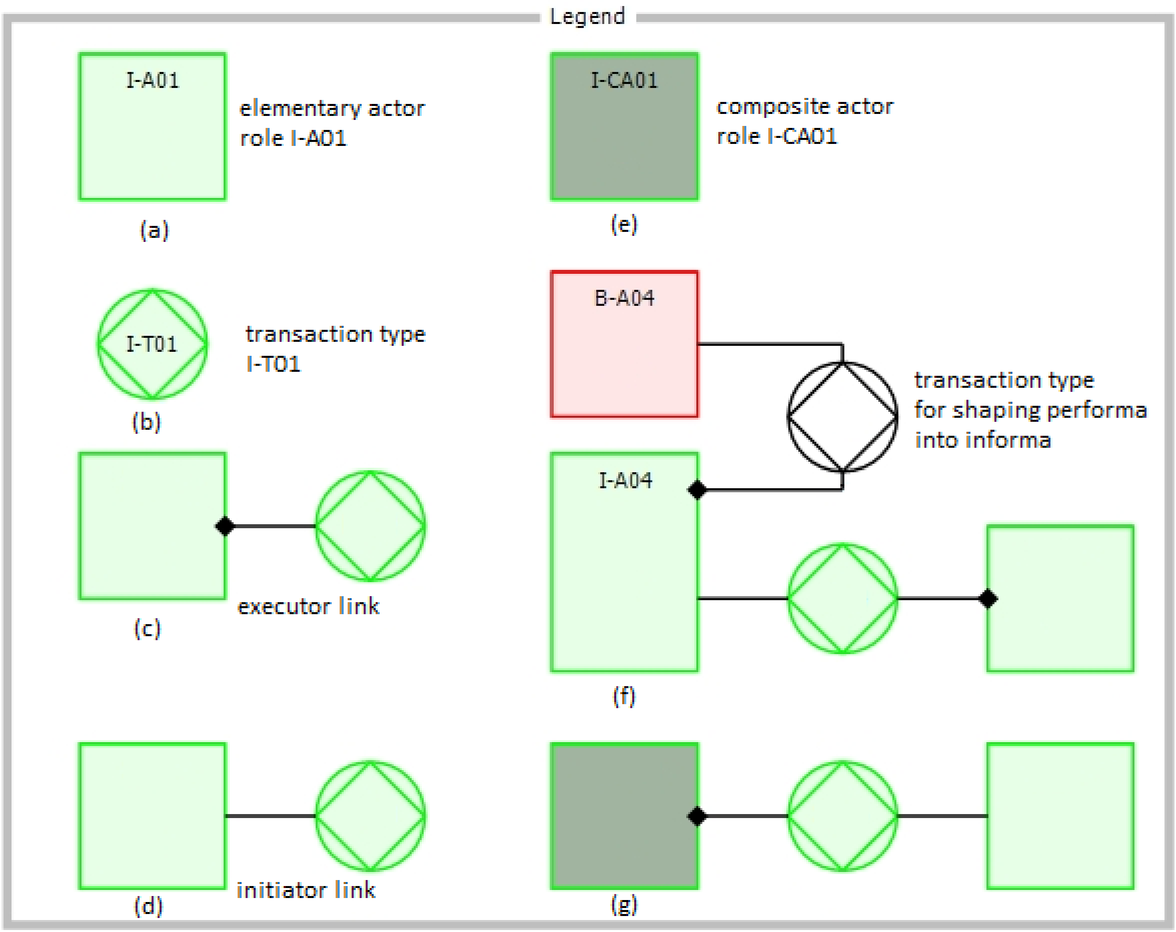


Figure 37: Legend for the interaction model of the I-organization

(a) The symbol denotes an elementary I-actor role
(b) The symbol denotes an I-transaction type
(c) The executor link
(d) The initiator link
(e) The composite actor role; used to handle transactions of the same bank
(f) The initiation of I-transaction by B-actor in its I-shape
(g) The execution of an I-transaction by composite I-actor role

C. Alpha's global actor transaction diagram

Alpha's business model expressed in a global actor transaction diagram. The model has been coded in order to guarantee confidentially. The business architects of Alpha already created this model before the case study. The case study embedded the model as a reference model for the organization after agreement of Alpha.

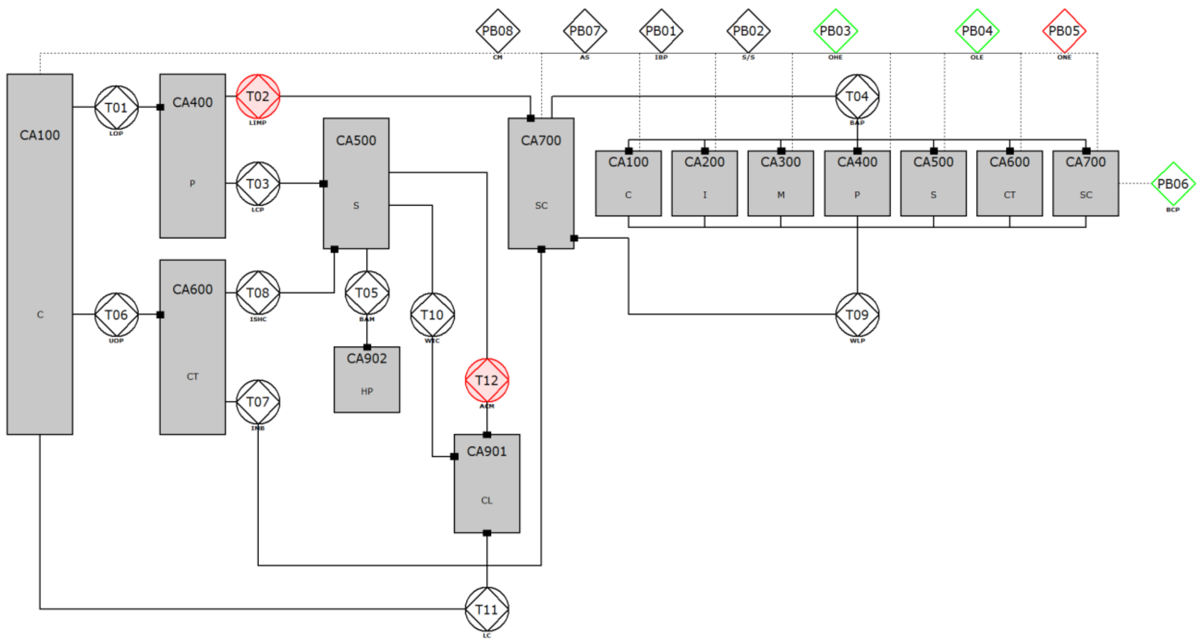


Figure 38: Global Actor Transaction Diagram of Alpha

D. Alpha's ontological models CA700

In order to scope the case study composite actor CA700 has been selected to be analyzed. The models that are shown in this paragraph have been created by collaborative modeling and were the first detailed organizational construction diagram and process structure diagram of actor CA700. These models provided the foundation for the several cycles in the action taking process. Figure 39 shows the organizational construction diagram, Figure 40 shows the process structure diagram, and Figure 41 shows the state diagram.

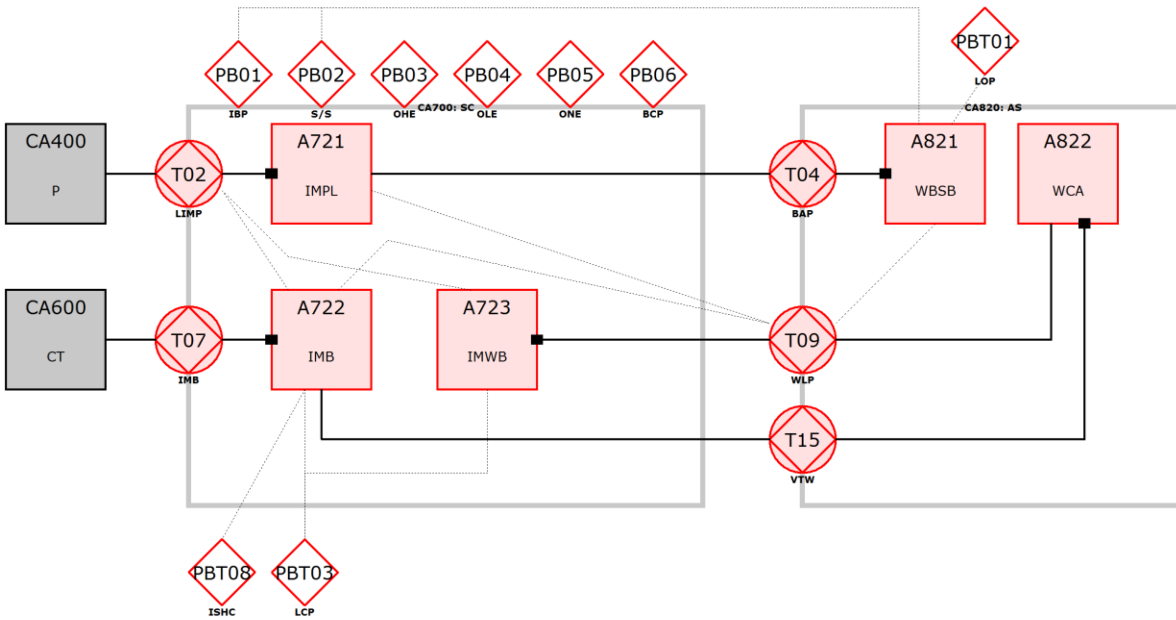


Figure 39: Detailed Organization Construction Diagram of CA700 after the first iteration

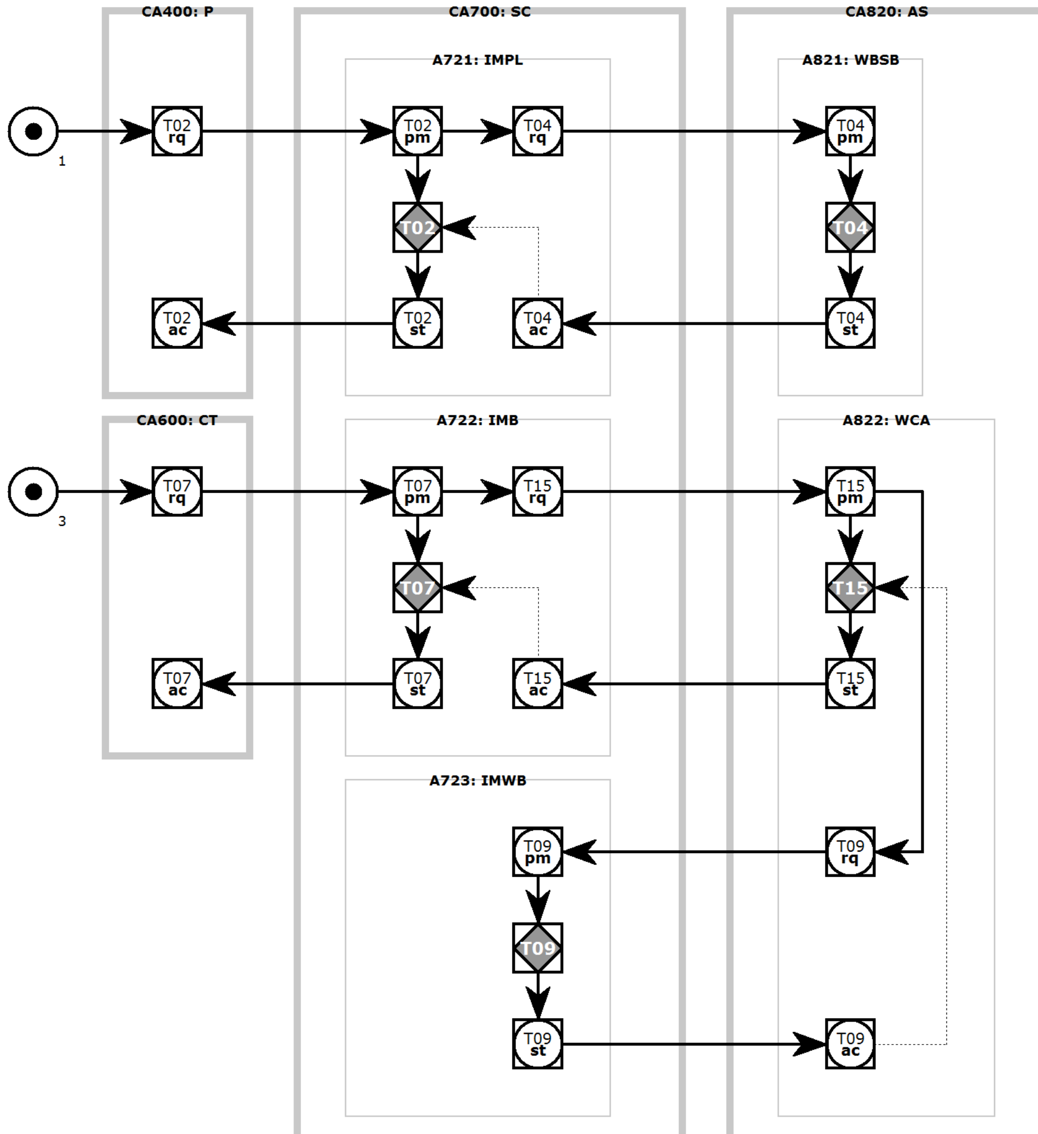


Figure 40: Process Structure Diagram of CA700 after the first iteration

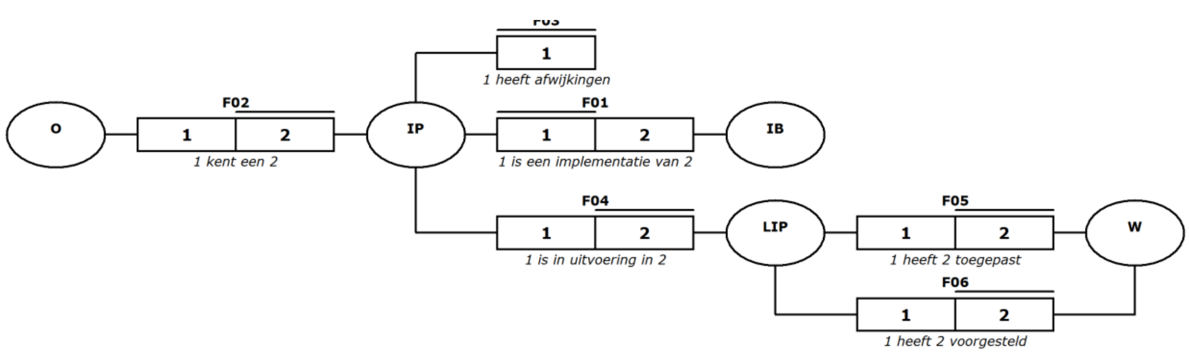


Figure 41: State Diagram of CA700 after the first iteration

E. Template infological analysis

This template for infological analysis states all C-acts from the basic transaction pattern of the transaction axiom for case study Alpha. All mentioned C-acts are part of the process structure diagram and are the foundation for the information analysis that follows. The required information objects are listed in the tables below per agenda of an actor. This table is an extended version of the Information Use Table (IUT), which is part of the DEMO methodology, and has been designed by Joop de Jong (De Jong & Dietz, 2010).

The table starts with the agenda of the actor. In case the agenda states T01/rq, it means its C-fact. This must be interpreted like the "on requested" state of the request act. The second column of the table addresses the description of the following act, when the agenda is executed. This means that an "on requested" C-fact follows by a "promise" C-act. The third column is mentioned for the situational information. This could be interpreted like the business rules or pre-conditions that are concerned with the C-act. The context of the agenda and the following C-act are evaluated against the situational information. When business rules or pre-conditions are not met for an "on requested" C-fact, then a "decline" could be given, as defined in the standard transaction pattern. See paragraph 4.3.2 for further details. The fourth column addresses the operational information. This must be seen as the required information that must be accessible and used in execution of the C-act. The last column mentions the data to process. These are the C-acts to be executed.

The used process structure diagram has been created at Alpha on February 23, 2010. The process structure diagram shows the sociological transactions for two composite actors within the total organization construction diagram of Alpha.

Actor CA400: P

Agenda	Act	Situational Information	Operational Information	Data to process
				T02/rq
T02/st				T02/ac

Actor A721: IMPL

Agenda	Act	Situational Information	Operational Information	Data to process
T02/rq				T02/pm
T02/pm				T04/rq
T04/st				T04/ac T02

				T02/st

Actor A821: WBSB

Agenda	Act	Situational Information	Operational Information	Data to process
T04/rq				T04/pm
T04/pm				T04 T04/st

Actor CA600: CT

Agenda	Act	Situational Information	Operational Information	Data to process
				T07/rq
T07/st				T07/ac

Actor A722: IMB

Agenda	Act	Situational Information	Operational Information	Data to process
T07/rq				T07/pm
T07/pm				T15/rq
T15/st				T15/ac T07 T07/st

Actor A822: WCA

Agenda	Act	Situational Information	Operational Information	Data to process
T15/rq				T15/pm
T15/pm				T09/rq

T09/st				T09/ac T15 T15/st

Actor A723: IMWB

Agenda	Act	Situational Information	Operational Information	Data to process
T09/rq				T09/pm
T09/pm				T09 T09/st

All C-acts are classified by means of execution. Every C-act can be executed manually, automated, or semi-automated. See paragraph 2.2.1. By classifying C-acts one gets a better understanding of the informational behavior and candidate informational services that are provided by an automated system.

C-act	Manual / Automated / Semi-automated
T02/rq	
T02/pm	
T02/st	
T02/ac	
T04/rq	
T04/pm	
T04/st	
T04/ac	
T07/rq	
T07/pm	
T07/st	

C-act	Manual / Automated / Semi-automated
T07/ac	
T09/rq	
T09/pm	
T09/st	
T09/ac	
T15/rq	
T15/pm	
T15/st	
T15/ac	

F. Alpha's infological construction model

Based on the informational analysis the infological model has been created. This model in Figure 42 is the first version.

This modeling method continues with the eIUTs and transforms it into an infological construction diagram, which inherits the same characteristics as the ontological diagrams (De Jong & Dietz, 2010). To get from the eIUTs toward an informational construction diagram, one has to start by modeling an information boundary. This boundary is the first step in getting towards an implementation of a system (De Jong & Dietz, 2010). By following the modeling approach of De Jong (2008) and Sandhyaduhita (2009) the elementary B-actors in the ontological model are to be placed outside of the boundary. These B-actors become composite actors in the information organization. These composite I-actors are the representation of the B-actors that have been shaped into their informa ability. In their informa ability the actors are able to initiate infological transactions. The best way to start modeling the information model is to create an elementary I-actor, which is an information manager for the composite actor (Wiersma, 2009b). This information manager is considered to be an information access point for the external I-actor, which is the informa shaped B-actor. The elementary information manager is capable of initiating all the necessary I-transactions. Introducing the information manager actor is the second step in realizing an information organization. The information manager can be perceived as an interface for information, like a user interface for applications or a WSDL for a web service call.

The information manager will access production facts and is responsible for reproducing the semantics of the data to the initiator. The production facts to be reproduced are stated in the business rules column and the operational information column of the eIUT. From the eIUT two types of facts reproduction can be distinguished in the I-organization. These two types are (De Jong & Dietz, 2010):

(a) Original facts reproduction
Original facts are P-facts, entities, dependent facts and original properties. A B-actor in its informa ability or other I-actor of the same organization can initiate an I-transaction of an original fact reproduction to an I-actor.

(b) Derived facts computation
A derived fact is computed based on other facts, i.e. original facts, other derived facts, or external facts. It is a part of a semantic meaning that cooperating I-actors exchange in order to provide B-actors gained semantic meanings for extracting information to be able to act with within the B-organization. A B-actor in its informa ability or other I-actor of the same organization can initiate an I-transaction of a derived fact production (computation) to an I-actor. The executor of an I-transaction of a derived fact computation is an I-actor role. The I-transaction normally encloses one or more I-transactions. This conforms to the composition axiom of the PSI-theory.

According to (Sandhyaduhita, 2009) a third type of reproducing facts exists. This is known as the external facts reproduction. From an external organization point of view, an initiating I-transaction is perceived as a B-transaction and is executed by a B-actor. Accordingly, the external organization delivers the requested fact as an information product. In (Wiersma, 2009b) external fact reproduction is modeled as integration services, which enables an external organization to initiate I-transactions directly.

The operational information from the eIUTs, which is on the P-act line, is transformed in infological composite actors that are responsible for the storage and retrieval of the production facts. Two I-transactions support this functionality, one for the storage of the production fact and one for the retrieval of the production fact.

Other operational information that is part of the eIUT is retrieval of information from production banks. Every accessed production bank will be represented by an elementary I-actor, which is responsible for retrieval of information from the bank. Retrieval of the bank information is initiated by an I-transaction that is created for this purpose.

All the operational information from the eIUTs must be modeled now, because operational information must be retrieved from an internal bank or an external bank. Otherwise the operational information is provided by the information manager and must be stored into the bank. In case operational information is used, which could not be addressed from the ontological diagram, then something is missing in the ontological model (Sandhyaduhita, 2009).

The transaction provider actors are required because of their responsibility to retrieve the location where the facts are stored. Finding its store only can be done by an intelligent action executed by an I-actor (De Jong & Dietz, 2010).

The business rules from the eIUTs also are grounded in internal or external production banks. One of the most explicit forms in information usage is the interstriction in the organizational construction diagram. The interstriction shows the sources of the information requirements that are used in the C-act or in the P-act. This means that the gathered information via interstriction is used to support the activity and therefore can be used to validate business rules. Following the methodology of the interstriction model all actor – bank combinations are to be considered as information links, which is shown in an Actor Bank Diagram (ABD). Grounded on this theory interstriction appears not only between non-related actors and transactions, but also between all actors that are related to their transactions (Dietz, 2006a).

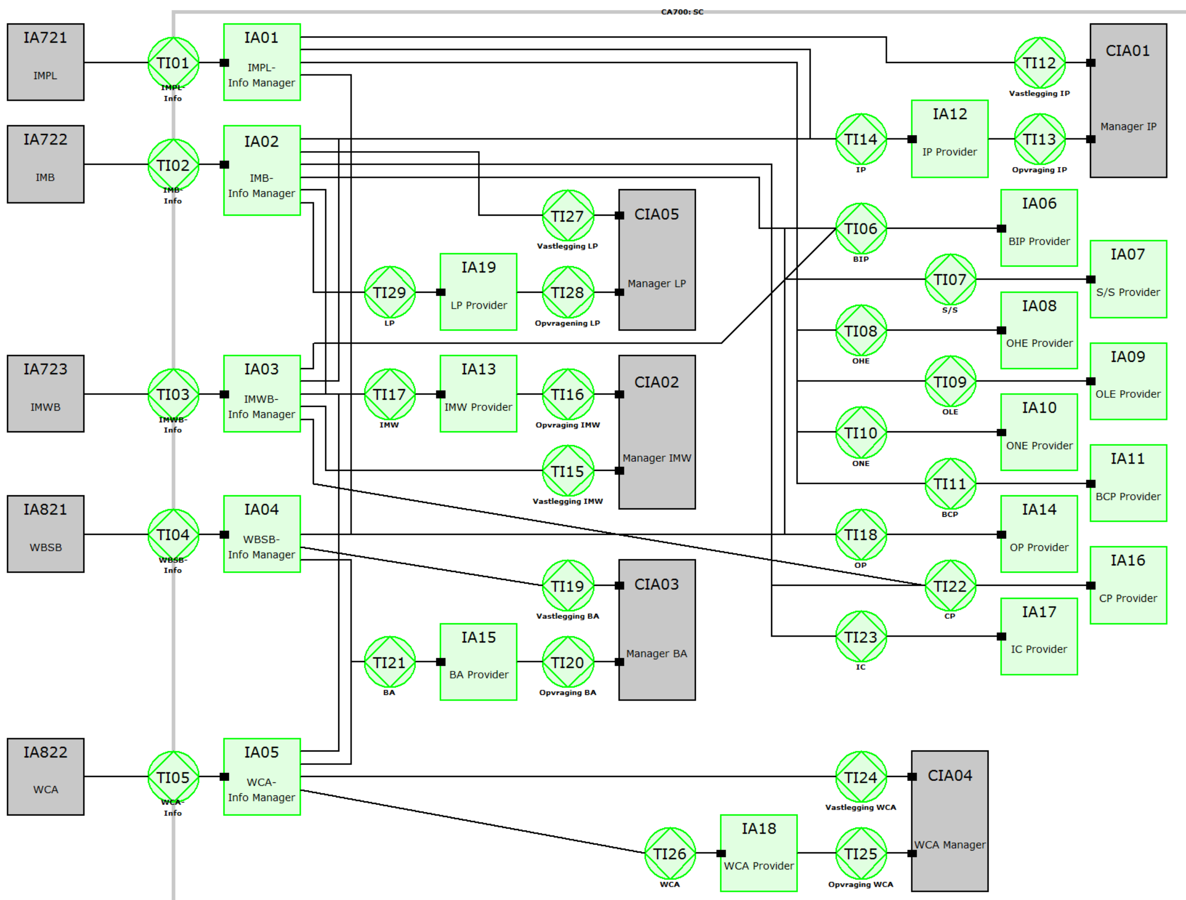


Figure 42: Infological model CA700 after the first iteration

G. Service granularity grid

To structure the service granularity discussion we made a grid of the determinants of service granularity at one side, and the infological transactions on the other side. The crossing cells in the grid are topics of discussion. The grid is shown in

Table 12. The orange colored cells imply that the determinants of service granularity are considered to influence in the software development process.

Table 12: *Service granularity grid*

Service granularity grid

Determinant

I-Transaction	Functionality	Flexibility in Business processes	Cognitive and Structural Complexity	Reusability	Composability	Sourcing	Genericity	Context-independence	Performance
TI01									
TI02									
TI03									
TI04									
TI05									
TI06									
TI07									
TI08									
TI09									
TI10									
TI11									
TI12									
TI13									

	Functionality	Flexibility in Business processes	Cognitive and Structural Complexity	Reusability	Composability	Sourcing	Genericity	Context-independence	Performance
TI14									
TI15									
TI16									
TI17									
TI18									
TI19									
TI20									
TI21									
TI22									
TI23									
TI24									
TI25									
TI26									
TI27									
TI28									
TI29									

H. Influencing service granularity determinants

While we were discussing the information transactions with respect to the service granularity determinants (see the grid in appendix G) we concluded the following. Our findings are reported in Table 13. Per infological transaction we explained why certain service granularity determinants influence implementation design choices. To save space, transactions and conclusions are consolidated in the table where conclusions are overlapping for multiple transactions. The conclusions were made by the participants in the joint session with senior operational business architects, their department manager, and the senior technical solution architect.

Table 13: *Conclusions of the service granularity grid*

I-Trx	Determinant	Why influencing implementation design
TI01 TI02 TI03 TI04 TI05	Functionality	The transaction supports no unambiguously defined unit of work. The provided amount of functionality is too diverse. The objective of transaction identification is to get transaction that can be translated in a software component of distinctive functional meaning that typically encapsulates a high level business concept. This is supported by the definition of (Krafzig, Banke, & Slama, 2005) in paragraph 2.2.1.
TI01 TI02 TI03 TI04 TI05	Flexibility	The central organized access to all information objects that a single performa requires, in its informa ability, limits the flexibility to change. In case an actor requires different information, then it has to be programmed into the central information access object, which possibly has impact on existing information objects that are already linked. A second limitation of central access to information is the single directed shaping possibility of performa´s. Performa´s can shape in just one informa ability instead of various informa´s.
TI01 TI02 TI03 TI04 TI05	Complexity	Realizing a central information access area for a single performa is complex. It can be interpreted like an information portal, which is a gateway to different kind of information sources. Constructing a centralized information access area in software is considered to be complex.
TI05	Reusability	When more than one performa requires the ability to shape into an informa that already provides the required information, then this shaping is not supported by centralizing information access per performa. Prohibiting this kind of shaping limits reusability of transactions.
TI06 TI07	Reusability	The information model already shows the reusability of this transaction. This means that the software implementation must support various contexts of use. The realized software component

I-Trx	Determinant	Why influencing implementation design
		must be as autonomous as possible, which is in favor of minimum coupling.
TI08 TI09 TI10 TI18	Composability	In essence constructing several transactions into one composition is also a form of reusability. These transactions overlap completely when one speaks about the technical implementation. Therefore one pleads to compose the transactions into one generic and reusable transaction.
TI06 TI07 TI08 TI09 TI10 TI18	Genericity	This is directly related to the conclusion on the previous service granularity determinant. A generic transaction supports its utilization in many different ways. Due to the defined unit of functionality that must be supported, a generic transaction can be implemented[7].
TI06 TI07 TI08 TI09 TI10 TI18	Context independence	Due to the reusability/composability determinant and the genericity determinant, the transaction must be independent of its context. When a transaction strives for minimum coupling, then the transaction must not have a lot of knowledge about its surroundings. These characteristics lead to loosely coupled and self-contained implementations of software components.
TI12 TI13 TI15 TI16 TI19 TI20	Genericity	When a transaction's purpose is to save or retrieve information from a transaction bank, then it has to be generic. To implement this in software the transaction has to be aware of the underlying data structures. Therefore, when saving or retrieving of information is in a generic nature, all requesting informa's do not have to have knowledge about the underlying data structure.
TI12 TI13 TI15 TI16 TI19 TI20	Performance	Saving or retrieving information from a transaction bank must consume a minimum amount of time. Because DEMO is transaction oriented we have to have supporting software that also executes single transactions in a minimum amount of time. This could lead to split transaction oriented software implementations from batch oriented implementations.

[7] The unit of functionality is defined by the transaction after applying the DEMO methodology strictly. Because DEMO obviously defines the unit of functionality in this case, no participant doubted about the functionality determinant. Therefore the functionality determinant is not mentioned for these transactions.

I-Trx	Determinant	Why influencing implementation design
TI14 TI21	Functionality	Transforming the retrieved information from the transaction bank into the format that has been requested requires specific functionality. The required functionality is not specified precisely by these transactions.
TI14 TI21	Flexibility	The transactions must provide all the required transformations of information that are based on the retrieved information from the transaction bank. This centralized the transformations of data into a single transaction, which contributes to the ease of changing according to the situation.
TI14 TI21	Complexity	It could be relative complex to implement all the information transformations within the purpose of one transaction.
TI14 TI21	Reusability	The information model already shows the reusability of this transaction. This means that the software implementation must support various contexts of use. The realized software component must be as autonomous as possible, which is in favor of minimum coupling.
TI22 TI23	Reusability	The information model already shows the reusability of these transactions. The transactions retrieve information from production banks and are commonly used within the system boundaries. This means that the software implementation must support various contexts of use. The realized software component must be as autonomous as possible, which is in favor of minimum coupling.
TI22 TI23	Sourcing	Transactions that retrieve information from production banks must be easily changeable or replaceable by other sources. This makes that these kinds of transactions are not coupled or are coupled with a minimum of dependencies.
TI22 TI23	Genericity	A generic transaction that retrieves information from a production banks must support its application in many different ways. Due to the defined unit of functionality that must be supported, a generic transaction can be implemented[7].
TI11 TI17 TI24 TI25 TI26	Not applicable	These transactions are obsolete from a technical point of view, because the purposes of the transactions are not supported by the underlying data structures. Therefore the infological model must be reorganized in order to apply to the underlying data structures. Declaring these transactions obsolete does not mean that the

I-Trx	Determinant	Why influencing implementation design
TI27 TI28 TI29		purpose is not justified, but that the result of the transaction is constructed in a different way.

I. Alha's ontological model after second iteration

The models that are shown in this paragraph are created by collaborative modeling. All the participants of the joint sessions contributed with their fundamental knowledge about the business and IT environment of Alpha. These models represent the organizational construction diagram in Figure 43, the process structure diagram in Figure 44, the state diagram in Figure 45 and the infological construction model in Figure 46. These models were created during the action taking phase of the action research cycle after the second iteration.

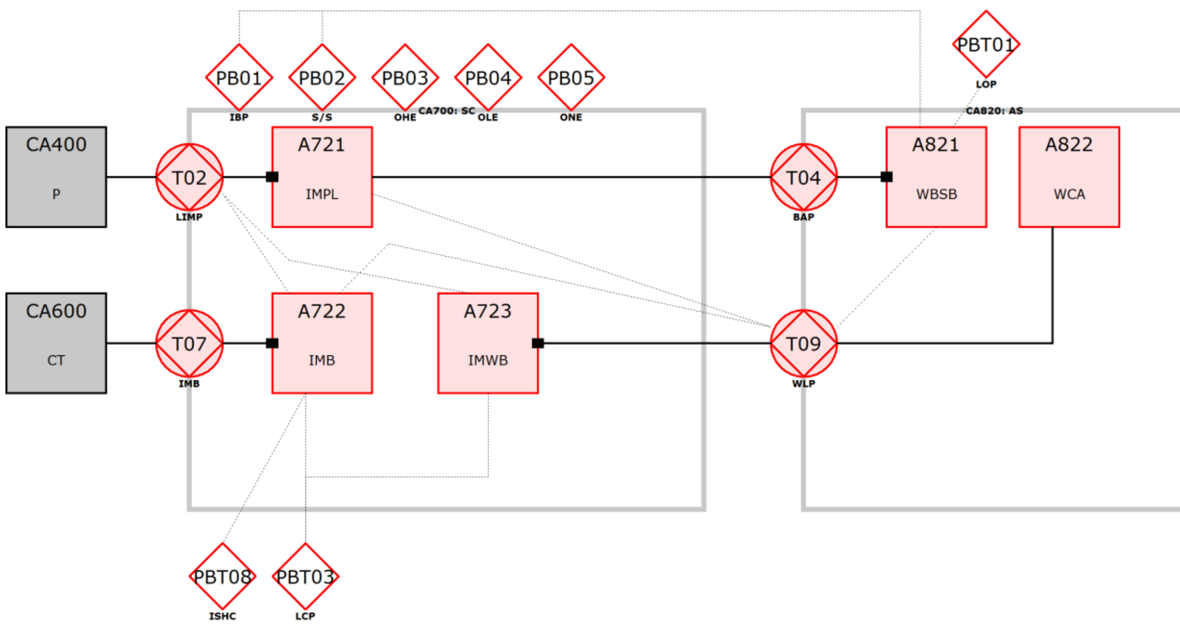


Figure 43: Detailed Organization Construction Diagram of CA700 after the second iteration

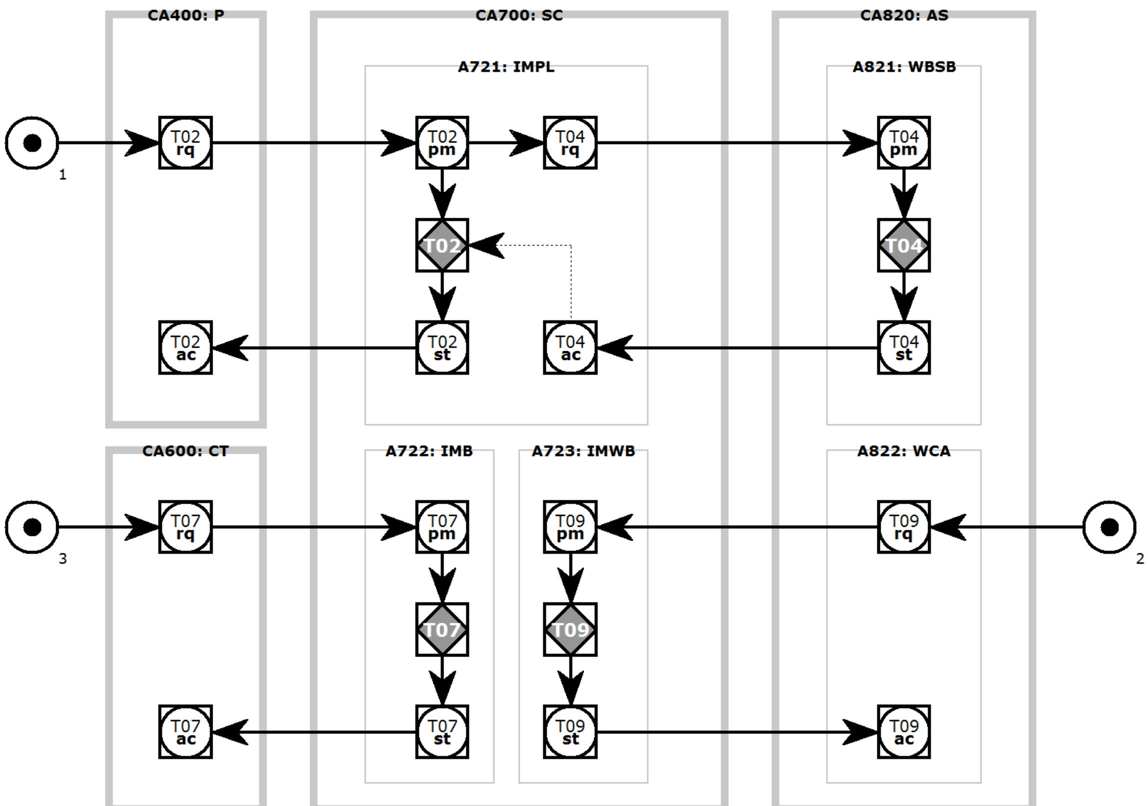


Figure 44: Process Structure Diagram of CA700 after the second iteration

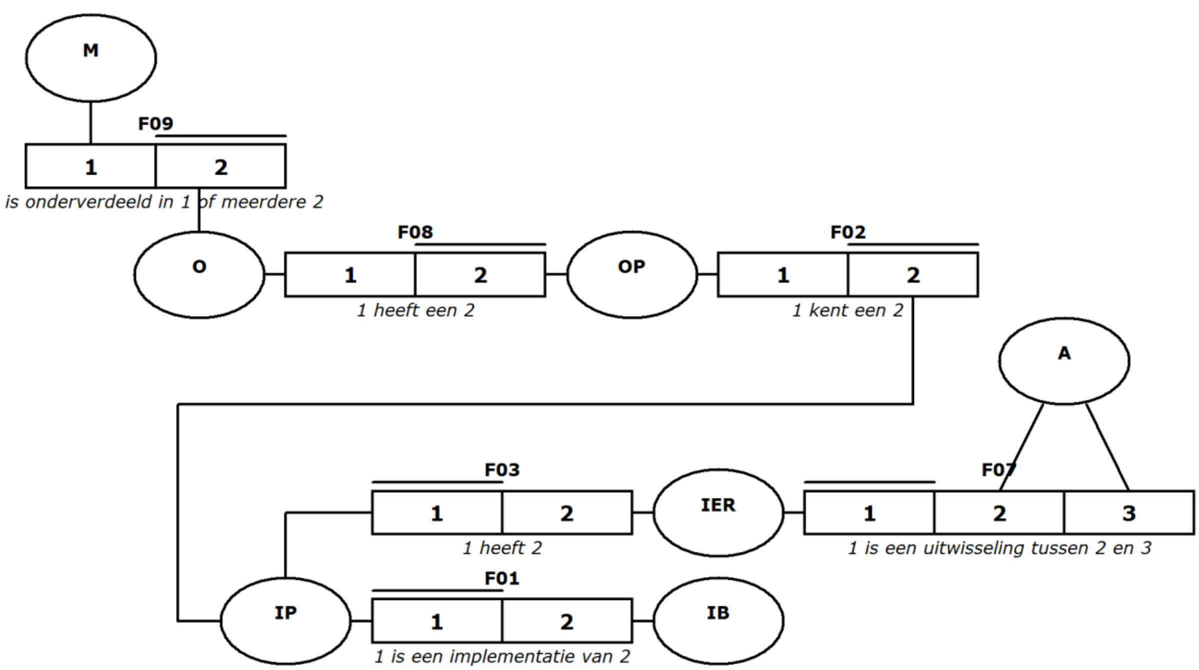


Figure 45: State Diagram of CA700 after the second iteration

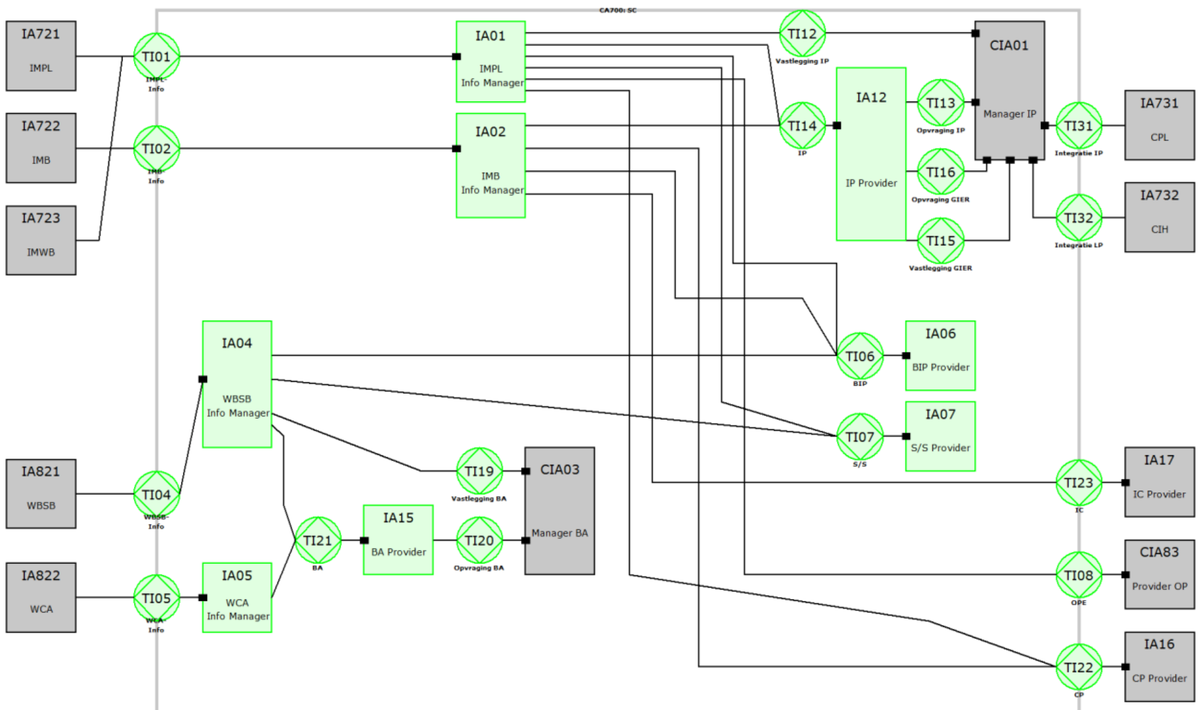


Figure 46: Infological model CA700 after the second iteration

J. Use case template

The paragraph shows the template that has been used to define kite level use cases that are correlated to an information transaction. The template is an adjusted version of the template illustrated in (Fowler, 2008).

Use Case: <name>

Goal Level: Kite level

Part of transaction: <Transaction identifier>

Main Success Scenario: <Description of the main success scenario>

Extensions: <Description of the alternative scenarios>

K. Alpha's infological contruction model after third iteration

The model that is shown in this paragraph is created by collaborative modeling. All the participants of the joint sessions contributed with their fundamental knowledge about the business and IT environment of Alpha. The model represents the infological construction model in Figure 47. The model was created during the action taking phase of the action research cycle after the third iteration.

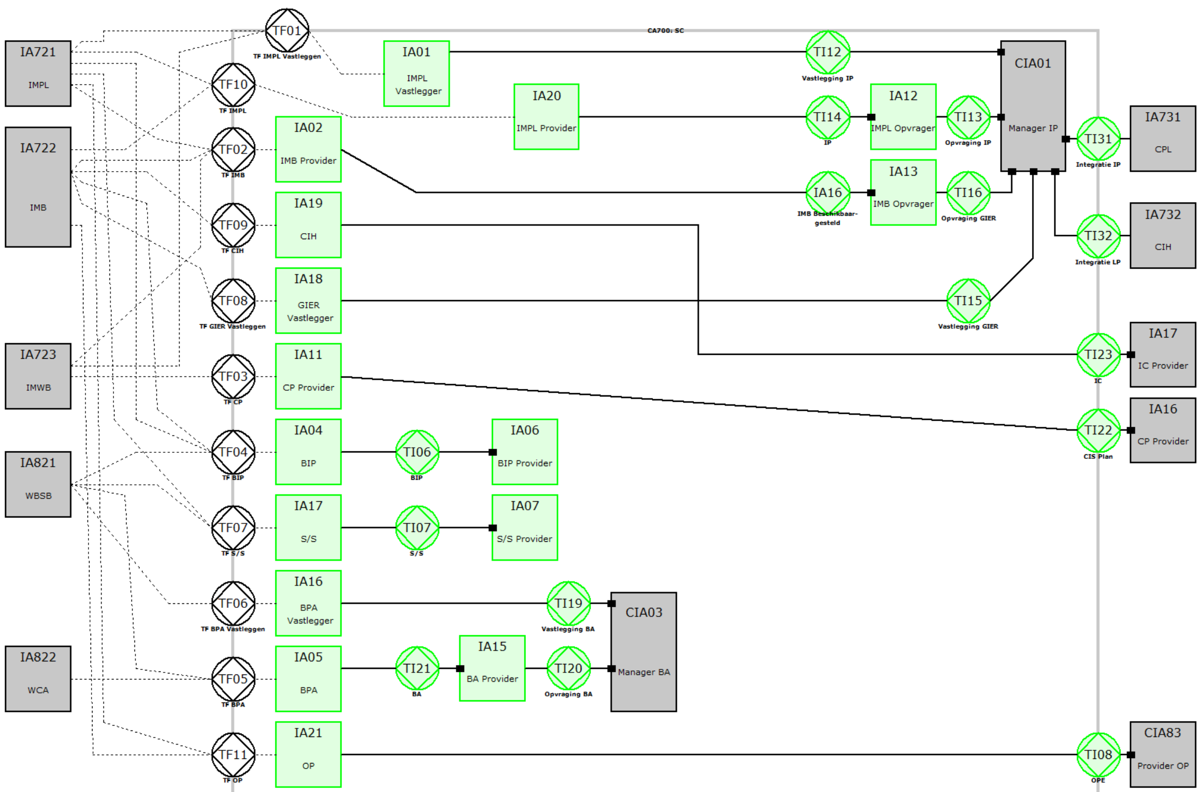


Figure 47: Infological model CA700 after the third iteration

L. Authorization matrix

To structure the available transformations we made a grid of the transformations. A transformation is the ability of a business actor to shape into an infological actor. There exists a n..m relationship between business actors (performa's) and infological actors (informa's). This means that a performa can shape in multiple informa's and that one informa can support the information need of multiple performa's. The crossing cells in the grid are the possible transformations. The grid is shown in Table 14. The orange colored cells imply the transformation, which are modeled in Figure 47.

Table 14: *Authorization matrix for actor transformations*

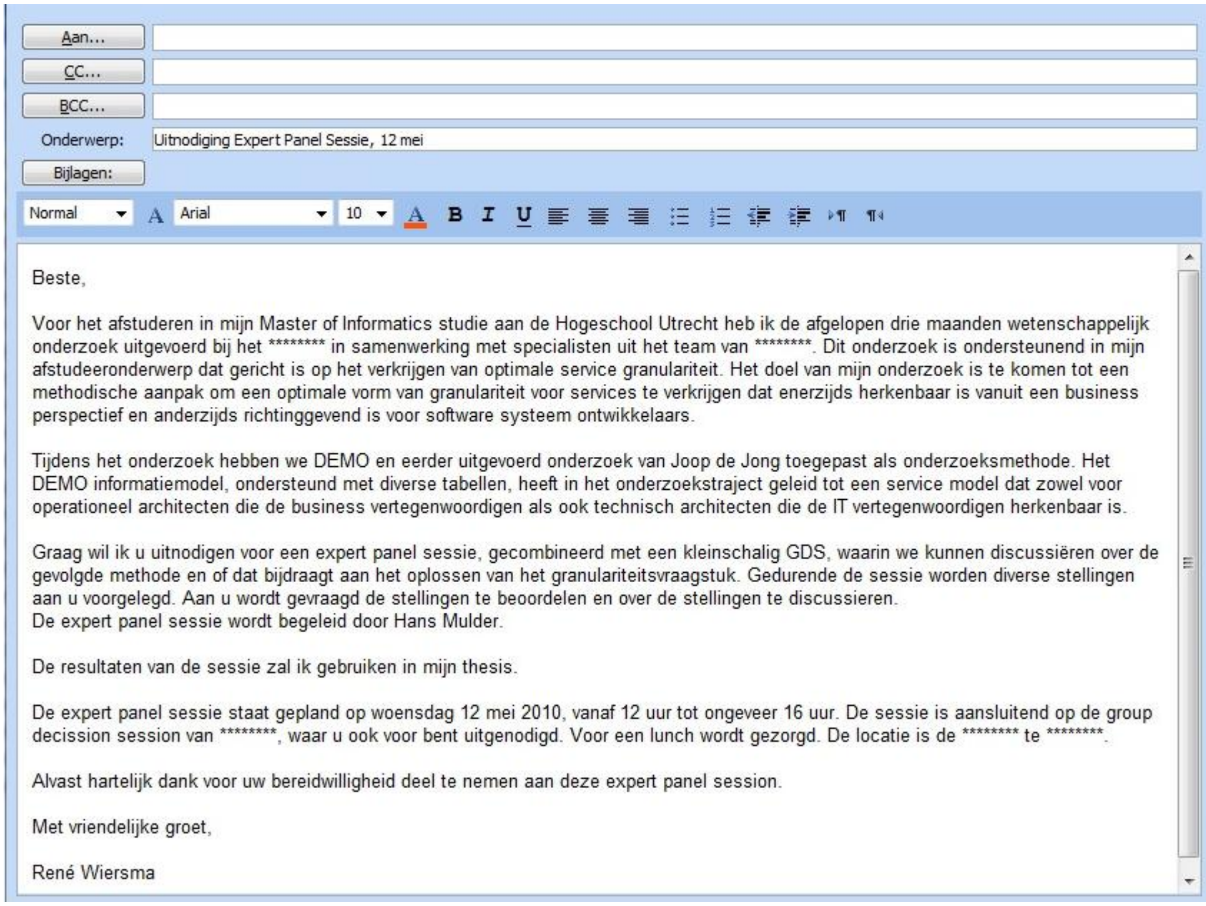
Autorization matrix

I-Actor

B-Actor	**I-A01**	**I-A02**	**I-A04**	**I-A05**	**I-A11**	**I-A16**	**I-A17**	**I-A18**	**I-A19**	**I-A20**	**I-A21**
B-A721											
B-A722											
B-A723											
B-A821											
B-A822											

M. E-mail invitation

To invite experts to participate in the GDSS session I sent the following e-mail. See Figure 48. The e-mail is in Dutch, because all participants are inhabitants of the Netherlands.



Aan...
CC...
BCC...
Onderwerp: Uitnodiging Expert Panel Sessie, 12 mei
Bijlagen:

Beste,

Voor het afstuderen in mijn Master of Informatics studie aan de Hogeschool Utrecht heb ik de afgelopen drie maanden wetenschappelijk onderzoek uitgevoerd bij het ******** in samenwerking met specialisten uit het team van ********. Dit onderzoek is ondersteunend in mijn afstudeeronderwerp dat gericht is op het verkrijgen van optimale service granulariteit. Het doel van mijn onderzoek is te komen tot een methodische aanpak om een optimale vorm van granulariteit voor services te verkrijgen dat enerzijds herkenbaar is vanuit een business perspectief en anderzijds richtinggevend is voor software systeem ontwikkelaars.

Tijdens het onderzoek hebben we DEMO en eerder uitgevoerd onderzoek van Joop de Jong toegepast als onderzoeksmethode. Het DEMO informatiemodel, ondersteund met diverse tabellen, heeft in het onderzoekstraject geleid tot een service model dat zowel voor operationeel architecten die de business vertegenwoordigen als ook technisch architecten die de IT vertegenwoordigen herkenbaar is.

Graag wil ik u uitnodigen voor een expert panel sessie, gecombineerd met een kleinschalig GDS, waarin we kunnen discussiëren over de gevolgde methode en of dat bijdraagt aan het oplossen van het granulariteitsvraagstuk. Gedurende de sessie worden diverse stellingen aan u voorgelegd. Aan u wordt gevraagd de stellingen te beoordelen en over de stellingen te discussieren.
De expert panel sessie wordt begeleid door Hans Mulder.

De resultaten van de sessie zal ik gebruiken in mijn thesis.

De expert panel sessie staat gepland op woensdag 12 mei 2010, vanaf 12 uur tot ongeveer 16 uur. De sessie is aansluitend op de group decission session van ********, waar u ook voor bent uitgenodigd. Voor een lunch wordt gezorgd. De locatie is de ******** te ********.

Alvast hartelijk dank voor uw bereidwilligheid deel te nemen aan deze expert panel session.

Met vriendelijke groet,

René Wiersma

Figure 48: Invitation to participate in GDSS session

N. Set-up GDSS meeting room

The set-up of the meeting room to facilitate the GDSS. The set-up is shown in Figure 49.

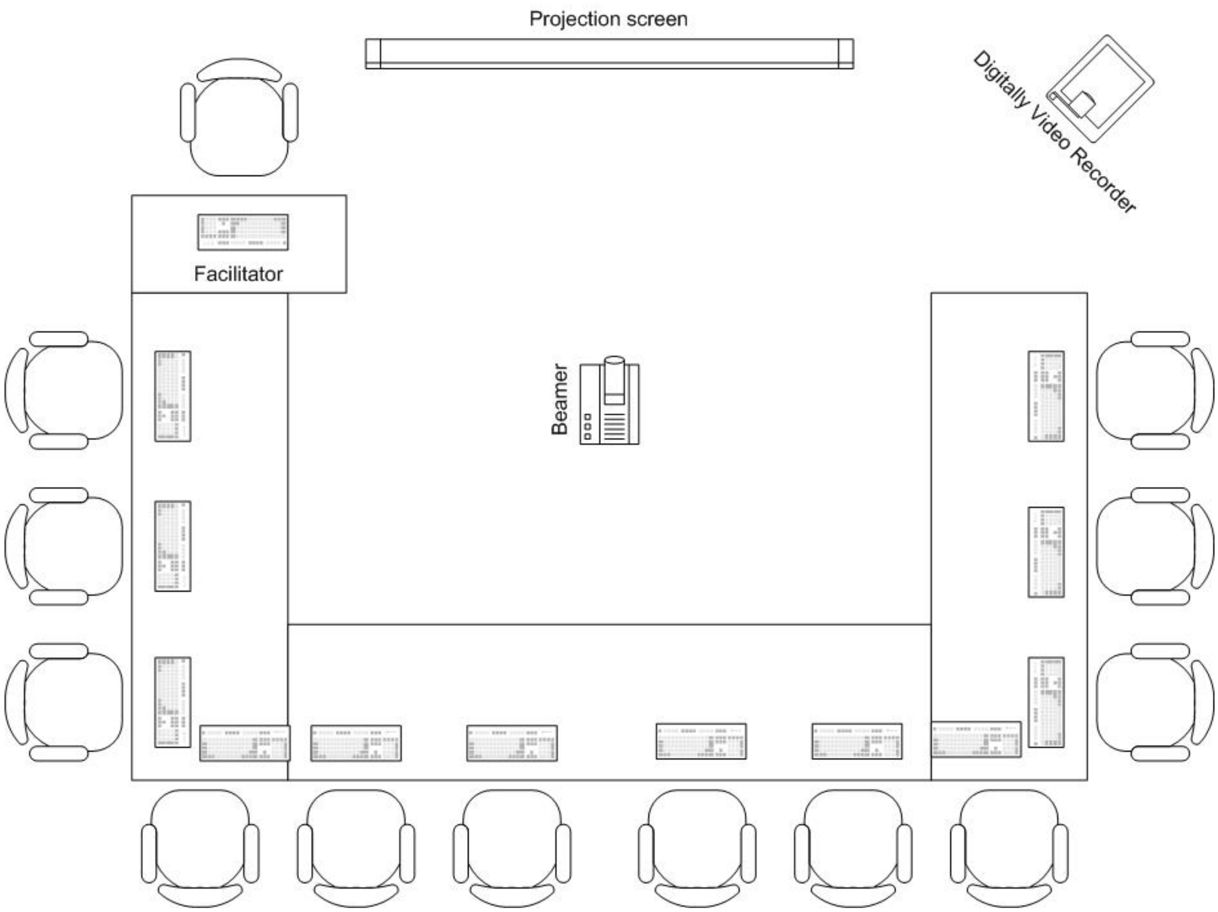


Figure 49: Set-up of the meeting room to facilitate the GDSS

O. GDSS propositions

The propositions that I formulated to validate the conducted improving case study including its results measures against the objectives are listed below. I categorized the 32 (thirty two) propositions in 5 (five) main categories. Per category the propositions are listed. See Table 15, Table 16, Table 17, Table 18 and Table 19 below. The header row of the table defines the main category. The left column is the proposition's coding. The right column is the formulated proposition in Dutch. The italic written sentences are the same propositions translated in English.

Table 15: *GDSS propositions about service granularity*

Service Granularity	
GR1	Granulariteit van services is een kritische succes factor in informatie systeem ontwikkeling? *Service granularity plays a critical success factor in information system development?*
GR2	Een optimale granulariteit van services draagt bij aan de herbruikbaarheid van informatie objecten? *An optimized service granularity contributes to the reusability of information objects?*
GR3	Een optimale granulariteit van services draagt bij aan de flexibiliteit van bedrijfsprocessen? *An optimized service granularity contributes to the flexibility in business processes?*
GR4	Een optimale granulariteit reduceert de complexiteit van de informatie huishouding? *An optimized service granularity reduces the complexity of the information landscape?*
GR5	Granulariteit van services is een generiek vraagstuk dat in iedere organisatie speelt? *Service granularity is a common topic of discussion in every organization?*
GR6	Optimale service granulariteit is in iedere organisatie anders? *An optimized form of service granularity is different in every*

	organization?
GR7	Informatie analyse moet sturend zijn in het optimaliseren van service granulariteit? *The process of information analyses must be directive in optimizing service granularity?*

Table 16: *GDSS propositions about the process of information analyses*

The process of information analyses	
IM1	Het Generic System Development Proces (GDSP) is uitvoerbaar op het proces van informatiemodellering? *The Generic System Development Process (GDSP) is applicable for the process of information analyses?*
IM2	Het informatiemodel moet herkenbaar zijn voor de business die het ondersteund? *The information model must be in a recognizable fashion for the business that the model supports?*
IM3	Het informatiemodel moet richtinggevend zijn voor software systeem ontwikkelaars? *The information model must be directive for information system developers?*
IM4	Een optimaal granulair informatiemodel verhoogt de kwaliteit van software? *An information model that reflects an optimized service granularity raises the quality of software?*
IM5	Informatie analyse kent een top-down aanpak, redenerend vanuit de business laag? *The process of information analyses follows a top-down strategy, starting in the business layer?*
IM6	Informatie analyse moet het bestaande systeemlandschap meewegen in voorgestelde informatiemodellen?

The process of information analyses	
	The process of information analysis must include the current information landscape in the information models?
IM7	Collaboratief modelleren vanuit een business perspectief en een IT perspectief leidt tot betere oplossingen? *Collaborative modeling with participating business representatives and participating IT representatives results in better solutions?*

Table 17: *GDSS propositions about activities in the conceptual model*

Activities in the conceptual model	
AB1	De activiteiten uit het basismodel zijn herkenbaar voor het optimaliseren van service granulariteit? *The activities that are part of the conceptual model are recognizable for optimizing service granularity?*
AB2	De activiteiten uit het basismodel zijn relevant voor het optimaliseren van service granulariteit? *The activities that are part of the conceptual model are relevant for optimizing service granularity?*
AB3	De activiteiten uit het basismodel zijn samenhangend in het optimaliseren van service granulariteit? *The activities that are part of the conceptual model are coherent for optimizing service granularity?*
AB4	De activiteiten uit het basismodel dragen bij tot het optimaliseren van service granulariteit? *The activities that are part of the conceptual model contribute in optimizing service granulalrity?*

Table 18: *GDSS propositions about the contribution of DEMO in finding an optimum in service granularity*

The contribution of DEMO in finding an optimum in service granularity	
DM1	Het DEMO infologische model is een functionele weergave van de informatie samenstelling? *The DEMO infological model is a functional representation of the information construction?*
DM2	Het actor-transactie-actor patroon in het DEMO infologische model maakt de herbruikbaarheid van informatie transacties inzichtelijk? *The actor-transaction-actor pattern in the DEMO infological model shows the reusability of information transactions?*
DM3	Het DEMO infologische model is een logische weergave van software services die moeten worden gerealiseerd? *The DEMO infological model is a representation of software services that must be constructed?*
DM4	Het DEMO infologische model is een logische weergave van functionele granulariteit van services? *The DEMO infological model is a functional representation of the granularity of services?*
DM5	Services in het DEMO infologische model zijn coarse-grained? *The DEMO infological model shows coarse-grained services?*
DM6	Het DEMO infologische model is vergelijkbaar met een informatie services model? *The DEMO infological model is equal with an information services model?*
DM7	Het basismodel ter optimalisatie van service granulariteit met behulp van DEMO is toepasbaar in alle organisaties? *The conceptual model for finding an optimum in service granularity is applicable for all organizations when using DEMO?*

Table 19: *GDSS propositions for the expert panel discussion*

Expert panel discussion	
DC1	Hoe zien jullie granulariteit van services in relatie tot software systeem ontwikkeling? *What is your opinion about service granularity in relation with information system development?*
DC2	Wat beschouwen jullie als optimale granulariteit van services? *What do you consider to be an optimum service granularity?*
DC3	Is er, zover bij jullie bekend, een methodische aanpak om tot een optimale granulariteit te komen? *Is a methodological approach in optimizing service granularity available?*
DC4	Hoe verhoudt granulariteit zich tot herbruikbaarheid en complexiteit van services? *What is your opinion about the relationship between granularity of services and the reusability and complexity of services?*
DC5	Hoe draagt een optimale granulariteit van services bij aan de flexibiliteit van bedrijfsprocessen? *What is your opinion about the contribution of optimum service granularity and the flexibility in business processes?*
DC6	Wat is jullie mening over de bijdrage van DEMO in het optimaliseren van service granulariteit? *What is your opinion about the contribution of DEMO in finding an optimum in service granularity?*
DC7	Wat is jullie mening over de uitdrukbaarheid van het informatie services model in DEMO? Is het eenvoudig leesbaar, begrijpbaar, toepasbaar, etc? *What is your opinion about the expressiveness of information services in the DEMO infological model? Is it easy to read, easy to understand, is it applicable, etc?*

P. Results GDSS

The results of the GDSS are illustrated in this chapter. Per main category the results are presented including any additional comments of the participants, a summary of the discussion with the participants and the underlying diagrams to substantiate the variability. The results are in Dutch.

Service granularity

The participants are asked to give their opinion about the propositions belonging to this main category.

Summarizing graph

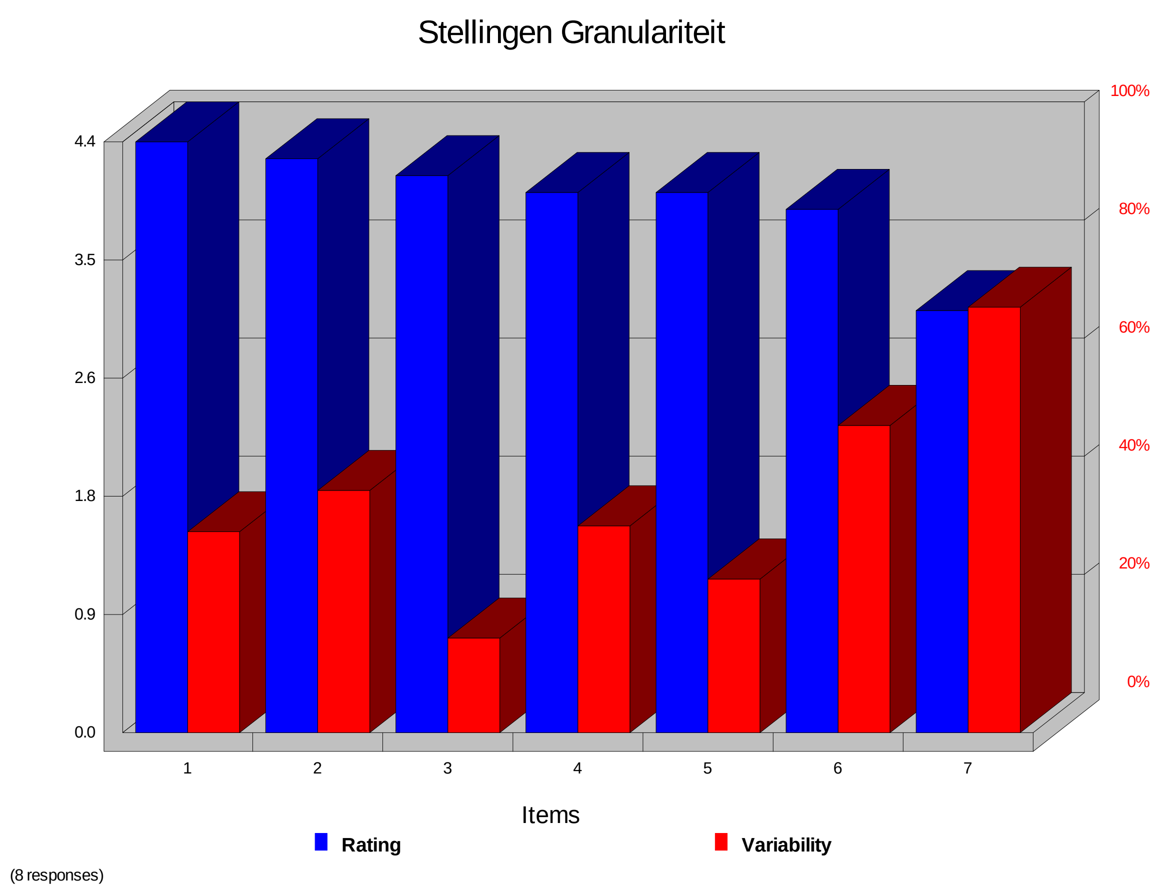


Figure 50: Summarizing graph of GDSS on service granularity

Results per proposition

The number between brackets [] corresponds with the item number in the summarizing graph. See Figure 50.

Table 20: *Summarizing table of GDSS on service granularity*

# Item	Average Rating	Variability
[1]. GR2. Een optimale granulariteit van services draagt bij aan de herbruikbaarheid van informatie objecten?	4.4	34%
[2]. GR5. Granulariteit van services is een generiek vraagstuk dat in iedere organisatie speelt?	4.3	41%
[3]. GR1. Granulariteit van services is een kritische succes factor in informatie systeem ontwikkeling?	4.1	16%
[4]. GR4. Een optimale granulariteit reduceert de complexiteit van de informatie huishouding?	4.0	35%
[5]. GR7. Informatie analyse moet sturend zijn in het optimaliseren van service granulariteit?	4.0	26%
[6]. GR3. Een optimale granulariteit van services draagt bij aan de flexibiliteit van bedrijfsprocessen?	3.9	52%
[7]. GR6. Optimale service granulariteit is in iedere organisatie anders?	3.1	72%

Remarks of the participants

The number before the remark corresponds with the provided rating.

GR2. Een optimale granulariteit van services draagt bij aan de herbruikbaarheid van informatie objecten?

3.0 Hangt het niveau van de definitie van de informatieobjecten niet samen met de granulariteit van de services? Ik vind deze vraag daardoor lastig te beantwoorden.

GR4. Een optimale granulariteit reduceert de complexiteit van de informatie huishouding?

3.0 Ligt eraan, met het principe van information hiding wordt complexiteit afgeschermd. Echter, op een lager niveau komt deze wel weer terug. Maar het 'hogere' niveau wordt er dan in ieder geval niet onnodig mee lastiggevallen.

GR7. Informatie analyse moet sturend zijn in het optimaliseren van service granulariteit?

AB Onthouding omdat me niet duidelijk is wat met informatieanalyse wordt bedoeld.

GR3. Een optimale granulariteit van services draagt bij aan de flexibiliteit van bedrijfsprocessen?

4.0 Ja, maar dan alleen wel flexibiliteit in IT-ondersteuning van bedrijfsprocessen.

GR6. Optimale service granulariteit is in iedere organisatie anders?

2.0 Niet vanuit ontologisch perspectief.Maar wel als er wordt ontwikkeld.

Discussion with the participants

The discussion with the participants started in the 36th minute and lasted until the 48th minute.

The discussion started by posting a question in which one participant suggested that the level of granularity is basically determined by the definition of information objects. The example that was given was about an insurance policy. Although the insurance policy is a reusable object in terms of business purposes, what does the reusability of information objects imply in this proposition? The response of another participant pointed at reasoning about information objects as statics or variables and the sense of their behavior. This means that something variable for the business can be something static in technology. Therefore, the question on reusability is hard to answer without mentioning the perspective how to look at it.

When evaluating proposition GR6 we concluded a big diversity among the participants. We saw that 50% agreed or totally agreed with the proposition, but that the other 50% disagreed or totally disagreed with the proposition. The discussion started by one participant who totally disagreed with the proposition. His opinion was that if the ontology is right, then there should not be any differences at all. He stated: *"Within a right ontology, the information services are all the same, independent of the type of organization."* The statement was substantiated by examples about the automotive branch. The products that are used in cars are uniformed in a sense. When the products are the same, then the processes must be equal. Another participant, who was totally agreeing with the proposition, responded that organizations are not constructed in a uniformed way. He said *"every organization is different, so the set of required services are different as well."* He pointed at the differences among organizations and their personal objectives to outperform on their competitors. The example was about a distribution channel that although two organizations make use of a distribution channel, their purpose of using is not the same if you compare a grocery with a lumberyard. One participant, who disagreed with the proposition, stated that if you look from an ontological perspective at two or more organizations in the same branch, they share their construction. He summarized that *"every organization has its own face, identity and strategic goals, but if you look at the construction of organization in the same branch, they have a lot in common."* After some more discussion the participants agreed that every organization has its own information service requirements, but they could be uniformed when you look branch specific.

Considering the opinions and the discussion on the topic of service granularity we conclude that the participants are relatively united. The participants recognize the benefit of an optimized service granularity in order to gain reusability in information objects or reduce the level of complexity, even if complexity is reduced by the utilization of information hiding via composition of services. The participants agree with a relative small variability. The participants also agree that service granularity is an important topic in information system design in every organization. This makes service granularity a universal topic in information system design, independent the type of organization. Remarkable is the outcome of proposition GR1. Without exception, all the participants conclude that service granularity is a critical success factor in information system development. The combination of proposition GR5 and GR1 ultimately concludes that the participants share the opinion of the universal importance of service granularity in information system development and the level of success that could be reached if service granularity has been optimized. According to the opinions of the participants regarding proposition GR7, when we leave out the abstinence of one participant, we can conclude that the information analyses process must be directive in defining the coarseness of granularity for services. This means that the information analyses process is responsible for gaining the benefits of an optimum service granularity.

Variability diagrams

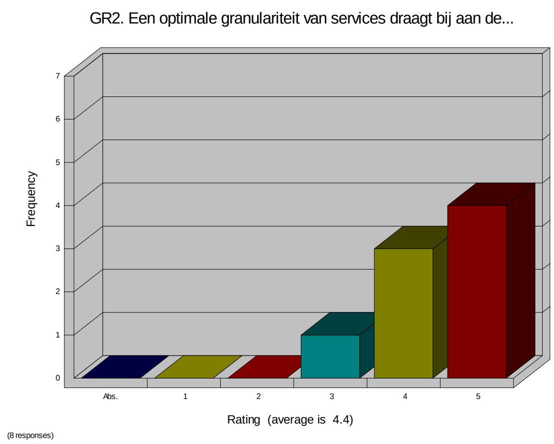


Figure 51: Variability diagram of GDSS GR2

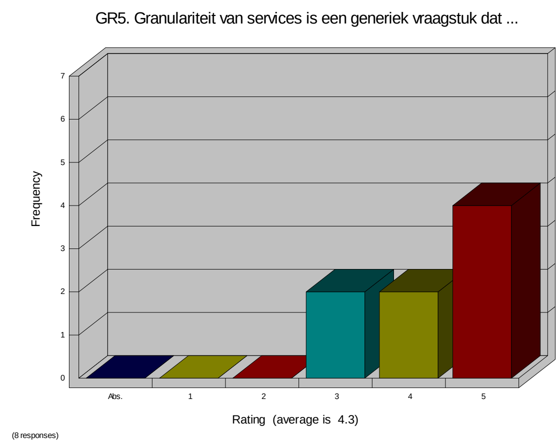


Figure 52: Variability diagram of GDSS GR5

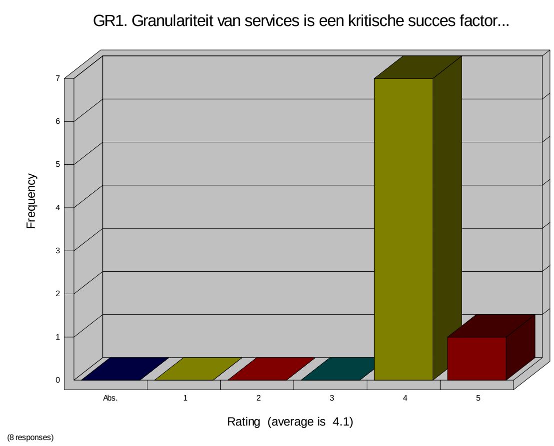


Figure 53: Variability diagram of GDSS GR1

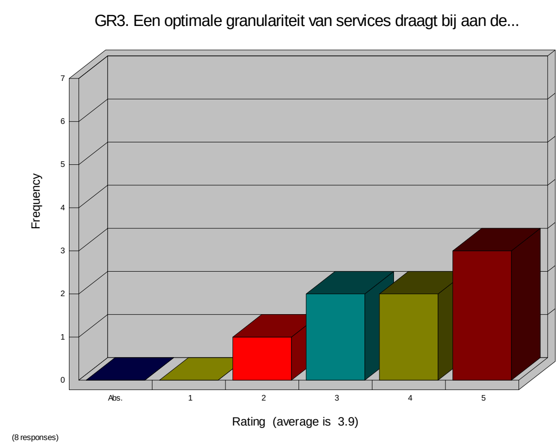
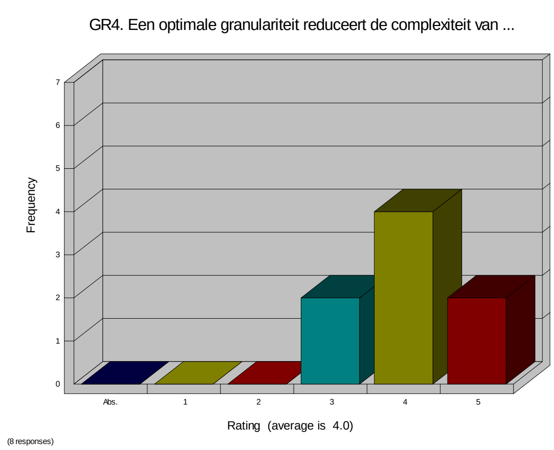


Figure 54: Variability diagram of GDSS GR3



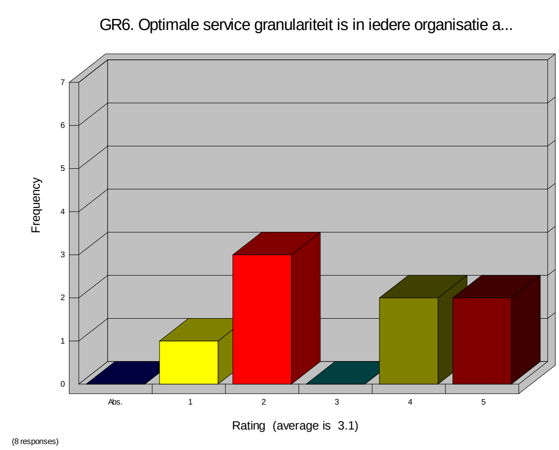


Figure 55: Variability diagram of GDSS GR4

Figure 56: Variability diagram of GDSS GR6

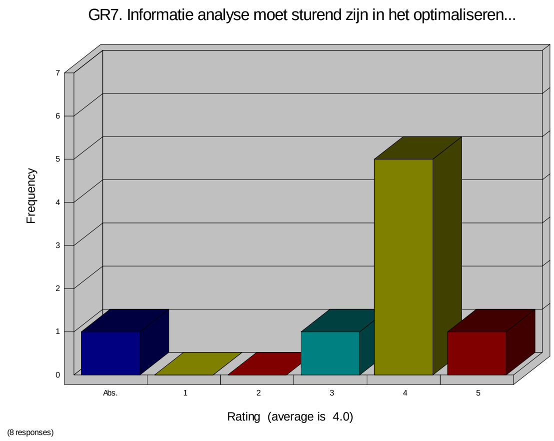


Figure 57: Variability diagram of GDSS GR7

The process of information analyses

The participants are asked to give their opinion about the propositions belonging to this main category.

Summarizing graph

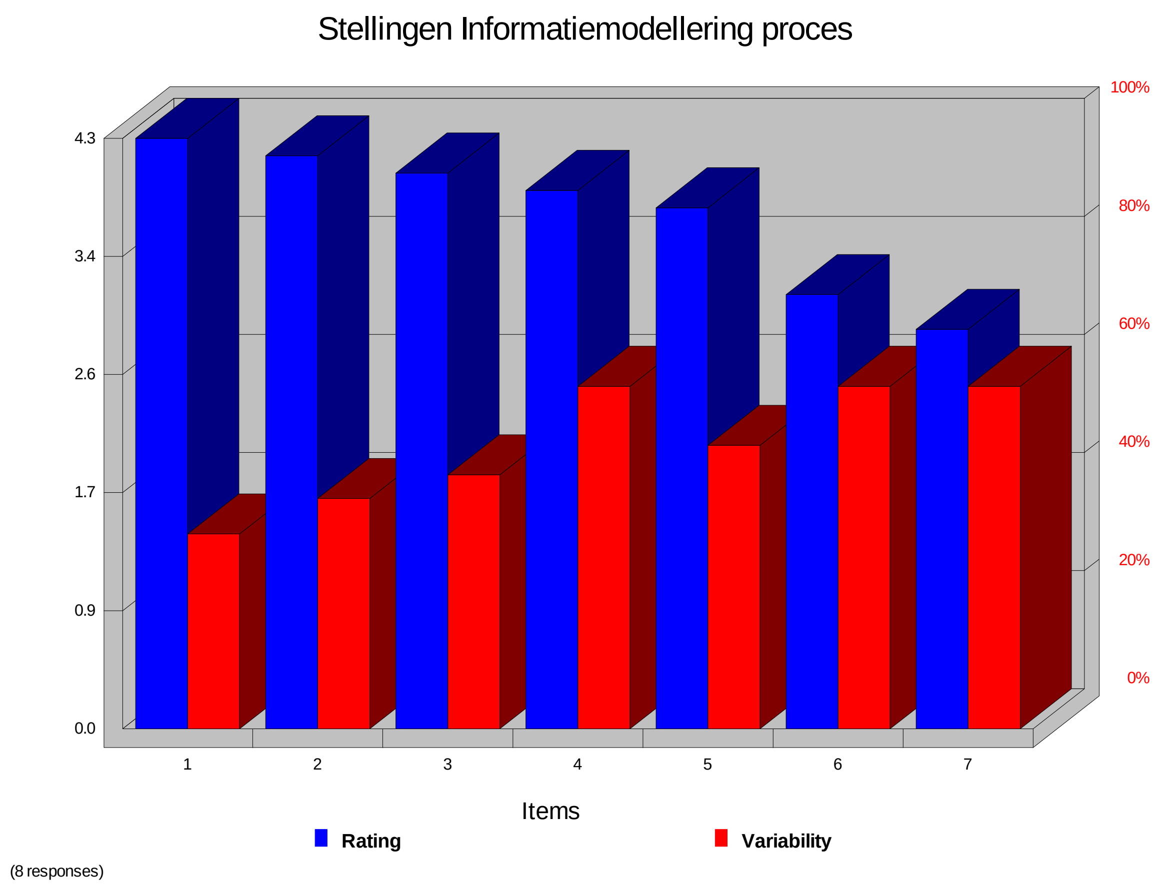


Figure 58: Summarizing graph of GDSS on the process of information analyses

Results per proposition

The number between brackets [] corresponds with the item number in the summarizing graph. See Figure 58.

Table 21: *Summarizing table of GDSS on the process of information analyses*

# Item	Average Rating	Variability
[1]. IM3. Het informatiemodel moet richtinggevend zijn voor software systeem ontwikkelaars?	4.3	33%
[2]. IM5. Informatie analyse kent een top-down aanpak, redenerend vanuit de business laag.	4.1	39%
[3]. IM1. Het GSDP is uitvoerbaar op het proces van informatiemodellering?	4.0	43%
[4]. IM7. Collaboratief modelleren vanuit een business perspectief en een IT perspectief leidt tot betere oplossingen.	3.9	58%
[5]. IM2. Het informatiemodel moet herkenbaar zijn voor de business die het ondersteund?	3.8	48%
[6]. IM6. Informatie analyse moet het bestaande systeemlandschap meewegen in voorgestelde informatiemodellen.	3.1	58%
[7]. IM4. Een optimaal granulair informatiemodel verhoogt de kwaliteit van software.	2.9	58%

Remarks of the participants

The number before the remark corresponds with the provided rating.

IM1. Het GSDP is uitvoerbaar op het proces van informatiemodellering?

5.0 Je modelleert vanuit de behoefte aan informatie in de business. Behoefte gespecificeerd in het action model.

IM7. Collaboratief modelleren vanuit een business perspectief en een IT perspectief leidt tot betere oplossingen?

1.0 Wat valt er te modelleren als je het business perspectief niet weet.

IM4. Een optimaal granulair informatiemodel verhoogt de kwaliteit van software?

3.0 Dat hoeft niet noodzakelijkerwijs. Je kunt toch knullige software maken, ook al is die netjes 'granulair' opgedeeld.

1.0 Helemaal afhankelijk van de kennis en competenties van de software ontwikkelaar.

Discussion with the participants

The discussion with the participants started in the 49th minute and lasted until the 66th minute.

Just before the GDSS session of this main category the discussion already opened when one participant asked about the definition of the information analyses process. When the process was explained with help of the GSDP the process became clear. Another participant got involved in the discussion and summarized that the information analyses process stands for the functional design of the I-organization. All participants agreed with that definition.

In the 55th minute, after the GDSS session was completed, we encountered a mixed set of opinions about proposition IM7. The spread in rating among the participants was large. The discussion started when one participant explained that he could not imagine what must be modeled, in a collaborate fashion, from an IT perspective? The feedback on that question was that IT specialists look from a different perspective to information collections if you compare them with business specialists. These two different points of views, which are both correct without a doubt, are united with the collaborative modeling approach we performed. Another participant summarized that the IT perspective in this proposition is not about the IT function, but the mix of competences that a person must have to do information analyses reasoning from a business background and/or and IT background. A participant, who agreed with the proposition, stated *"From an architectural point of view there should be principles. These principles must be a guideline in system development, but those principles also leaves room for the creative mind of the IT specialist who has to create the information system. Therefore I agree that when multiple disciplines are brought together, they come up with better results. It is good to look from an IT perspective to see what's handy for those guys."*

According to the rating for proposition IM6 there was a large spread in the opinions of the group. One participant, who disagreed with the proposition, said that *"The current information landscape is not relevant from an ontological perspective, but when looking at the implementation level then you must."* A participant who agreed with the proposition responded that after modeling you also want to implement something. So, when the objective of this information analyses process is about implementing software, then the existing information landscape must be part of the process.

When the facilitator summarized the rating for the proposition IM4 he asked the participants about their variability in opinions. Proposition IM4 is rated from totally disagree until totally agree and most of the participants responded undecided. One participant with an undecided opinion said *"An optimized granularity does not necessarily result in better software. One can still make awkward software. Optimized granularity transforms the mess only into a structured mess, but it still remains a mess"*. A participant who totally disagreed with the proposition stated *"The quality of software is totally dependent on the skill level of the software developer."* This statement triggered a participant to react that the level of clarity of the information model is directly linked to the quality of software. If the information model is not clear, then software developers make their own decisions. Based on his own experiences, these kinds of decisions are mostly not aligned with the business objectives. The overall conclusion on this proposition is that an optimized service granularity is not guaranteeing better software. The skill level of the software developer is a decisive factor.

Considering the opinions and the discussion on the topic of information analyses we conclude that the participants agree on the importance of the informational analyses. They recognize the directive nature of the informational analyses in information system development. Considering the remarks in the discussion on proposition IM4, the participants recognizes the correlation between a good and well structured information analyses and the realized software, based on their own experiences. About the quality of the realized software they are all on the same page, the skill level of the software developer is reflected in the quality of the software. This underlines the craftsmanship of the software developer. The participants also share the same opinion about the information analyses strategy. The top-down strategy is considered to be a good approach in the process of information analyses, in which the business remains involved during the process. After discussion almost all the participants agreed on the benefit of collaborative modeling, where business competences and IT competences are united in the informational analyses process. Uniting the competences minimizes the gap that is perceived between business and IT.

Variability diagrams

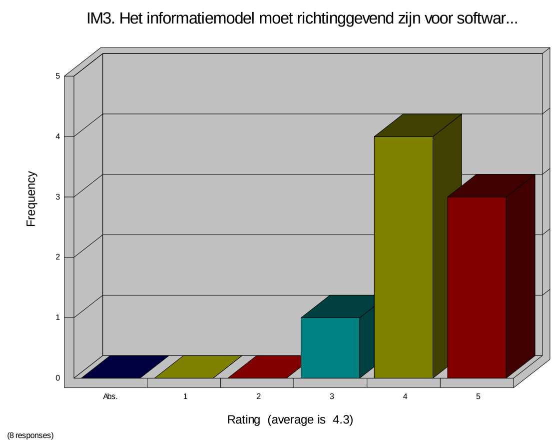


Figure 59: Variability diagram of GDSS IM3

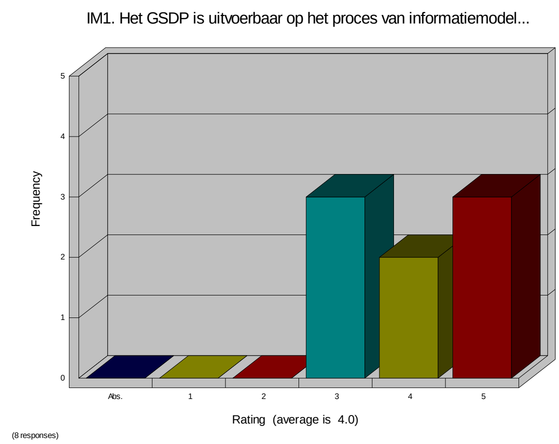


Figure 60: Variability diagram of GDSS IM1

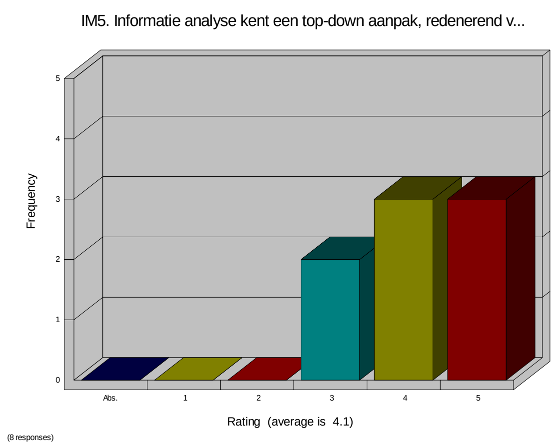


Figure 61: Variability diagram of GDSS IM5

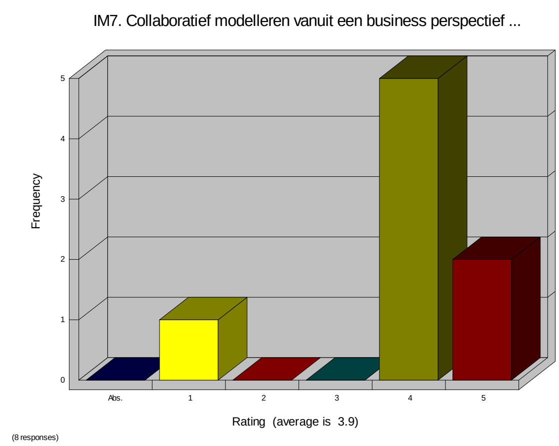
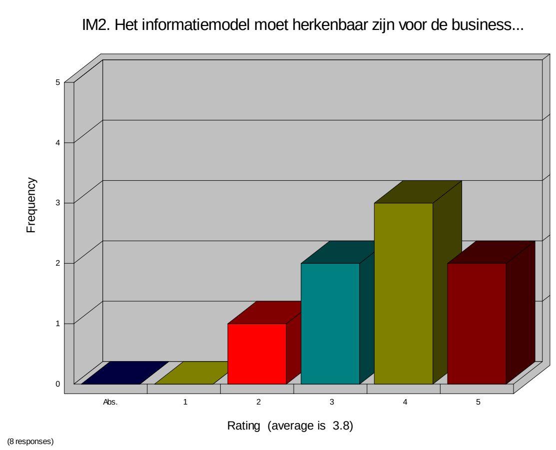


Figure 62: Variability diagram of GDSS IM7



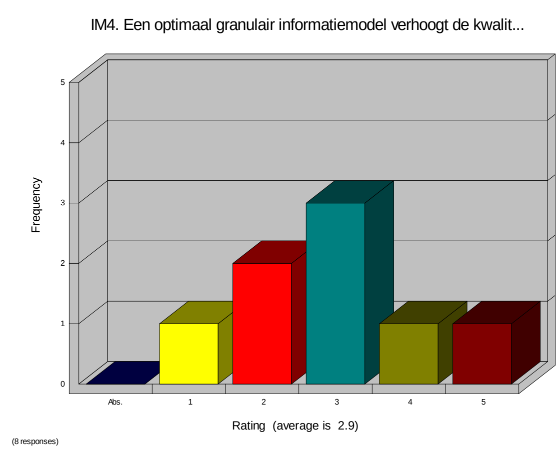


Figure 63: Variability diagram of GDSS IM2

Figure 64: Variability diagram of GDSS IM4

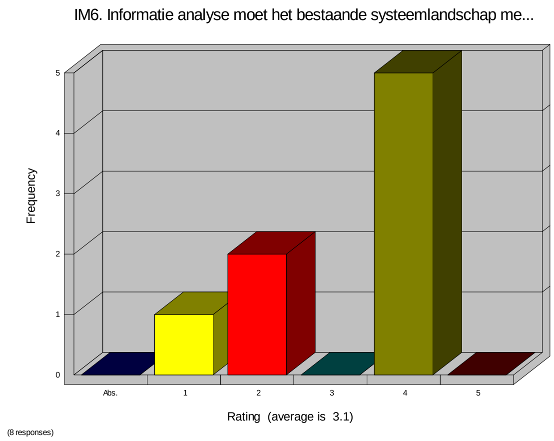


Figure 65: Variability diagram of GDSS IM6

Activities in the conceptual model

The participants are asked to give their opinion about the propositions belonging to this main category.

Summarizing graph

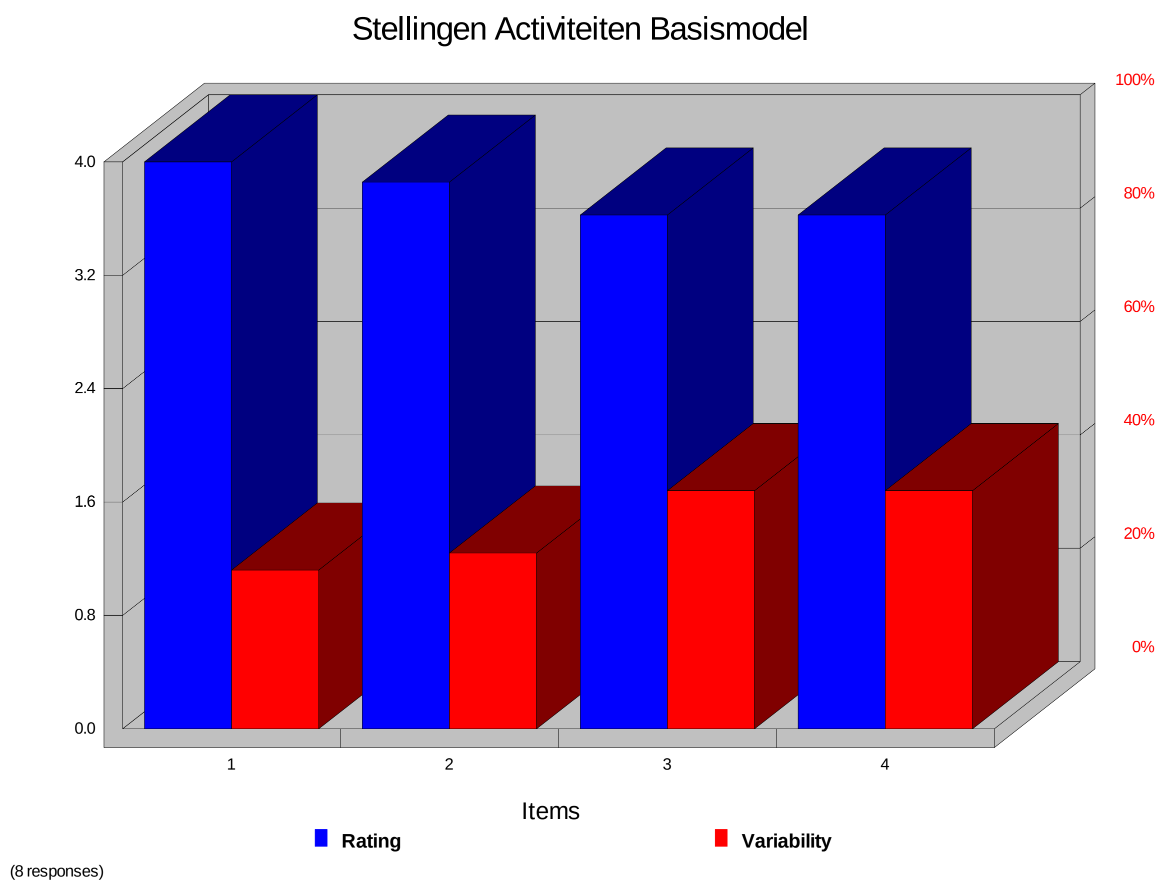


Figure 66: Summarizing graph of GDSS on the activities in the conceptual model

Results per proposition

The number between brackets [] corresponds with the item number in the summarizing graph. See Figure 66.

Table 22: *Summarizing table of GDSS on the activities of the conceptual model*

# Item	Average Rating	Variability
[1]. AB1. De activiteiten uit het basismodel zijn herkenbaar voor het optimaliseren van service granulariteit?	4.0	28%
[2]. AB3. De activiteiten uit het basismodel zijn samenhangend in het optimaliseren van service granulariteit?	3.9	31%
[3]. AB2. De activiteiten uit het basismodel zijn relevant voor het optimaliseren van service granulariteit?	3.6	42%
[4]. AB4. De activiteiten uit het basismodel dragen bij tot het optimaliseren van service granulariteit?	3.6	42%

Remarks of the participants

The number before the remark corresponds with the provided rating.

AB1. De activiteiten uit het basismodel zijn herkenbaar voor het optimaliseren van service granulariteit?

3.0 De meeste wel maar sommige niet (bijv. externalize I-actors).

AB Ik vind deze vraag te vaag om hier een goed antwoord op te geven.

AB3. De activiteiten uit het basismodel zijn samenhangend in het optimaliseren van service granulariteit?

4.0 Uiteraard, is van DEMO afgeleid en Enterprise Ontology is een samenhangende theorie. Geen willekeur zoals in bijv. BPMN.

AB2. De activiteiten uit het basismodel zijn relevant voor het optimaliseren van service granulariteit?

4.0 Ik snap alleen de use cases niet zo, kun je daarin ook weer geen verschil in granulariteit hebben?

3.0 De meeste wel.

3.0 Compose en Split Information Services zou ik samentrekken tot Define Information Services.

AB4. De activiteiten uit het basismodel dragen bij tot het optimaliseren van service granulariteit?

5.0 Na het doorlopen van het gehele proces is het voor mij duidelijk dat de elementen die op de slide staan nodig zijn om tot een goed eindresultaat te komen. Als je geen ervaring hebt is het moeilijk om deze vragen te beantwoorden.

4.0 Ja, maar ik mis link naar implementatie. Leuk om bepaalde services te bedenken, maar je moet ze ook kunnen maken. Vraag is weer of je alleen kijkt naar ontologisch niveau en het implementatieprobleem aan anderen overlaat of dat je dit laatste ook op wil lossen. In de praktijk heb je natuurlijk allebei nodig. Je hebt niet veel aan een SOA op papier.

2.0 Dragen bij aan optimalisering? Wel aan de granulariteit, maar ik weet niet of dat bijdraagt aan de optimalisering.

Discussion with the participants

The discussion with the participants started in the 72th minute and lasted until the 86th minute.

The discussion started when one participant responded on proposition AB2 that he does not see the specific purpose of the activities “Compose information services” and “Split information services” in relation with the activity “Define information services”. In his opinion services are defined in all three activities, but the decision to split or compose an information service is the process of reconsidering the information services model. From this point of view he stated that the activity is to define information services, and sub activities are composing or splitting services. The group of participants all agreed with this recommendation.

Another interesting remark on proposition AB2 was about the benefit of use cases. For the participant it was not clear how the application of use cases contribute in an optimized service granularity, because use cases itself could also be different in the coarseness of the functionality it describes. The researcher responded to this remark that the application of use cases were bounded by the defined information transaction, but that the information transaction itself was not always well specified enough for the technical solution architects to implement. The unit of functionality was not explicit enough when looking just at the information transactions. Therefore the use cases were necessary to specify the kind of requests that the information service would accept and underlying to the requests is the unit of functionality that an information transaction supports. Basically this is about specifying the interface of the information service and the describing the unit of functionality. One participant completed that the use cases must be extended with specifying the quality of information as well. He stated *“Quality of information is a part of* *the information analysis and must be completed before creating software. The quality of information is an important indicator to verify the form the granularity of services and prerequisite in developing software."* All the participants agreed on the benefit of use cases after this explanation.

In response to proposition AB4 one participant, who totally agreed with the proposition, stated that he shared the experience of the conducted improving case study. Therefore he was able to identify the activities as relevant, because he was involved during the process. He stated *"…for someone who did not participate in this process, where several iterations were necessary to come to this conceptual model, for those persons these activities could only be accepted if they were verified with own experiences."* The other participants agreed, and in essence this statement is inextricable bound up with doing scientific research.

One participant remarked that the link is missing to implementation in proposition AB4. The expert noticed that the viewpoint of the conceptual model is important. When you stay on the ontological level, then the implementation considerations will remain undefined and the conceptual model will be a procedure on paper without coding anything. Or must the conceptual model contribute to the implementation level. In that case something concrete has to be realized. The researcher replied that the objective of the conceptual model is to be directive towards the implementation level. Therefore the conceptual model captures activities that are assignable to analysts with business competences as well as analysts with technical competences. The DEMO methodology that is applied structured the information analyses process and by validating the information transactions on the determinants of service granularity we really optimized the information services model to be directive for the technical solution architect to design software on. However, the remark of the participant remains valid because during the improving case study no software has been developed. After the explanation the participant recommended to rephrase the remark into a question for further research. The researcher confirms this recommendation.

Considering the opinions and the discussion on the activities of the conceptual model we conclude that most of the participants agree with the propositions. In a total of 32 (thirty two) votes a participant rated three times in abstinence and seven times undecided. This results that 31% did not have an opinion about the conceptual model. Considering the statement on proposition AB4 this could be explained as being unfamiliar with the completed process of the improving case study. On the average level the participants agreed with the defined activities and they all recognized the contribution of the activities in relation with the objective of the improving case study.

Based on the shared remark on proposition AB2 of a participant the conceptual model will change the activities for defining information services, splitting information services and composing information services. The latter two must be considered as a way of modeling information services within the activity of defining them.

Variability diagrams

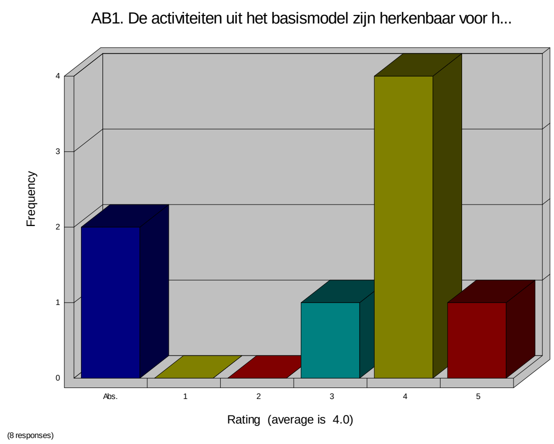


Figure 67: Variability diagram of GDSS AB1

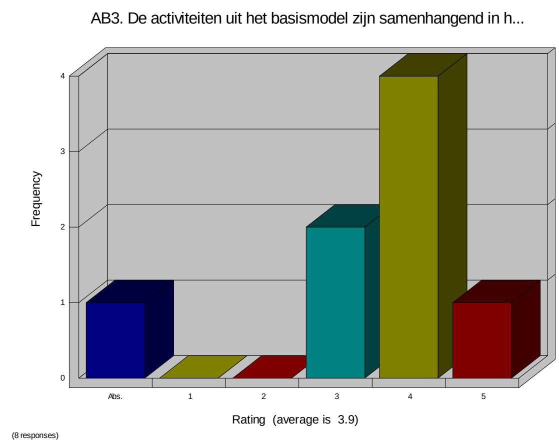


Figure 68: Variability diagram of GDSS AB3

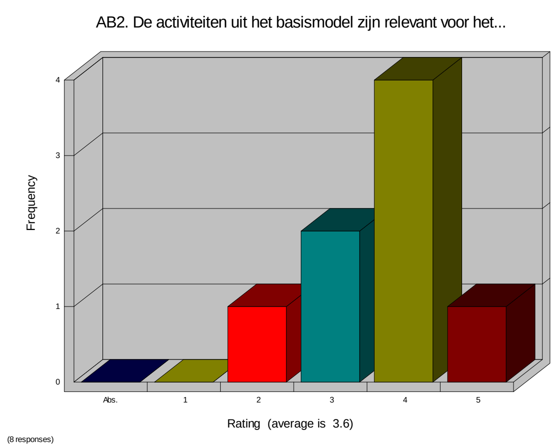


Figure 69: Variability diagram of GDSS AB2

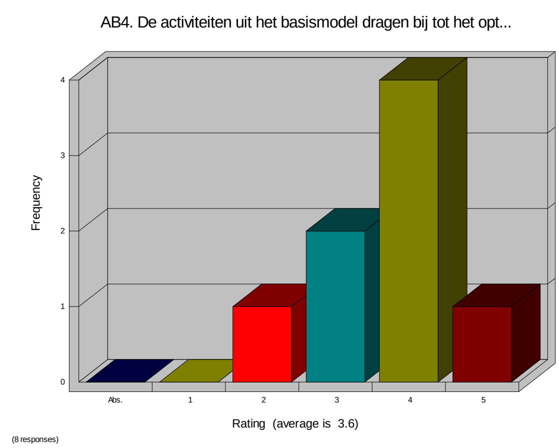


Figure 70: Variability diagram of GDSS AB4

The contribution of DEMO in finding an optimum in service granularity

The participants are asked to give their opinion about the propositions belonging to this main category.

Summarizing graph

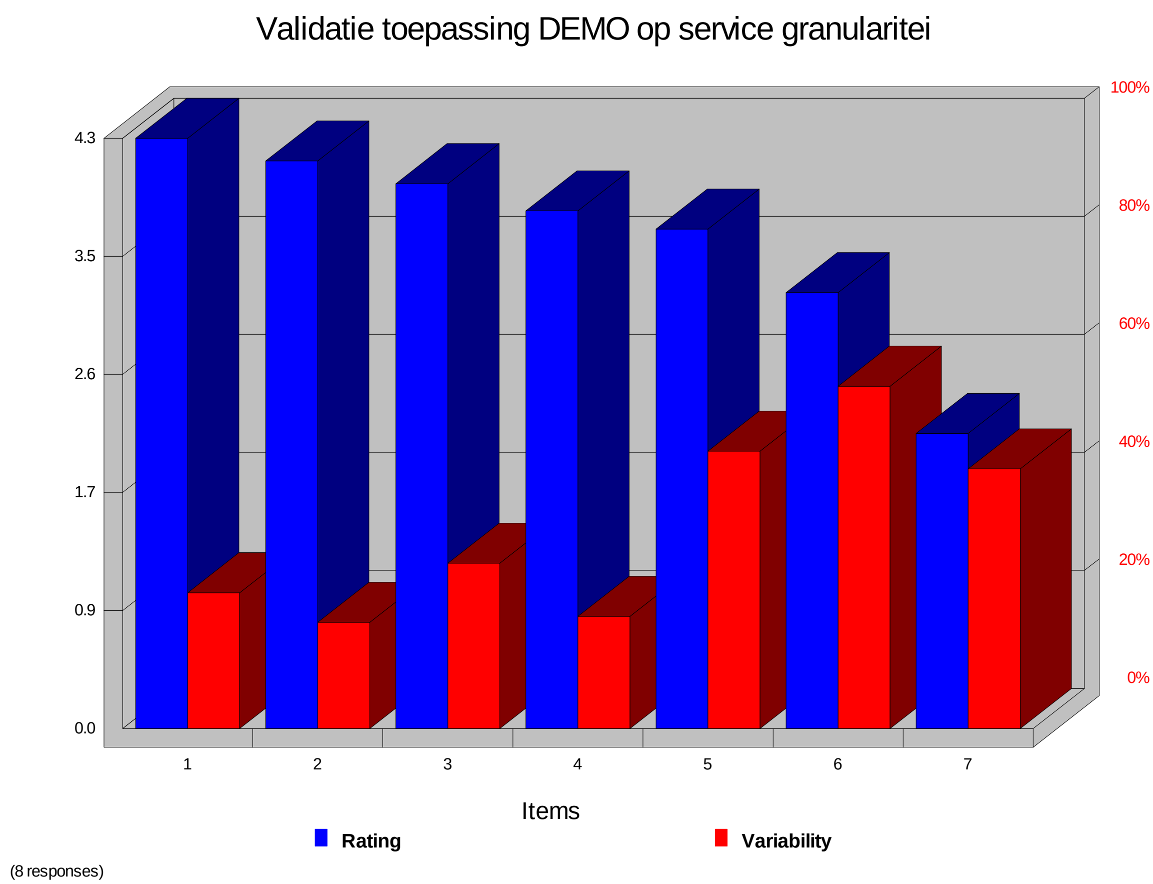


Figure 71: Summarizing graph of GDSS on the contribution of DEMO in finding an optimum in service granularity

Results per proposition

The number between brackets [] corresponds with the item number in the summarizing graph. See Figure 71.

Table 23: *Summarizing table of GDSS on the contribution of DEMO in finding an optimum in service granularity*

# Item	Average Rating	Variability
[1]. DM7. Het basismodel ter optimalisatie van service granulariteit met behulp van DEMO is toepasbaar in alle organisaties?	4.3	23%
[2]. DM2. Het actor-transactie-actor patroon in het DEMO infologische model maakt de herbruikbaarheid van informatie transacties inzichtelijk?	4.2	18%
[3]. DM6. Het DEMO infologische model kan dienen als een informatie services model?	4.0	28%
[4]. DM4. Het DEMO infologische model is een logische weergave van functionele granulariteit van services?	3.8	19%
[5]. DM3. Het DEMO infologische model is een logische weergave van software services die moeten worden gerealiseerd?	3.7	47%
[6]. DM5. Services in het DEMO infologische model zijn coarse-grained?	3.2	58%
[7]. DM1. Het DEMO infologische model bevat voldoende informatie om de ondersteunende software te maken?	2.2	44%

Remarks of the participants

The number before the remark corresponds with the provided rating.

DM4. Het DEMO infologische model is een logische weergave van functionele granulariteit van services?

AB Functionele granulariteit is een erg lastig begrip, vooral als je het baseert op een constructiemodel. Functie is subjectief.

DM3. Het DEMO infologische model is een logische weergave van software services die

moeten worden gerealiseerd?

4.0 Alleen als de granulariteit is geoptimaliseerd.

2.0 Waarom software? wordt per definitie alles in de I-laag geautomatiseerd?

DM5. Services in het DEMO infologische model zijn coarse-grained?

1.0 Fijner hoeft toch (blijkbaar) niet?

DM1. Het DEMO infologische model bevat voldoende informatie om de ondersteunende software te maken?

3.0 Er is natuurlijk nog steeds een hoeveelheid ontwerpvrijheid over. Dus, het is wel nodig maar niet voldoende.

Discussion with the participants

The discussion with the participants started in the 87th minute and lasted until the 104th minute.

Just before the GDSS session one participant already opened the discussion by asking for explanation about proposition DM4. In his perception the DEMO infological model is a construction model and could never be a functional representation of something. In essence the infological model expresses the construction of information and not the function. He stated that *"an ontological model is abstracted from implementation. This counts for the business layer as well as the information layer as well as the data layer. This means that an ontological model is not an implementation model."* The researcher responded that the purpose of the infological model is to express the construction of the information, but with the captured experiences and data from the improving case study we have concluded that the infological model itself is still too abstract for an implementation in software. Therefore, the proposition is formulated as it is. Considering the experiences from the improving case study, would you still see the infological model as construction model to create software from or must it be used as a logical layer on top of the software development layer. The participant understood the purpose of the proposition, but he found that the proposition was not well formulated. He recommended rephrasing the proposition into *"The infological model of DEMO is contains enough information for software development?"* All participants agreed on rephrasing the proposition, which has been implemented in the GDSS by the facilitator. The GDSS session now uses the rephrased proposition.

In the 95th minute the discussion opened about proposition DM5. Considering the ratings of the participants 50% of them rated in abstinence and undecided. The researcher asked why this significant amount of participants could not formulate an opinion. All the participants who chose one of these options responded that they did not have the theoretical knowledge about coarseness of services. They all found the term "coarse grained" too complex or one did not know what it means. One participant responded that considering the methodology and how information transactions are derived from the business model that information transactions are by definition in the right granularity. He stated *"Apparently finer grained* *services are not necessary."* A participant asked what about if the information service is too complex to realize? In that case one can decide not to offer the service as designed, but to adjust it in a sense it can fit within the possibilities of realizing the service, during software development. The participant who opened the discussion repeated his statement from before the GDSS, the infological construction model is not an implementation model. It is the best fit at ontological level, because it provides precisely the information that is required by the business layer. He said *"… for a business actor it is irrelevant how the service is implemented, as long as the actor has disposal of the information."*

The variability in rating proposition DM3 is relatively high. One participant who disagreed with the proposition responded that the infological model is not reflecting implementation. So, when the proposition refers to software services, then there already has been made a choice about implementation. This is not always the case, because one can also decide to support a service manually. The essence of the infological model is not to reflect implementation choices, but to remain on an abstract level, identifying precisely the kind of information that is required by the business organization.

Considering the rating of proposition DM4 50% of the participants rated in abstinence and undecided. The participants responded that the proposition was ambiguously defined. If the proposition mentions the DEMO infological model, then it is about construction. In that case the infological model could never reflect a functional model of something, because it is a construction model.

Considering the opinions and the discussion on the contribution of DEMO in finding an optimum service granularity we conclude that although the DEMO infological model can be seen as a model with information services, substantiated by the rating of proposition DM6, the infological model is not meant to be an implementation model. All the propositions in this main category have a relatively high number of participants who rated in abstinence and undecided. According to the discussions this could be declared by the complexity of the subject. The participants agree on DEMO being applicable to face the quest on service granularity in all organizations by rating proposition DM7 and according to proposition DM2 DEMO makes reusability of services explicit. The participants share the opinion that the DEMO infological model is not suitable for an implementation in software. The ratings of proposition DM1 support this conclusion. Apparently the DEMO infological model does not contain enough information to be directive for software development. This makes the infological model a logical model of the construction of information services. This conclusion is supported by the rating of proposition DM3 with respect to the discussion about it. In case the proposition did not mention software services but services, then the rating of the participants would be aligned more.

Variability diagrams

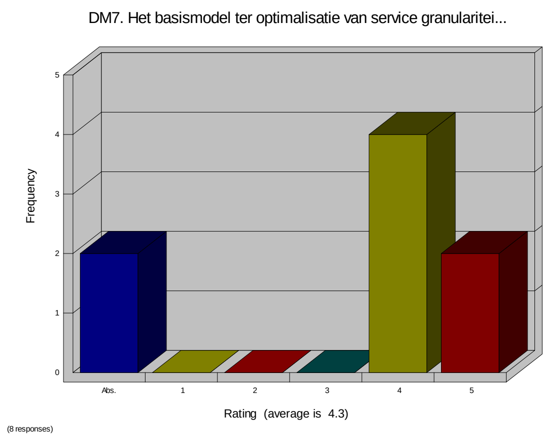


Figure 72: Variability diagram of GDSS DM7

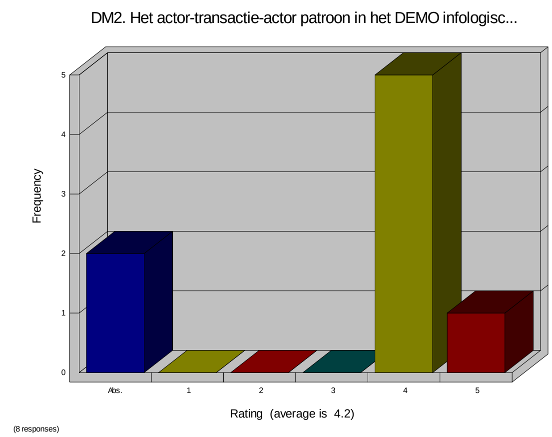


Figure 73: Variability diagram of GDSS DM2

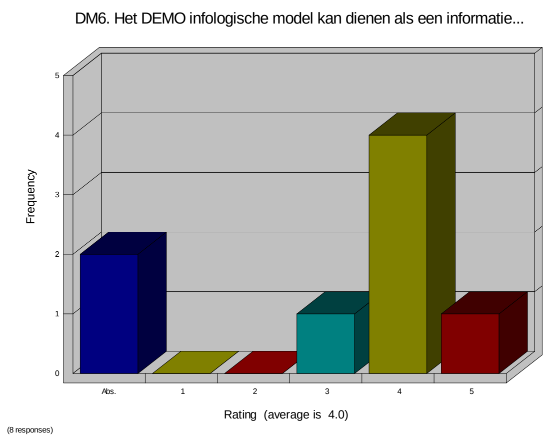


Figure 74: Variability diagram of GDSS DM6

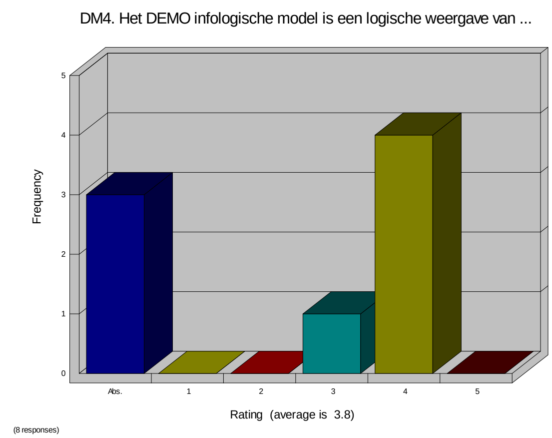


Figure 75: Variability diagram of GDSS DM4

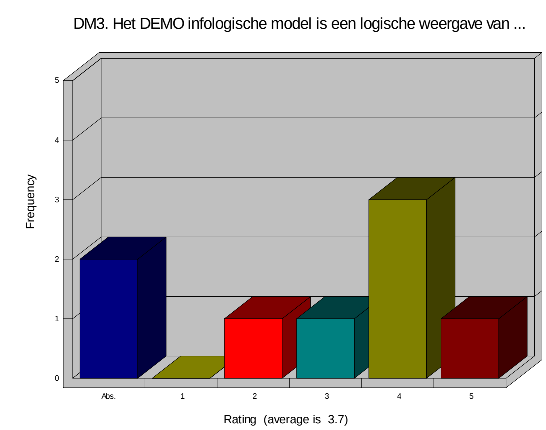


Figure 76: Variability diagram of GDSS DM3

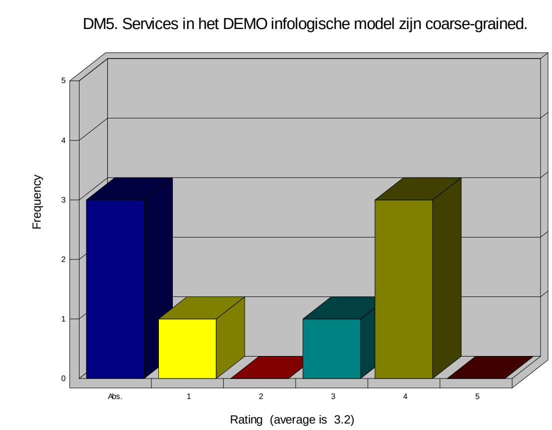


Figure 77: Variability diagram of GDSS DM5

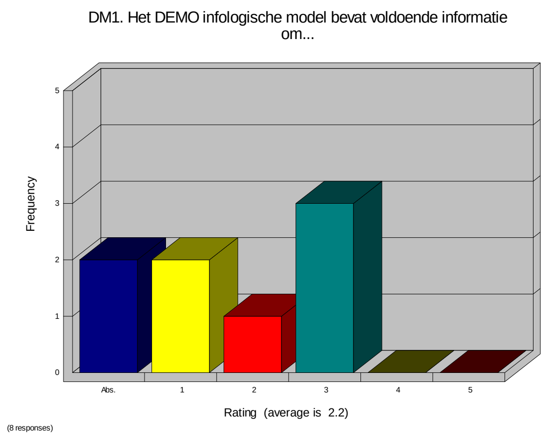


Figure 78: Variability diagram of GDSS DM1

Expert Panel Discussion

The participants are asked to give their opinion about the propositions belonging to this main category. The expert panel discussion is a trace of statements and opinions given by the participants of the GDSS. Per proposition the trace log is shown. The purpose of the characters in the left column is showing the trace sequence.

DC1. Hoe zien jullie granulariteit van services in relatie tot software systeem ontwikkeling?

A Sorry, dit gaat me boven de pet.

B Door gebruik te maken van het finetunen van services wordt het maken van software gemakkelijker gemaakt. Maar het is niet de toverdoos!!!!!

C Het kan zijn dat de ontologische granulariteit te grof is (of te fijn?) en dat je in de software een andere granulariteit kiest.

D We moeten weten welke behoefte er is vanuit de bedrijfsprocessen aan informatie. Deze informatiebehoefte wordt vertaald naar software services. Bij voorkeur zijn software services zoveel mogelijk herbruikbaar.

E Het heeft m.i. te maken met het bouwen van modulaire systemen. De hoofdlijnen worden vastgelegd in het infologisch model. De invulling vindt plaast door softwareontwikkeling.

F Optimale granulariteit van services kan leiden tot betere generieke/specifieke sercvices, wat weer kan leiden tot bv hergebruik tijdens de softwareontwikkeling.

G De klant bepaald de behoefte aan services en hier aan gerelateerd de ontwikkelde software.

DC2. Wat beschouwen jullie als optimale granulariteit van services?

A Sorry.

B Zoals ik het zie heb je een direct aan de B-organisatie gerelateerde granulariteit, die dus optimaal is als B-service, en een uit software maak- en onderhoudstandpunt te kiezen granulariteit.

C Continu blijven denken in begrippen als flexibilteit, onderhoudbaarheid, herbruikbaarheid van systemen.

D Optimaal is relatief. Zoals eerder gezegd kan het ontologisch hetzelfde zijn, maar implementatie-technisch kan het per organisatie-type valsnog verschillen.

E Een granulariteit die de actor biedt wat hij nodig heeft en ook te realiseren is in software. Moeilijk om hier harde uitspraken over te doen.

F Zodanig elementair (atomair), dat er verschillende composities (orkestraties, choreografieen) mee kunnen worden gemaakt afhankelijk van de behoefte vanuit de business.

G KI (Kunstmatige intelligentie)

H Durf hier geen uitspraak over te doen.

I De juiste vertaalslag van de klantwens/vraag.

DC3. Is er, zover bij jullie bekend, een methodische aanpak om tot een optimale granulariteit te komen?

A Nee.

B Ik denk het niet.

C Nee.

D Alleen voor het hoogste (= B-) niveau; daar zijn het de I-transacties.

E Geen idee.

F Nee.

G Weet niet.

H Er zijn verschillende SOA-aanpakken die hier uitspraken over proberen te doen. Ligt eraan of je het eens bent van definitie van 'optimaal' van de bedenkers van de methode (als ze een dergelijke definitie al geven). Ik ben nog niet echt overtuigd.

DC4. Hoe verhoudt granulariteit zich tot herbruikbaarheid en complexiteit van services?

A Optimale granulartiteit leidt tot hergebruik.

B Het moet de herbruikbaarheid vergroten en de complexiteit verkleinen.

C Geen idee nog geen praktijk ervaring.

D Lastig. Grove granulariteit zal in het algemeen tot meer complexiteit leiden. Herbruikbaarheid is een heel ander onderwerp.

E Optimale granulariteit kan leiden tot herbruikbaarheid van services, deze generieker maken, en daardoor complexiteit omlaag brengen.

F Herbruikbaarheid van services is belangrijk. Echter hoe "kleiner" je de services maakt, hoe groter de herbruikbaarheid is. Loosely coupling is belangrijk om complexiteit te beheersen.

G Fine-grained services zijn heel herbruikbaar, coarse-grained in principe minder (kun je ook nog wat discussie over voeren overigens). Coarse-grained services zijn complexer om te maken (worden ook vaak door meerdere softwaresystemen geleverd m.b.v. een composite service).

DC5. Hoe draagt een optimale granulariteit van services bij aan de flexibiliteit van bedrijfsprocessen?

A Maakt het makkelijker met herinrichten van een organisatie a.g.v. een reorganisatie.

B Makkerlijker te wijzigen als niet alles opnieuw hoeft te worden gebouwd, maar alleen anders hoeft te worden samengesteld.

C Hier kan ik niets over zeggen theoretisch wel maar er is nog geen praktijk ervaring (bij mij).

D Verkleind de flexibiliteit doordat de samenhang van diverse services niet zichtbaar zijn.

E Als de granulariteit een-op-een is met de I-transacties, beperkt dat niet de verandering van de bedrijfsprocessen.

F Het zorgt voor flexibiliteit op IT-gebied, m.a.w. ondersteunende IT kan gemakkelijker aangepast worden aan veranderde bedrijfsprocessen. Er zijn alleen veel andere oorzaken die flexibiliteit in bedrijfsprocessen mogelijk tegenhouden. Flexibele IT biedt geen garantie op flexibele bedrijfsprocessen.

G Wat versta je onder flexibiliteit van bedrijfsprocessen? Pas als de bedrijfsprocessen moeten worden aangepast (dus ook wijzigingen in actor rollen moeten worden aangebracht) zal granulariteit belangrijk worden.

H Deze maken de services breder toepasbaar/ inzetbaar, en dus breder inzetbaar in meerdere org-onderdelen.

I Geen idee.

DC6. Wat is jullie mening over de bijdrage van DEMO in het optimaliseren van service granulariteit?

A Na dit beoefend te hebben geeft DEMO een goede degelijke invulling om dit helder te krijgen. Alleen moet de tool beter worden.

B Helpt.

C Sorry.

D Geen ervaring mee.

E Kan zeer zeker bijdragen.

F DEMO biedt je het 'hoogste' niveau, dus datgene dat voor de business optimaal is.

G DEMO zorgt voor een goed gedefinieerd bedrijfsmodel dat als basis gebruikt kan worden.

H DEMO zorgt inderdaad uitsluitend voor een goed gedefinieerd bedrijfsmodel dat als basis dient.

I Vanuit ontologisch naar infologisch kan het een bijdrage leveren.

DC7. Wat is jullie mening over de uitdrukbaarheid van het informatie services model in DEMO? Is het eenvoudig leesbaar, begrijpbaar, toepasbaar, etc?

A Zie antwoord vraag 6 de tooling moet verbeterd worden om dit proces te ondersteunen.

B Te weinig gezien van het model.

C Zelf geen ervaring, is wel leesbaar en begrijpbaar (wel achtergrond info nodig)

D Sorry.

E Dat hangt af van de bekendheid met de methode, natuurlijk. Voor een DEMO Prof is het allemaal helder (?)

F Ik vind weergeven van een shaping m.b.v. een transactiesymbool heel verwarrend. Zou hier absoluut een andere symbool voor introduceren. Verder in principe wel OK, maar ik denk wel dat het heel snel heel groot wordt. Wat is jullie ervaring hier tot nu toe mee?

G Geen mening.

H Eens met opmerking 6!

I Zeker bruikbaar, echter zonder toelichting voor buitenstaander niet altijd even leesbaar.

Q. IIMA Article

This appendix captures the full article that has been posted for the International Information Management Association Conference in Utrecht, the Netherlands in October 2010. This article has been written according to the conducted explorative case studies.

A METHOD FOR DEFINING OPTIMUM SERVICE GRANULARITY

René Wiersma, Master of Informatics, HU University of Applied Sciences, Utrecht, The Netherlands, rene.wiersma@student.hu.nl and Eridium Consultancy, Rotterdam, The Netherlands, rene.wiersma@eridium.nl

Pascal Ravesteyn, Center for People, Process & Technology, HU University of Applied Sciences, Utrecht, The Netherlands, pascal.ravesteijn@hu.nl

Abstract

In the past 10 years a new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox and Kreger, 2005). However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. While Steghuis (2006) and others state the importance of optimum service granularity our research tries to answer the HOW question by using the Design & Engineering Methodology for Organizations (DEMO) (Dietz, 2006). Our research consists of a literature study and the execution of two essential case studies at organizations in the Netherlands. Validated by interviews that were held at the organizations the case studies learned that the applied methodology is directive in obtaining a functional optimum in service granularity for business services and information services.

Keywords: Web services, Granularity, SOA, DEMO

Introduction

In the past 10 years an important change has occurred in how to develop, integrate and reuse information systems. A new paradigm called Service Oriented Architecture (SOA) has emerged that is based on the development, deployment and reuse of (web) services which can easily be assembled in different ways allowing organizations to quickly adapt to changing business needs (Cox and Kreger, 2005). The SOA concept benefits business because IT can be developed and implemented much faster and at lower development costs. Also it makes organizations processes more adaptable to change. Even the definition of Service-Oriented Architecture points at the common benefits of flexibility and reusability. Regarding the aspect of flexibility Weske (2007) refers to the definition of Burbeck (2000) that services gain flexibility by runtime coupling with the service registry. This dynamic coupling of services is not reached most of the times. Contrary to the definition of Burbeck (2000) Weske (2007) states better to speak about enterprise services. Reuse is reached if a service contains functionality with a clear business value and can be used directly (Weske, 2007).

Besides this SOA also requires that organizations evaluate their business models to fit service-oriented analysis and design techniques, deployment and support plans, and carefully evaluate partner, customer, and supplier relationships (Papazoglou and van den Heuvel, 2006). However while SOA has a large potential for business one of the most complex issues in any SOA project is to define the right granularity of the services. The quest for the right granularity is complex if the promises of flexibility and reusability must be obtained. Most of the times a senior technical specialist answers this question on gut feeling. A method for service definition is important in environments where a lot of services are available and the set of available services changes over time. Burbeck (2000) states services to be grounded on shared organizational principles. These principles makes sure services operate without errors, support flexibility and can be joined together to fit in business processes.

The question how services should share organizational principles or how services should be modeled to obtain flexibility and be able to adopt organizational changes is not answered yet. Steghuis (2006) states that the quest for service granularity has been addressed in many articles (De Jong et al, 2010, Foody, 2005; Papazoglou et al., 2006; Sims, 2005; Feuerlicht et al., 2007; Rosen, 2007) but none of these sources answers the question of how to define service granularity properly, neither do they provide some kind of concrete guidelines. Most sources just mention the importance of finding a right service granularity.

While Steghuis (2006) and others state the importance of optimum service granularity our research tries to answer the HOW. In this paper we show that with the use of the Design & Engineering Methodology for Organizations (DEMO) (Dietz, 2006) a guideline can be found for finding an optimum service granularity on the business, informational and data levels of organizations.

In the remainder of the paper we will first describe our research method, and then in section 3 we explain the concept of service granularity. Section 4 briefly describes the DEMO method while in section 5 the use of this method to define service granularity is shown. The used approach is validated via case studies in section 6 and conclusions are given in section 7.

Research Method

As a starting point in developing a guideline for finding an optimum service granularity we chose the Information System Research Framework of Hevner, March, Park and Ram (2004). This is based on the fact that Hevner et al. (2004) propagate that studies in the IS research domain contain both descriptive and prescriptive research.

The descriptive part of the research (knowledge-producing activity) aims to understand and explain how service granularity is defined, while the prescriptive approach (knowledge-using activity) aims at improving service oriented architecture (March and Smith, 1995; Hevner et al., 2004).

The research consists of two major activities based on the framework. First a literature study of existing research was conducted (the knowledge base). Based upon this study it was decided to use the DEMO method as a foundation to develop our guideline on defining service granularity.

Secondly by using and extending the DEMO method in two case studies the guidelines on how to define optimum service granularity were developed. In the framework of Hevner et al. (2004) these activities are related to the ‘environment’ and ‘develop/build’ aspects.

Guided by the case study structure of Bryman and Bell (2007) two essential case studies have been executed at organizations. One case study has been executed at Pretium Telecom in Haarlem, the Netherlands. Pretium Telecom provides telecom services for about one hundred and fifty thousand customers in the Netherlands. The second case study is about a fictitious value chain as part of an assignment at the HU University of Applied Sciences in Utrecht, the Netherlands.

At Pretium Telecom interviews were conducted with the Chief Information Officer and the lead technical architect. The interview questions were based on the Enterprise Engineering Framework of Op ‘t Land (2008). The framework structured the interview to cover both the business context and the technical aspect of software development. Based on the interview results a DEMO organizational construction diagram is created. To validate this model it is discussed with participants of the interviews.

The second case study is grounded in the architectural description as part of the assignment. The architectural description followed the Enterprise Engineering Framework structure

What is Service Granularity

Granularity is a term that reflects the degree of modularity of a system. Papazoglou et al. (2006) define service granularity as the unit of modularity of services. It is the amount of functionality that is exposed by a service. There exist two types of granularity of services. Fine grained services typically implement a single atomic operation and exchange limited amounts of data. Coarse grained services implement high-level business functions (Feuerlicht and Wijayaweera, 2007).

Steghuis’ (2006) research concluded that several types of granularity exist for different kind of services. Several authors are referring to several types of services. Papazoglou et al. (2006) uses the service types Business Services, Infrastructure Services, and Component Based Service Realisations. Erl (2005) uses the service types Orchestration Services, Business Services, and Application Services. Schekkerman (2004) describes the Business of Organization Services, Information (System) Services, and Technology Infrastructure Services, which is part of the Capgemini Integrated Architecture Framework. This paper uses the service types Business Services, Information Services, and Data Services. The definition of these service types are aligned with the Integrated Architecture Framework of Capgemini and also with the Enterprise Engineering Framework (Op ‘t Land, 2008), and it is aligned with the three homogenous layers of the organization theory (Dietz, 2006).

Based on literature study Steghuis (2006) made a classification of the aspects of granularity and drew it on the several service types. This paper adjusted the terminology of service types used by Steghuis. This paper uses the service types Information Service and Data Service respective to the service types Information System Service and Software Service as used by Steghuis. Table 1 shows the classification of aspects of service granularity grouped by service type.

Business Service	**Information Service**	**Data Service**
Functionality	Functionality	Functionality
Flexibility in Business processes	Flexibility in Business processes	
Problem Complexity	Cognitive and Structural Complexity	Cognitive and Structural Complexity

Reusability	Reusability	Reusability
Composability	Composability	Composability
	Reusability of Legacy	
Sourcing	Sourcing	
Genericity	Genericity	Genericity
Context-independence	Context-independence	Context-independence
	Performance	Performance

Table 1 – Classification of service granularity aspects grouped by service type (Steghuis, 2006)

The relation with DEMO is substantiated by the foundation of the organization theory. The organization theory exists of three homogenous systems: the B-organization (for Business); the I-organization (for Intellect); and the D-organization (for Document) (Bunge, 1979), applied by Dietz (2006). This concludes that the B-organization provides business services, the I-organization provides informational services and the D-organization provides document or data services.

The results of the two case studies are being tested against the aspects of service granularity. In preparation the ontological models of the organizations are modeled. An ontological model on business level gives overview of the essential business activities. These activities are named as transactions between actors. A transaction is defined by the execution of production activities and coordination activities within a generic social pattern (Dietz, 2006). The ontological model is the organizational construction diagram and draws the organization abstracted from its implementation.

The DEMO methodology is briefly explained in the next paragraph.

What is DEMO?

DEMO is a method for ontology based enterprise modelling. An ontological system is defined as the cohesion of the composition, the structure, and the environment of the system (Dietz, 2006). DEMO has proven to be an effective method for decomposing organizations. The method is based on the Ψ-theory. The Greek letter is pronounced like PSI and stands for Performance in Social Interaction. This is the basic paradigm of the theory and is about the performance of the organization related with the social interaction of the organization or other systems. DEMO is grounded on three organizational layers from the organization theorem, called the B-organization that performs ontological transactions, the I-organization that performs infological transactions, and the D-organization that performs datalogical transactions. Figure 1 expresses the organizational theorem.

The business organization is the essential organizational layer that communicates and produces facts to realize business results. This organizational layer exists out of actors who are producing unique and definitive facts. On this organizational layer the actors interact with other social entities (actors) in the system. For example in our case study at Pretium Telecom an actor is the contract manager who requests the bank (which also is an actor) to proceed with the automatic money transfers for the current period. The activities between these two actors within the system generate new facts. These new facts are the transfer of money from one account to another account.

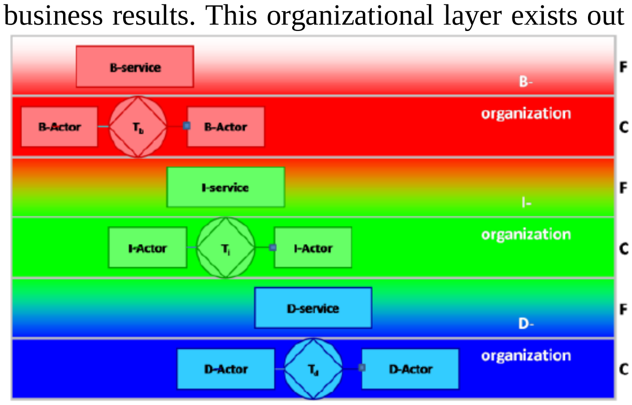


Figure 1: The organizational layers of the organization theorem

With the informational organization layer we mean the information processing layer. Information processing is the calculation of data via algorithms into other data and presents the information to the business layer. Within the informational organization of Pretium Telecom an information transaction exists that calculates an invoice amount based on the usage figures, the rates, contract data and elapsed time.

The data organization layer has the responsibility of storing, copying, searching, changing and removing of data. The transactions on the data layer are not familiar with information, but only with the raw data itself. The data organization layer is only allowed to communicate with the informational layer. This means that the data layer provides the data that is required for an informational process and stores the data afterwards. One of the data transactions at Pretium Telecom is the storage of contracts in a database table.

Defining Service Granularity using DEMO

The Cambrigde University Press dictionary (2009) translates optimum in “most likely to bring success or advantage”. The process for defining the optimum service granularity starts on the business layer of the organization by creating an organizational construction diagram. Every transaction on this layer is part of the essential business where only unique facts are produced. For each case study we conducted the organizational construction diagram. This paper elaborates only on the case study at Pretium Telecom.

Business Services

A part of the organizational construction diagram of Pretium Telecom in figure 2 consists of three transactions. These are T01, T02 and T03. Every transaction serves a unique and essential goal of the organization. As mentioned before by Dietz (2006) a transaction is the total set of coordination acts and production acts within a generic social pattern. This means concrete for transaction T01: Admit new contract that all data required for execution of the production act to process new contracts including all communicative acts to get the desired result are embedded. The transaction is being executed by actor A01: Contract admitter following the request of actor CA01: Prospect. Grounded in the DEMO theory every transaction can only produce one fact. The services definition of Burbeck (2000) says that every transaction adds specific business value. Therefore a transaction is synonym with a business service.

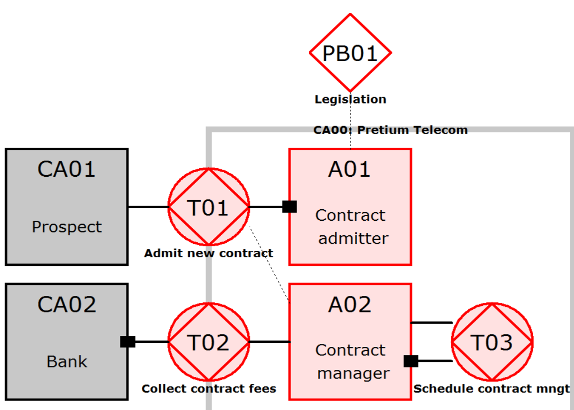


Figure 2: Pretium Telecom

Every transaction in the organizational construction diagram is tested against the determinants of service granularity of Steghuis (2006) as shown in table 1.

For the determinant of functionality Fenton et al. (1997) states that business functionality adds value to an organization. Every transaction in the organization construction diagram is part of the essence of the organization and therefore is every transaction of added value for the business.

The Cambridge University Press dictionary (2009) translates flexibility in “able to change or be changed easily according to the situation” while Steghuis (2006) concluded that the meaning of flexibility in business and IT is the level of changeability of business processes or IT modules. Every transaction in the organizational construction diagram embeds the generic social pattern. This pattern determines the process of communication including the production of the fact. Due to the generic character of the pattern every transaction follows the same process. The inter process dependencies can be connected randomly. This supports the flexibility of the business process orchestration on the level of transactions.

Steghuis (2006) puts one of the four areas of the description of complexity by Fenton et al. (1997) within the domain of business services. This is the complexity of the problem. Steghuis (2006) advises to split the problem in smaller parts to reduce the amount of complexity. This suggestion makes the service granularity lower grained in case the amount of complexity it too big to oversee at once. The DEMO theory does not support this kind of splitting transactions, unless every transaction is producing a unique fact that could be composed into a solution for the total problem. At business level this is a decomposition of production activities for executing a transaction. The decomposition of causally related transaction types is important for the notion of a business process (Dietz, 2006). This is called the composition axiom in DEMO.

Steghuis (2006) refers to the definition of Basset (1997) for the reusability determinant. Basset (1997) defines reusability like the following: “Reuse is the process of adapting a generalized component to various contexts of use”. Steghuis (2006) concluded that while services are defined the granularity of services will be small grained because of the reusability determinant. In this case services contain only a small unit of functionality. Following the theory of DEMO makes reusability of services very clear on all layers of the organization. Every transaction has only one actor who is responsible for producing the fact, but the request for producing the fact can originate from different actors. This shows that reusability is completely embedded into the DEMO theory and is expressed visually by the design.

The compose ability determinant is addressed in the composition axiom of the DEMO theory and already explained in the discussion about complexity.

The sourcing determinant of service granularity is about ordering services to be aligned with sourcing strategy. DEMO is a successful language in expressing implementation decisions within the field of business and IT (Op ‘t Land, 2008). For example, in figure 2 one can discuss about outsourcing transaction T01. In case a decision has been made to outsource T01, the responsible actor A01 becomes redundant within the business domain. Within the DEMO model the impact on the organization becomes clear. The dashed arrow between transaction T01 and actor A02 shows an information link for tacit knowledge. This means that actor A02 needs the knowledge of the production facts, stored within transaction T01, while performing the production acts that are associated with its responsibility. The information links between actors and (transaction) banks restrict the nature of the interaction to the information exchanged (Hoogervorst, 2009). This is called interstriction.

Steghuis (2006) cites Foody (2005) on the aspect of generality. Foody (2005) says that services should be designed from the business point of view. The usability of services by the business is of greater importance compared to simple or generic services. In essence the transactions within an organization construction diagram are designed based on the production facts an organization produces. This makes a direct link to the business value. The aspect of generality is implicitly embedded in every transaction, because DEMO focuses on the ontology of the organization only. For example transaction T01 in figure 2 is capable of processing all kinds of telecom contracts for Pretium Telecom. At the ontological level the processing parameters are of subordinate importance for executing a transaction.

Papazoglou (2003) uses the concept of “loosely coupled” when discussing context independency. This means that a service should be able to execute unaware of the context it runs in. The transactions of the organization construction diagram do not support to be fully context independent. The embedded generic social patterns within a transaction enable to create a chain of transactions fulfilling a process. This chain of transactions causes several waiting moments in the process of communication. In this way a parent transaction can only be finished when the child processes are finished. These dependencies are made visible in a transaction pattern diagram of DEMO, which is not part of this paper. For example the relationship between the transactions T02 and T03 is explained in figure 2. The process starts with transaction T03. At a certain time actor A02 starts transaction T03 to manage contracts. This communication process leads to the request for execution of transaction T02 at the bank. Only when transaction T02 finishes, transaction T03 can finish. This is a periodical cycle. Another context dependent factor is shown in figure 2. Actor A02 can only execute transaction T02 in case actor A02 owns explicit tacit knowledge about transaction T01. The DEMO theory calls this dependency interstriction and is made visible by a dashed line.

Based on the case study DEMO models we can conclude that the transactions defined in the organizational construction diagram addresses the determinants of service granularity at the business layer.

Informational services

The informational layer, as part of the organization theorem, is about collecting information and providing that information to the business layer. The organization theorem defines that the function of the informational layer supports the construction of the business layer (Dietz, 2006). Every actor on the business layer needs information to perform its responsibilities. That information can be versatile, but at least it is associated with a transaction on the ontological level or it is required by interstriction on the ontological level. Thorough information analysis should demonstrate which information exactly is required. De Jong, et al. (2010) uses a methodology that focuses on the situational information and operational information required for executing the responsibility reasoning from the business layer. This method of information analysis results into the exactly necessary information that is required per business transaction on the ontological layer. This paper will not elaborate further on De Jong, et al.'s method of analysis.

The infological model of Pretium Telecom associates the ontological layer by actors CAI01 and CAI02 (see figure 3). These actors are equal to the internal actors on the business layer of Pretium Telecom. Two infological transactions are defined to execute the responsibilities of the actors. These transactions provide access to the informational layer of the organization theory. Because De Jong, et al. (2010) reuses the concepts of the DEMO theory while researching the ontology of the informational layer and the document layer, the characteristics of business transactions also count for transactions on the informational layer. This also means that the determinants of service granularity can be measured with informational transactions.

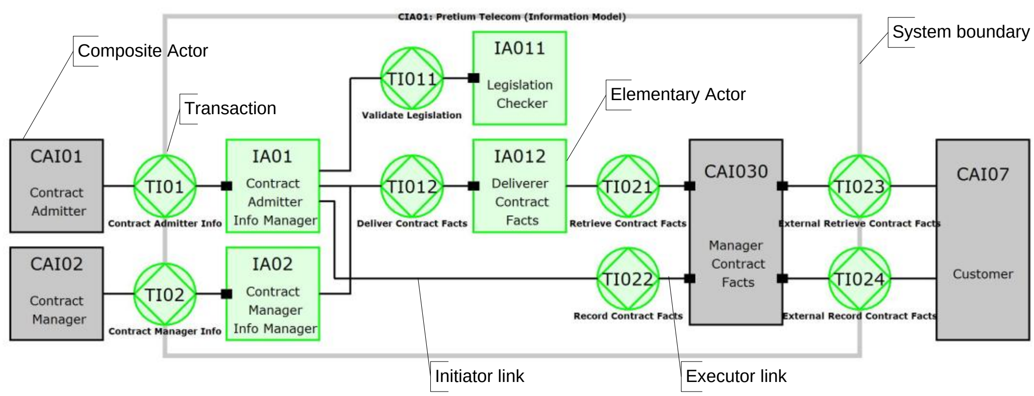


Figure 3: Infological model Pretium Telecom

Despite that DEMO models are abstracted from implementation and order; the case studies showed, based on experience that it is recommended to create the informational models in a left to right order. This way of modelling supports reasoning from a front-end towards a back-end. In this case the front-end functions as a portal towards information, and the back-end as the gateway to the data.

The front-end and back-end approach was applicable for analyzing the informational models in both case studies. The pattern allows to visually splitting the infological model in three areas. Figure 4 shows the front-end and back-end approach in the infological model of Pretium Telecom. The front-end of the informational model is substituted by I.A.P, which stands for Information Access Point. This is the entrance for an actor to get access to the informational layer. The information access path can be implemented in several ways, like a portal or other kinds of user interface. Associated with I.A.P. is INF.S, which stands for Information Services. Per I.A.P. at least one information service is connected. This directly shows which information services are required for an actor. An actor within the domain of Information Services shall access certain banks of facts in processing information. In case the bank of facts is part of the organization it is modelled like actor CAI030. See figure 4. Otherwise, in case the bank of facts is not part of the organization it is modelled like actor IA011. This type of actor points at retrieving information from an external Legislation bank of facts.

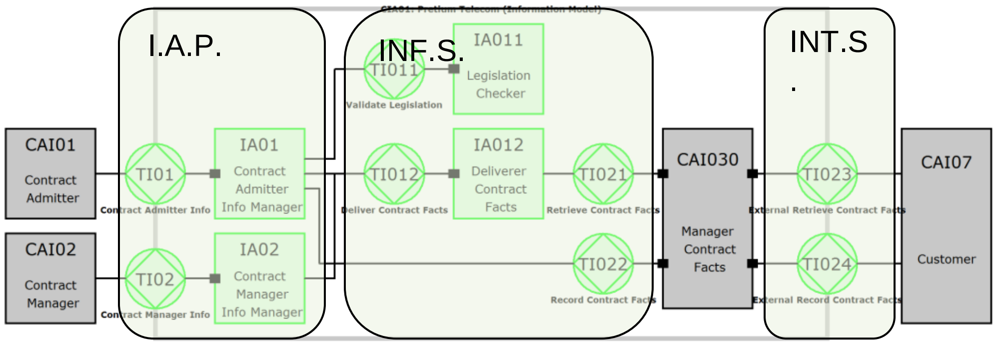


Figure 4: Infological model Pretium Telecom (Front-end and Back-end)

The third area that is recognized in the informational model are the Integration Services, abbreviated as INT.S.. The integration services support the information exchange in the value chain. This has been modelled in figure 3, where a customer can verify and update its own contract information. The integration services are specifically mentioned for external information exchange outside the borders of the organization. Web services can be an applicable implementation pattern for this kind of services.

When the aspects of service granularity for information services from table 1 are being measured on the informational model, then the determinant of functionality is supported by the fact that every information service is primary supporting the construction of the business layer.

The flexibility of the business process is determined at the ontological level. The informational layer exists by supporting the business process and provides information towards that business process. How the information is being interpreted is not relevant within the informational layer.

Cognitive or structural complexity is described by Fenton et al. (1997) as respectively the unit of understanding and interpretation by human beings of complexity in software and the unit of complexity in the structure of software programs. The informational layer itself does not say anything about complexity of the implementation of services. The informational model does provide guidelines for implementation, like transactions TI021 and TI022 in figure 3, where retrieval of information is separated from saving information. An actor could execute several calculations or derivations before the information is provided. The information analysis method by De Jong, et al. (2010) gives a better overview of the required information.

Reusability of information services becomes directly clear in figure 3. Both information managers, actor IA01 and actor IA02, use the informational transaction TI012. The reusability of legacy can be made clear in the same way.

The composability determinant is almost not applicable for the informational layer. The informational model provides a clear overview in the essential information transactions that are required in supporting the business layer. Information analysis will show when an information object must be composed from other information objects.

The infological model can make proposals supporting decision making on sourcing of informational actors within or outside the organizational borders. The infological model expresses this by creating new integration services that are associated with the external information provider in case the information provider is placed outside of the organization.
Sourcing of information services are supported by the principles of the service-centric approach (Arnold et al., 2002). By integrating external information sources into the information process shows the reusability of third

party services. E.g. in the case of Pretium Telecom can a credit check be incorporated on the infological level to check a new customer before the contract is accepted. The credit check service is not part of the responsibilities of Pretium Telecom, but it lies somewhere else.

The transactions in the infological layer do not provide a clear view on the business rules that are applicable for the infological transaction at runtime. A business rule provides a specific actionable guidance to implement business policies, which are non-actionable directives whose purpose is to govern or guide the enterprise (Hoogervorst, 2009). The absence of a clear view on business rules makes informational transaction generic, but when the informational transaction will be implemented those rules become valid. Because the information service does not have any knowledge of the business context, the information service must support to process all kinds of information objects. Due to modelling essential information services only, a generic and concise information service can be designed.

De Jong, et al. (2010) says that informational actors can only possess explicit knowledge. Every information service is responsible for collecting its own data. The information service is not dependent on information from other information services, but maximum limited by the executing order within the business process. The principle of “loosely coupled”, as used by Papazoglou (2003) in explaining context independency is therefore partly applicable. E.g. in figure 3 the transaction TI012 can only be executed when actor IA012 has collected the required data from transaction TI021.

The infological model is abstracted from its implementation. Therefore the infological model cannot provide a guideline for performance. Foody (2005) wrote in his weblog a guideline for information services. An information service or an integration service must be executed between five milliseconds and 5 seconds and the size of the message should not be greater than one megabyte. During implementation of the information services and integration services the performance aspect must play an important role defining the size of the service. The performance aspect itself shall not change the informational model in its essence.

The directive of Information Access Points, Information Services and Integration Services, in relation with a thorough information analysis shows that the infological model meets most of the aspects for service granularity. The essential informational transactions are modelled within the infological model and those transactions are primary supportive for the business transactions on the business layer of the organization.

Data Services

The data logical layer of the organization is about saving, retrieval, removal, and transportation of data. The organization theorem defines that the function of the data layer supports the construction of the informational layer (Dietz, 2006). The functions on the data logical level are of great importance for the actors within the informational layer (De Jong, et al., 2010).

The data logical model of Pretium Telecom associates the infological layer by the actors CAI030 and IA011 (see figure 3). These actors are both associated with the data logical layer. The actor CAI030 is associated because of its responsibility to retrieve and store data. The actor IA011 is associated because of its responsibility to retrieve Legislation data, which is available in an internal or external database.

When the functionality aspect for service granularity from table 1 is being measured with the data logical model, then the aspect of functionality is substantiated by the fact that every data service is primary supportive to the infological layer of the organization.

The data logical model does not provide an answer for the service granularity aspect for cognitive and structural complexity. The data logical model shows that data needs to be stored, retrieved, and transported only, but the model does not show for which data objects these activities count per data logical transaction.

Reusability of data services is expressed in the data logical model by the function of an actor. One actor is responsible for storing data, one actor for transporting data, etc.. Due to the generic responsibility an actor has, the data service can be reused for all data storing request respectively data transporting requests.

The service granularity aspect of composability is not visible within the data logical model. The model shows the essential data logical transactions that are required by the informational layer.

The transactions in the data logical layer do not provide any details about the underlying data structures that are required to process a data service. This makes the data logical layer generic in usage. The business rules that counts for data, when implementing a data structure are not visible from the data logical model. Therefore this kind of business logic must be implemented separately.

Context independency, as aspect of service granularity, is supported by the data logical layer by associating explicit responsibilities per actor for data collecting, data storing, and data transporting. Every actor does not have to know anything about other actors when executing its responsibility. This supports the statement of “loosely coupled” services, which makes the data services in the data logical model completely context independent.

The performance aspects of service granularity on the data logical layer are equal with the informational layer. The data logical layer is implementation independent and therefore cannot provide a guideline on the topic of performance.

Regarding the data logical model associated with the aspects for service granularity concludes that the data logical model’s abstraction level is too high. It does not contribute much in obtaining optimum service granularity.

Findings from the case studies

Designing an information architecture grounded in the requirements of the business and/or the market, and the opportunities of IT are one of the driving forces for the IT function (Poels, 2007). The information architecture is translated into construction principles. These principles are directive in selecting standard components or developing custom made software (Poels, 2007).

Both case studies started from the construction principles. The ontological model of business services shows the essence of the organization without forcing any form of implementation. Measuring the aspects of service granularity showed the connection between business transactions on the business layer and business services. Every business service can be associated with the business value it obtains and how it supports the goals of the organization. The business service is not only about executing its responsibility, but also about the coordination activities within the social domain.

Every business service represents a part of the ontology of the organization and therefore every business service provides a unique unit of functionality. This precise amount of functionality within a business services results into an optimum in service granularity. The aspect of context independency is limited by the interstriction possibilities between business services. With help of a transaction pattern diagram the relative context independence of business services is shown.

To switch between the ontological business level and the ontological information level a business actor can shape itself into an information actor (De Jong, et al., 2010). A thorough information analysis must be executed that results in information services. The case studies showed that the list with information objects is directive in defining the service messages that actors interchange. Case studies showed that the information analysis must not be limited to the requirements of the business level only, but also must take into account the already existing physical data structures, like existing database models. The information analysis associates the business requirements with the data.

The information services that are designed grounded in thorough information analysis can be split up in three areas. These are: Information Access Points, Information Services, and Integration Services. The Information

Access Points provides a clear organization of the information model. They are services that provide access to the information. The Information Services are supporting the business requests for information. These information services are supporting the business process and represent a large part of the essential information supply. The integration services are specifically mentioned to support the information integration with external parties. These information services are not primary supportive to business actors, but are supportive for the business goals in the area of the extended enterprise. A practical form of implementation is web services.

Every information service is a piece of functionality that must be supported by the system. This provides structure to the functionality that has to be designed. The information service is an expression of the complete social pattern of coordination and production. This means that the information service itself must support the social pattern as well. Based on the case studies results this is hard to combine together in one physical service. By creating a composite service at the implementation level, the DEMO informational model provides a functional directive in obtaining optimum service granularity. When the functional information service model of DEMO is being implemented, the service granularity determinants complexity and performance require extra attention, because these aspects are not directly supported by the methodology.

Both case studies showed that the data logical layer of DEMO has a too high level of abstraction. This makes it impossible to conclude that the data logical layer contributes to obtain optimum service granularity. The physical implementation of the data structures are a prescribing factor, which is not taken into account in the data logical model. The infological model, a thorough information analysis, and the existing data structures are together qualifying optimum service granularity for data services.

Conclusion

The hypothesis of this paper says that DEMO in combination with current scientific research of De Jong, et al. is directive for an optimum service granularity. Based on the results of the case studies the hypothesis is considered to be valid. The business model and the informational model of DEMO are both offering enough leads to make it plausible that optimum service granularity can be reached on a functional level.

At ontological layer the DEMO transactions are synonym with business services. The business services of both case studies are measured against the aspects of service granularity of Steghuis (2006). For almost all aspects the business services own the characteristics to achieve optimum service granularity. The context independency aspect of service granularity is not measured in the organization construction model, but with the transaction pattern diagram.

One of the case studies conclusions is the categorization of the information services on the infological layer. These categories are: Information Access Points, Information Services, and Integration Services. The information access points associate the business actor with the informational layer and offer the business actor the information supply that is required for executing the business service. The information services are basically the essence of the information supply and supportive for the business layer. The integration services are mentioned for information integration with external parties, like in an extended enterprise.

Every type of information service is measured with the aspects of service granularity. For almost all aspects the information services supports an optimum service granularity. Because that the information services could be prescriptive for physical implementation, the remark must be made that the informational model gives a functional directive in obtaining optimum service granularity. During implementation extra attention must be given on the aspects of complexity and performance.

The final conclusion based on the results of the case studies is the high abstraction level of the data logical model. Therefore this model hardly contributes in obtaining optimum service granularity.

The two case studies in this paper showed that the applied methodology is directive in obtaining a functional optimum in service granularity for business services and information services. Further research should demonstrate if the methodology used in this paper could be raised as best practice for obtaining optimum service granularity.

References

Arnold, B., and Op 't Land, M. (2002). An Architectural Approach to the Implementation of Shared Service Centers. In: Dutch National Architecture Congress 2002 (LAC2002). Zeist, The Netherlands

Basset, P., (1997). Framing Software Reuse – Lessons from the Real World. (1st ed.). New York: Prentice Hall

Burbeck, S. (2000). The TAO of E-Business Services – The Evolution of Web Applications into Service-Oriented Components with Web Services. Tech. rep., IBM Software Group

Bunge, M.A., (1979). Treatise on Basic Philosophy. A World of Systems, Vol. 4. Dordrecht: D. Reidel Publishing Company

Bryman, A., and Bell, E. (2007). Business Research Methods – second edition. Oxford University Press

Cambridge University Press. Cambridge Advanced Learner's Dictionary. Retrieved April 2009 from http://dictionary.cambridge.org/

Cox, D. and Kreger, H., 2005. Management of the service oriented-architecture life cycle, IBM systems Journal, Vol 44, No 4.

Dietz, J.L.G., (2006). Enterprise Ontology – Theory and Methodology. Springer Verlag

De Jong, et al., J., and Dietz, J.L.G., (2010). Understanding Organization Realization. 6th International Workshop on Cooperation & Interoperability – Architecture & Ontology. DESRIST 2010

Erl, T., (2005). Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall

Fenton, N., and Pfleeger, S., (1997). Software metrics – a Rigorous & Practical Approach. (Second Edition ed.). Boston: PWS Publishing Company

Feuerlicht, G., Wijayaweera, A., (2007). Determinants of Service Reusability, New Trends in Software Methodologies, Tools and Techniques, H. Fujita and D. Pisanelli (Eds.) vol. 161, pp 467-474, IOS Press

Foody, D. (2005). Getting web service granularity right. Retrieved April 2009 from http://www.soa-zone.com/index.php?/archives/11-Getting-web-service-granularity-right.html#extended

Hevner, A.R., and March, S.T., and Park, J. and Ram, S. (2004). "Design Science in Information Systems Research", *MIS Quarterly*, 28

Hoogervorst, J.A.P., (2009). "Enterprise Governance and Enterprise Engineering." Diemen, the Netherlands. Springer

March, S. T. and G. F. Smith (1995). "Design and natural science research on information technology." Decision Support Systems 15(4): 251-266.

Op 't Land, M., (2008). Applying Architecture and Ontology to the Splitting and Allying of Enterprises. Ph.D Thesis, Delft University of Technology

Op 't Land, M., and De Jong, et al., J., and Goedvolk, H., (2008). Enterprise Engineering Framework. Not yet published

Papazoglou, M., (2003). Service-Oriented Computing: Concepts, Characteristics and Directions. In: Fourth International Conference on Web Information Systems Engineering. IEEE. Roma, Italy

Papazoglou, M., and Van den Heuvel, W. (2006). Service-Oriented Design and Development Methodology. International Journal of Web Engineering and Technology

Poels, R., (2007). Haal meer uit uw ICT: Interventies die ertoe doen. Scriptum

Rosen, M., (2007). A Methodology for Designing Service-Oriented Applications. Cutter Information

Schekkerman, J., (2004). How to survive in the jungle of Enterprise Architecture Frameworks: Creating or choosing an Enterprise Architecture Framework. Trafford

Sims, O. (2005). Developing the Architectural Framework for SOA – Part 2 – Service Granularity and Dependence Management. CBDi Journal, June 2005

Steghuis, C. (2006). Service Granularity in SOA Projects: A Trade-off Analysis. MSc Business Information Technology, University of Twente

Weske, M. (2007). Business Process Management: Concepts, Languages, Architectures. Springer-Verlag Berlin Heidelberg