A scientific evaluation of telecom industry standards

Using enterprise ontology and the identification of business components

MSc thesis by Jan Drenthen

TU Delft
Delft University of Technology

A scientific evaluation of telecom industry standards

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Jan Drenthen
born in Delft, the Netherlands

Information Systems Design Group
Department of Software Technology
Faculty EEMCS, TU Delft
Delft, The Netherlands
`www.ewi.tudelft.nl`

Alcatel-Lucent Switzerland
Friesenbergstrasse 75
Zurich, Switzerland
`www.alcatel-lucent.ch`

Cover picture: *Alcatel-Lucent Switzerland mission statement - Daniel Reichenbach.*

A scientific evaluation of telecom industry standards

Author: Jan Drenthen
Student id: 1149776
E-mail: `jan@drenthen.ch`

Abstract

Alcatel-Lucent Switzerland (ALU) manages the networks of major telecommunication service providers. The extent of these insourced telecom services is described in a Service Level Agreement. It it essential for ALU to comply with the agreed service level, since violation will result in severe penalties. As of this moment, ALU's service desk monitors the networks for Sunrise, the second largest telecom service provider in Switzerland. Shortly, the network management for Orange, the third largest, will be insourced as well. Industry standards are currently used to identify the relevant processes and underlying ICT functionality that are present in the service desk. However, these industry standards lack a scientific foundation. This thesis evaluates the used industry standards by comparing them with two promising scientific concepts: enterprise ontology and the identification of business components. Enterprise ontology is believed to provide an overview of the organizational activities in such a way that it can be used as a point of departure for devising supporting ICT functionality. The latter is done by using the business component identification method, which uses an optimization algorithm in order to identify an optimal distribution of ICT functionality.

Thesis Committee:

Chair:	Prof. dr. ir. Jan L.G. Dietz, Faculty EEMCS, TU Delft
University supervisor:	Dr. Antonia Albani, Faculty EEMCS, TU Delft
Committee member:	Dr. Joseph Barjis, Faculty TPM, TU Delft

Preface

This document describes the results of my master's thesis project, which I did at Alcatel-Lucent Switzerland in Zurich. I had the pleasure of spending 8 months in this inspiring environment, in which I first did my research assignment, diving in the depth of telecommunication industry standards. My thesis project, immediately following my research, evolved around the scientific evaluation of these standards. Before diving into depth, I would like to thank some people that made this project happen.

The first step towards the birth of this project was taken when I walked in Antonia Albani's office. Antonia was an assistant professor in the Software Engineering group of my faculty. She gave lectures in two courses which I participated in the Master of Science curriculum. I asked her for the possibility of doing my master's thesis project in Switzerland, her home country. Before I noticed, she had contacted her brother, Jon Erni, who is heading the Marketing & Sales organization of Alcatel-Lucent Switzerland. A few months later, I was having a meeting in Zurich with Jon Erni and Jürg Wenger, discussing my assignment.

First of all, I would like to thank Antonia for her guidance and assistance during and before the project. I am very grateful for her proactive approach, both in initializing this project as well as during my time in Zurich.

Furthermore, I would like to thank some people at Alcatel-Lucent Switzerland. First, I thank Jon Erni and Jürg Wenger, for welcoming me in this great company. I would also like to thank Mike Jäger for his support and introducing me in his team. Last but not least, I thank Thomas Buff for spending much time with me in giving valuable feedback and sharing his extensive expertise.

Jan Drenthen
Delft, the Netherlands
August 24, 2010

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Traditionally, companies that provide telecommunication services to their customers, so-called *telecom service providers*[1] (TSPs), were primarily concerned with keeping their networks up and running. There usually were a few big players in each country—who knew their competitors very well—struggling to be the lowest cost provider [26]. In order to assist TSPs in defining their approach for effectively managing their networks, standardization organizations for telecommunications have largely contributed by defining solution frameworks and guidelines in this field. In the late 1980s, the International Telecommunication Union (ITU) developed the so-called FACPS model, in which FACPS represented the five main functions of telecommunication network management: Fault Management, Accounting Management, Configuration Management, Performance Management and Security Management [19]. Altogether, during the first era of telecommunications, the TSPs focused their activities mainly at the operational level rather than the business level.

Starting from the 1980s, the integration of computations and telecommunication technologies created new business opportunities and the TSPs faced a much more demanding environment. Mobile phones and the Internet gained global acceptance. Being able to cope with the changing demand for communication services is key for TSPs in order to survive. Successful TSPs typically are able to profoundly leverage groundbreaking technologies, by introducing services that fulfill the consumer requirements such as personalization and interactivity for end-customers and cost-effectiveness and efficiency for enterprises. Although the FCAPS model has been widely adopted by TSPs as their guideline to develop their network management capabilities, it lacks the focus on delivering high quality services to customers. Therefore, the model is no longer sufficient for managing today's critical focus on service management. Shifting from network management to service management is a major challenge for TSPs. A telecom industry organization that largely contributed in this field by developing four frameworks, is the TeleManagement Forum[2] (TM Forum). The TM Forum solution frameworks, as they are called, replaced the

---

[1] e.g. Swisscom, Vodafone, T-Mobile, KPN

[2] `http://www.tmforum.org/`

deprecated FCAPS model and are nowadays widely adopted by TSPs. They include frameworks that give an overview of the business processes that TSPs should implement, along with an extensive list of applications that support these processes.

1.1 Problem statement

Alcatel-Lucent (ALU[3]) supports service providers, companies and governments in delivering voice, data and video communication services to end-users. In Switzerland, ALU insources a significant part of the operations of Sunrise and Orange, the second and third largest of Swiss TSPs, respectively. Every part of the entire operational activities of these companies is insourced, with the exception of brand management, end-customer relations and billing. ALU's vision is to be the sole supplier of multi-vendor managed services for and with carriers in 2012 [5].

In insourcing contracts with service providers, each detail of the delivered services is described in qualitative and quantitative ways, in so-called *Service Level Agreements* (SLAs) and *Key Performance Indicators* (KPIs), respectively [15]. Whenever ALU does not comply with the agreements, penalties have to be paid to the customer. In order to serve their customers in a high-quality and cost-effective way, ALU's goal is to design a unified environment in which common service enablers can constitute value-adding telecom services, a so-called *service delivery environment*. In order to successfully insource their customers' operations, ALU uses the aforementioned TM Forum solution frameworks in order to identify business processes and supporting information and communication technology (ICT) functionality.

Due to the tough competition in the market and the harsh economical situation, it is important that ALU operates the networks of their customers at the lowest possible costs. This implies that processes have to run as efficiently as possible and that a significant reduction in business process complexity has to be reached, as well as the cost-effective utilization of ICT resources. Although the TM Forum solution frameworks are implemented within ALU, there is no scientific foundation stating that ALU is performing in the most efficient way. Although the frameworks are believed to be sufficiently extensive when it comes to identifying processes and supporting ICT functionality, they cannot demonstrate that processes are run as efficiently as possible and that ICT functionality is distributed as it is supposed to be in an optimal Service Oriented Architecture (SOA) environment.

1.2 Research question and steps

This thesis evolves around this question whether ALU's business processes and ICT systems are structured in the right way. In order to scientifically approach this question, two promising scientific concepts are applied: *enterprise ontology* and the identification of *business components*.

---

[3] from now on, the abbreviation 'ALU' refers to Alcatel-Lucent *Switzerland*

Enterprise ontology builds on a profound theory about the elements that constitute the complexity of an organization. It makes use of a modeling methodology, in which organizational activities are modeled in a *consistent*, *comprehensive*, *concise* and *essential* way [12]. Models of organizational activities that adopt the notion of enterprise ontology are completely independent of their implementation, as only the essential elements are included within the models.

For reasons mentioned in the previous paragraph, ontological models are suitable as a starting point for designing information systems. The first step in the development of a modern day information system, is the identification of business components. The business component identification (BCI) method, described in [1], uses ontological models of the organizational activities within an enterprise, and applies a number of steps on these models to capture business components. The BCI method's underlying algorithm finds the optimal distribution of functionality. Key to note is that the derived business components are not ICT systems themselves, but rather clusters of coinciding functionality.

1.2.1 Research question

Having introduced the two scientific methodologies to approach the inquiry posed in the beginning of this section, the research question of this graduation project has been defined as follows:

What are the advantages and disadvantages of combining enterprise ontology and the business component identification method in identifying business components and ICT services for supporting ALU's operate-processes, compared to the current approach taken by Alcatel-Lucent, which is based on the TM Forum solution frameworks?

1.2.2 Reseach steps

In order to answer the research question above, the following research steps have to be performed:

1. Obtain a clear understanding of the TM Forum solution frameworks;
2. Obtain a clear understanding of enterprise ontology and the identification of business components;
3. Describe the business processes that are present in operating telecommunication networks;
4. Describe the ICT services that are used in operating telecommunication networks;
5. Build ontological, infological and datalogical models of these business processes;

6. Apply the BCI method, using these models as an input, in order to obtain business components and ICT services;

7. Identify the differences of the results using **a)** enterprise ontology and the BCI method, and **b)** ALU's approach based on the TM Forum solution frameworks.

1.3 Research scope

This section describes the scope of the thesis project. Firstly, it describes which part of the network operations are taken as input for the scientific evaluation. Next, the extent to which enterprise ontology is being applied will be described.

1.3.1 Scope of network operations

As mentioned before, ALU has network operations insourcing contracts with two major Swiss TSPs: Sunrise and Orange. Within the telecom industry, there is a shared agreement on the four different functional groups that comprise telecom operations. The first, *fulfillment*, provides customers with their requested products in a timely and correct manner. Once these products are provided, operations in the *assurance* group concern the execution of preventive and corrective maintenance activities to ensure that services provided to customers are continuously available and performing to SLA performance levels. Next, *billing* includes determining and charging billing information, producing bills and processing payments. Lastly, *operations support and readiness* is about providing management, logistics and administrative support to the fulfillment-assurance-billing process groupings.

For this thesis project, detailed information on ALU's business processes is required. However, it is not necessary to use business processes from the full width of the organization. As mentioned before, the scope of this project is to use the operate-processes of ALU as an input, specifically processes in the assurance group (see above). The operate-processes include spare part management, preventive maintenance and on-site support.

Currently, described and implemented operate-processes are in place at ALU's network operations insourcing activities for Sunrise in Switzerland. Assurance activities for Sunrise are performed by the *service desk*. The information obtained and used throughout this document will be based for a large part on input from service desk representatives. Althought the final outcome (i.e. business components based on ontological models) is solely based on Sunrise processes performed by ALU, they are representative for network operations processes for both Sunrise and Orange.

Although this thesis evaluates whether processes and ICT services are run efficiently, there will be no emphasis on the actual implementation of the ICT services that fulfill the clustered functionality of business components. First of all, this is not a focus topic of this research. Secondly, the implementation of ICT services would most likely be very time consuming. Instead, the focus will be on optimizing the distribution of ICT services.

1.3.2 Scope of enterprise ontology

This subsection describes the scope of the ontological models that will be produced for this research. Some terms are not yet introduced. Additional background information of enterprise ontology is given in section 2.2.

As mentioned before, enterprise ontology abstracts the essence of organizational activities and is expressed in a model. Such a model is an ideal point of departure for devising high level business components, i.e. clusters of coherent functionality, by using the BCI method. The business components that are derived can then be directly supported by ICT services. Until now, the BCI method has mainly been applied to the B-organization of an enterprise.

Hypothetically, the BCI method could also produce a result using models of the I-organization as an input. These are transactions that do not involve performa activities, but rather involve actions like reproduction, deducing, reasoning, computing etc. On an even lower level, the D-organization, actions involve storing, transmitting, copying and so on.

Logically, the result would be ‘business’ components that are actually clustered functionality supporting the I- and D-organization, in addition to the B-organization. Resulting from that, it would be interesting to know whether B- business components would *contain* I- business components, or whether I- business components would support B- business components. Investigating these relationships constitute the scientific contributions of this thesis project.

1.4 Motivation

This section describes the justification of this project, both for ALU as well as for science.

1.4.1 Motivation for ALU

For ALU, network operations are a large part of the company’s total revenue. Although annual results are still written in red ink, there are strong beliefs that this will change within a few years, due to synergies, such as having one network operations center for multiple customers. ALU would benefit from a insightful representation of their operate-processes, such that the implementation of operate-solutions at new customers will be much faster. Furthermore, having business components that are reusable, is a huge cost saving opportunity. This thesis project gives insight in whether ALU is on the right track with using industry standards as the TM Forum solution frameworks.

Using the operate-processes as input, is representative for two reasons. First of all, one process may run over multiple actors, such as customers, ALU's service desk and external partners. Process steps include the movement of people and resources, the execution of tangible actions (e.g. repairing a base station) and the use of ICT resources.

Second, ALU's operate-processes are analogous to the TM Forum's process framework—especially the 'assurance' vertical processes. Using the university's methodologies next to this framework, makes this research applicable for other companies that, like ALU, perform network operations, as well as for telecom service providers.

1.4.2 Scientific motivation

The thesis project is supported by the faculty of Electrical Engineering, Computer Science and Applied Mathematics at Delft University of Technology. In particular, this research is in line with the ideas of the *Cooperation & Interoperability—Architecture & Ontology* (CIAO!) group, whose mission is to stimulate the development of the emerging discipline of *enterprise engineering*, as well as its practical application in improving the societal performance of enterprises [9]. Methodologies that are developed by this research group, include those that focus on the field of enterprise ontology and the identification of business components.

First of all, this project contributes to existing research topics of the CIAO! group, as it goes beyond the traditional identification of ontological business components, because it will also use an infological and datalogical model as input. Investigating the relationships between the resulting identified business components would add valuable knowledge to this research topic.

Secondly, many practical cases are know where these methodologies are (successfully) applied, but the field of telecommunications has remained unexplored. This project gives new insights that can be applicable within other businesses and industries. Furthermore, making the comparison with industry standards that have a similar focus, may compliment these scientific methodologies with some best practices.

1.5 Report structure

To answer the research question defined in section 1.2.1, the set of research steps in section 1.2.2 were executed. The structure of this document largely corresponds with the research steps. Figure 1.1 provides an overview of the order of research steps taken and their expected results. Simultaneously, figure 1.1 illustrates the structure of this report and shows which research steps are (partly) answered in each chapter.

The current chapter obviously provides the research question of this thesis project. Its definition leads to research steps that need to be performed and consequently results in a complete graduation assignment formulation.

In chapter 2, background information is gathered to obtain a thorough understanding of the context of this project. The first section discusses the TM Forum solution frameworks and its applications at ALU. The second and third section explore the theory and methodology of enterprise ontology, as well as the BCI method. The combination of industry standards and scientific methodologies form a necessary foundation for the succeeding chapters. This chapter performs research steps (1) and (2).

Chapter 3 gives an overview of the processes that are performed at the service desk. The processes are obtained through interviews with employees and documents provided by employees. The processes are described by means of flowcharts and textual descriptions. This chapter also provides an overview of the ICT systems which are used to support the processes. This chapter describes the ‘as-is’ situation and performs research steps (3) and (4).

In chapter 4, the processes of the previous chapter are used as a point of departure for devising models using enterprise ontology. The methodology which is used to design the models, will be explained in section 2.2. This chapter performs research step (5).

Chapter 5 uses the ontological models as input for the BCI method. The method itself will be explained in section 2.3, but the way in which the method will be applied is explained in this chapter, which performs research step (6).

Chapter 6 evaluates the standards from industry by comparing the outcome of chapter 3 (network operations ‘as-is’ situation) with the results of chapter 5 (business components). This chapter is performs the ultimate research step (7), which results in an overview of the differences between the approach using industry standards and the scientific approach using enterprise ontology and the BCI method.

In chapter 7 the conclusions based on summaries of the previous chapters are provided. These conclusions build up to a final conclusion: the answer to the research question. This chapter also describes the recommendations for potential future work.

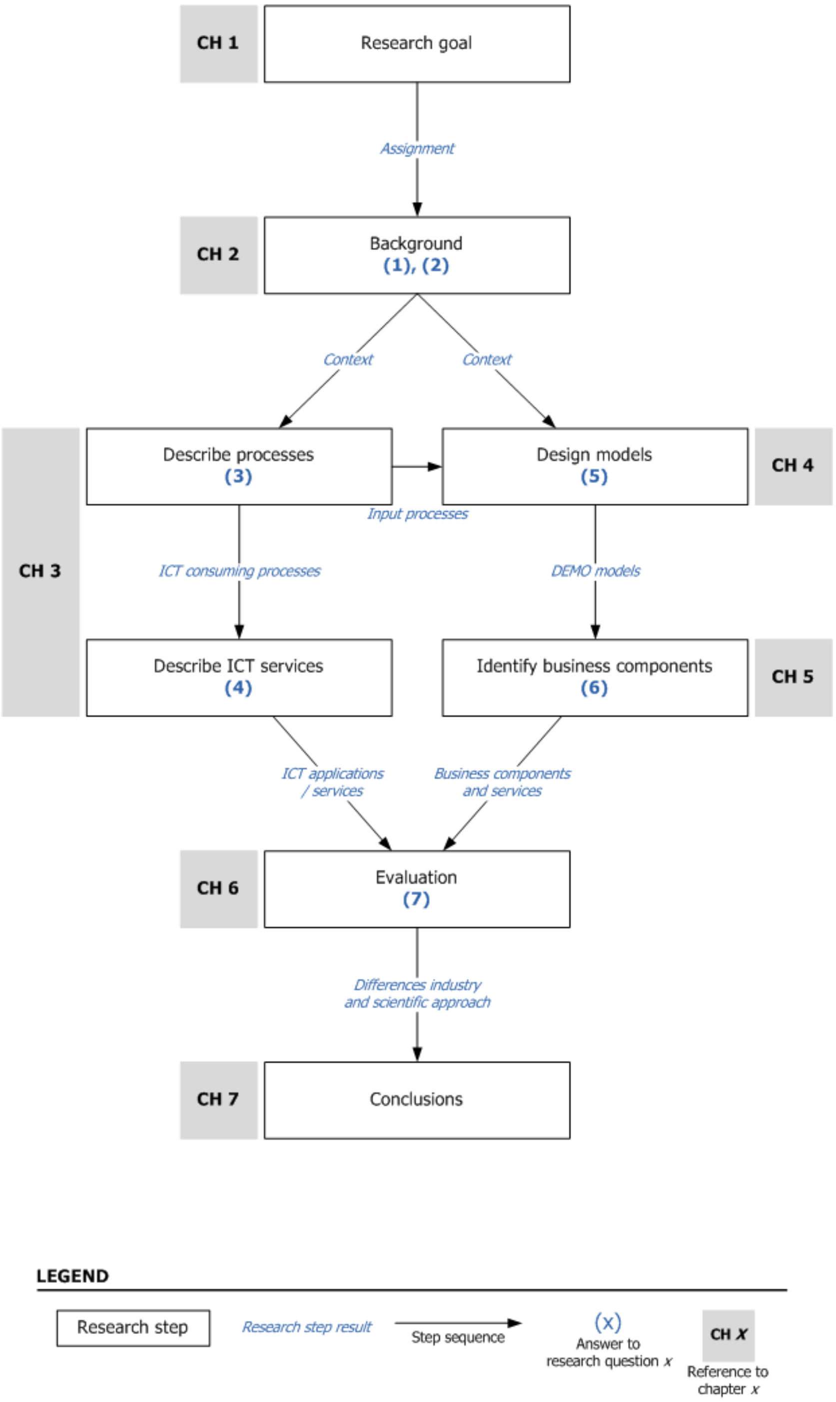


Figure 1.1: Report structure with research steps

Chapter 2

Background

This chapter describes the necessary background information required to obtain a thorough understanding of the context of this project. Section 2.1 discusses the two most relevant TM Forum solution frameworks and its applications at ALU. Sections 2.2 and 2.3 explore the theory and methodology of enterprise ontology, as well as the BCI method. The combination of industry standards and scientific methodologies form a necessary foundation for the succeeding chapters. Some concluding remarks are given in section 2.4. This chapter performs research steps (1) and (2).

2.1 TM Forum solution frameworks

The TeleManagement Forum (TM Forum) is a non-profit industry organization that helps telecom service providers performing all aspects of their business. The TM Forum is continuously working on their solution frameworks, which are also referred to as the *New Generation Operations Systems and Software* (NGOSS). This is not a single deliverable. Rather it is a collection of documents and addenda, each describing a certain part of the solution frameworks. These documents are continuously improved with help of its members: in total around 700 ICT and telecommunication companies [22], including Alcatel-Lucent. The solution frameworks give a complete view of a service provider's organization, by addressing all dimensions of the organization: business processes, management data/information, applications and an integration methodology [23]. These four different dimensions are covered in frameworks, two of which (processes and applications) are discussed throughout this chapter.

2.1.1 Process framework

The TM Forum process framework[1] is a reference framework in which all business activities that a service provider will use are categorized in a structured manner, allowing these to be addressed at various levels of detail [24]. The process framework

---

[1] previously know as the Enhanced Telecom Operations Map (eTOM)

is an abstract 'one size fits all' framework. It does not address the strategic issues or questions of who a TSP's target customer should be, what market segments the service provider should serve and what a TSP's vision and mission statement should be. It will typically be customized and extended for TSP's own specific business needs, but in general provides a profound common reference that is recognized by the telecommunications industry and has become an official standard on business process definition [19].

The process framework categorizes process elements from business activities in such a way that these can be combined in different ways, to implement end-to-end business processes. The structure of the process framework can be seen from a few levels of detail [20]. In this section, the process framework is explained in a top-down way, i.e. first the 'big picture' is given. This highest conceptual view is given in figure 2.1. This view provides a clear separation between strategy and operational processes, seen as the two adjacent largest boxes in the picture. It also distinguishes four so-called 'functional' areas as horizontal layers that cross both process areas. The third process area, which is 'enterprise management', concerns the management of the enterprise itself. These are processes that are not specific for TSPs, as they are also present in any other enterprise. Additionally, the ovals in figure 2.1 represent the internal and external entities that interact, in some way or another, with the enterprise. The three process areas shown in this picture are called 'level 0' process groupings.

In figure 2.2, the level 0 processes in the figure 2.1, are decomposed into their constituent level 1 process groupings. This view is much more detailed and gives a complete overview of the process framework. Level 1 processes show a decomposition of the level 0 processes, in both vertical as well as horizontal process groupings.

The vertical process groupings in figure 2.2 are end-to-end processes that are required to support customers and manage the business. Among these seven process groupings, the main focus lies on the three core process groupings: fulfillment, assurance and billing (FAB). FAB are three vertical process groups in the operations process area, the fourth being operations support and readiness (OSR). OSR is differentiated from FAB, because OSR is an enabler of support and automation within FAB. In the other large process area—strategy, infrastructure and product—the strategy & commit vertical process group and the two life cycle management process groups are differentiated from the operations process area. This is because—unlike the processes in the operations process group—the processes in the strategy process group do not directly support the customer.

Whereas the vertical process groups divide the level 0 process areas into process groups for parts of the business, the horizontal function process groupings in figure 2.2 distinguish functional process. Having distinguished these horizontal process areas, the entire supply chain (from customers to suppliers/partners) is taken into account.

As mentioned in the introduction of this document, ALU is insourcing telecom service provider's operations. It is also stated that end-customer relations and brand

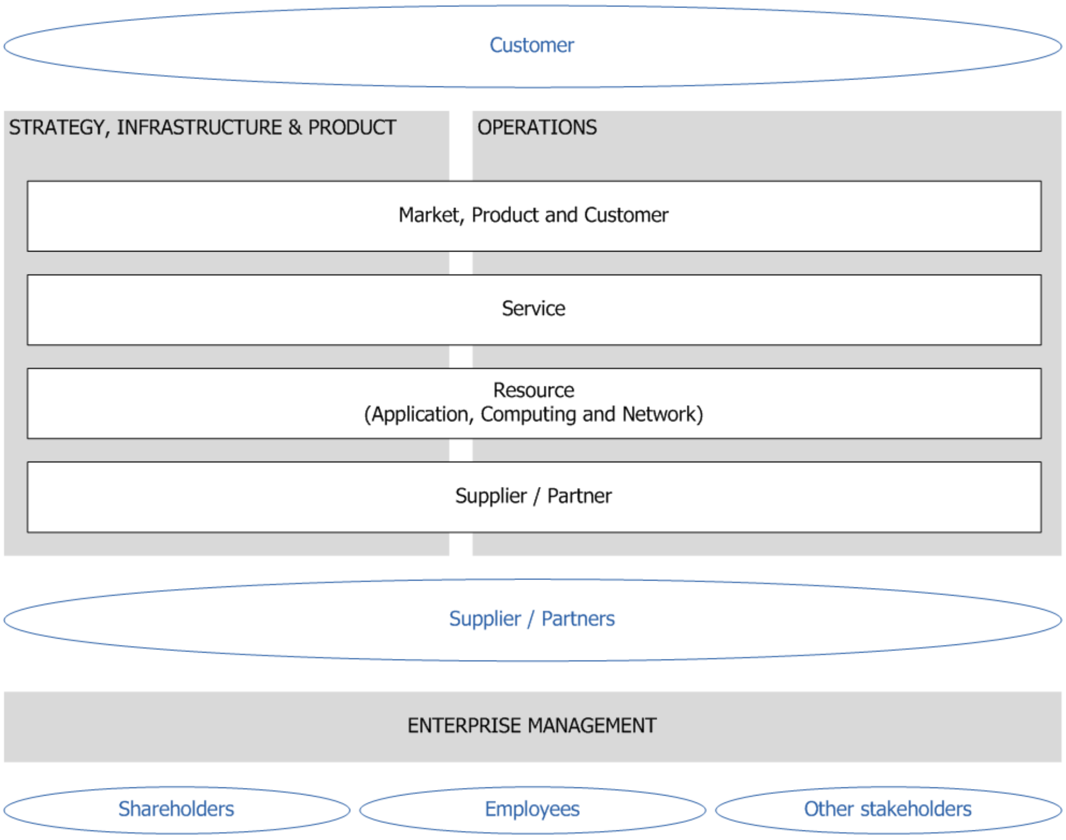


Figure 2.1: Process framework - level 0 processes

management are still performed by these companies. Looking at business processes that are insourced, the process framework provides an ideal overview in which insourced business processes can be depicted, as seen in figure 2.3 [7]. Both the green and blue marked process groups in this picture are entirely insourced by ALU. This thesis project concentrates on the blue marked process groups. These processes are described in the next chapter.

2.1.2 Applications framework

The applications framework, also known as the Telecom Applications Map (TAM), is used by telecom service providers and their suppliers, in order to share a common language to manage their complex network of interdependent systems. Where the process framework provides a reference framework for business processes and the information framework provides a reference framework for information entities, the applications framework provides a frame of reference for telecom applications [25]. The applications framework is expressed in a rather bulky document, listing the requirements for the various applications in the different domains [25].

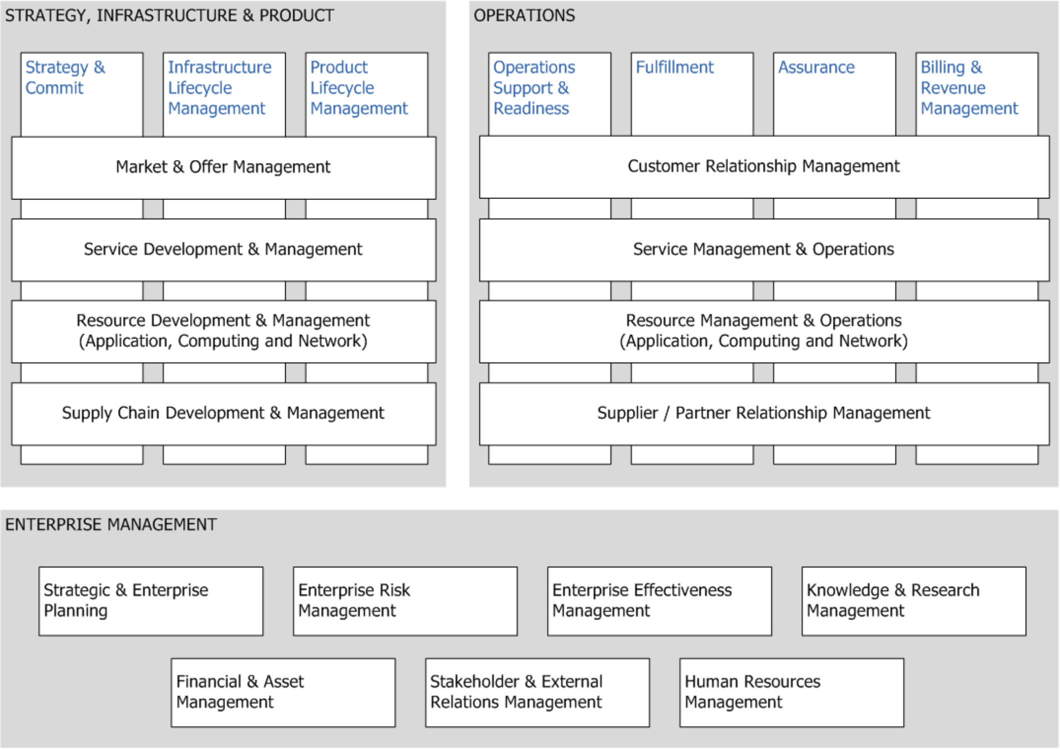


Figure 2.2: Process framework - level 1 processes

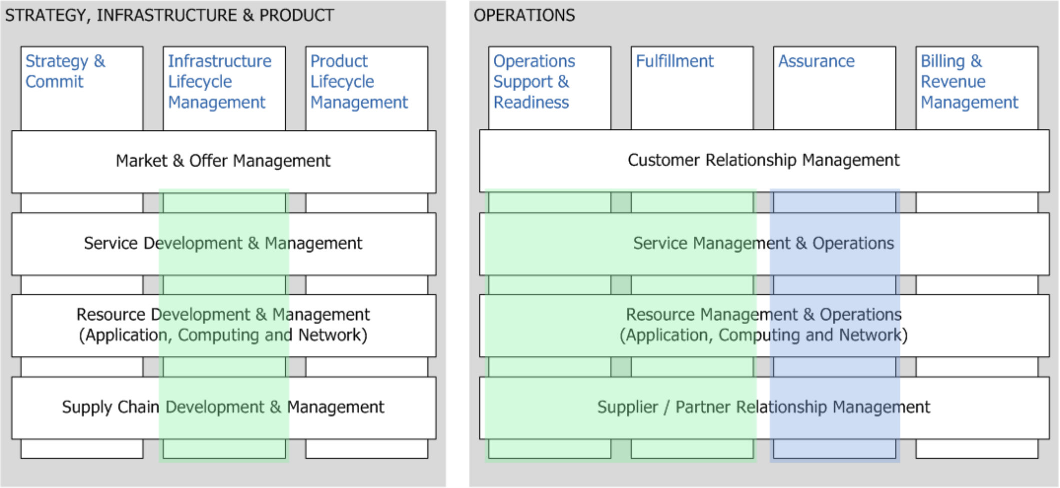


Figure 2.3: Network operations insourced by ALU (marked blue)

The applications framework provides a bridge between business processes, business entities and applications by giving descriptions of self contained applications or services, that fulfill grouped process functions and consume business entities. For each of the functional domains, which to a large extent align with two previously mentioned frameworks, a number of ‘applications’ are identified by the framework, that telecom service providers should be using to successfully run their business. These applications are not software systems that one can buy on the market. Rather these are clustered functionality that should be fulfilled by a single application.

Two types of applications are distinguished within this context: business support systems (BSSs) and operations support systems (OSSs). This distinction is commonly used in literature and throughout industry, some of which is far related to telecommunications. Therefore a definition will be given [21]:

Business support systems: Computer systems that enable telecom service providers to run their business operations towards customers.

Operations support systems: Computer systems that support the daily operation of a carrier’s infrastructure.

The applications framework has taken best practices on systems design and engineering into account and the framework will evolve as those principles evolve, triggered by input sessions with the TM Forum’s members. An example of how the applications framework can be used, is a telecom operator modeling its current (as-is) OSSs in a structured way and using the framework for developing the desired (to-be) applications landscape, obtaining gaps by a comparison of both. Using a common terminology enables involved actors, for instance consultants, suppliers or system integrators, to understand both the current situation as well as the required situation. Another example: a supplier uses the applications framework to highlight the systems that it supplies on its own, and those that are delivered together with a partner. It can be used to show a change in portfolio, or to analyze the OSS market in terms of growth. The applications framework can thus be used across the entire telecom value chain [14]:

1. Middleware suppliers (e.g. Cisco Systems);
2. Applications / Systems suppliers (e.g. IBM);
3. Solution suppliers (e.g. ALU);
4. Service and network operators (e.g. ALU).

As seen with the frameworks which were described in previous sections, the applications framework uses the same definitions. It has the same layered structure as the process framework and information framework. For each layer (from market/sales to supplier/partner and enterprise management) the applications framework describes the principal functions.

For ALU, the applications framework is most relevant on the operations side. All the knowledge on this subject within ALU is bundled in a competence center for OSS/BSS.

2.2 Enterprise ontology

Managing an enterprise nowadays is very complicated. The problems are well known and documented. Enterprise ontology provides a profound theory about the elements that constitute this complexity. It gives a conceptual model of the organizational activities in a *consistent*, *comprehensive*, *concise* and *essential* way [12]. Consistency means that the conceptual model does not contain any irregularities and is free from contradiction. Comprehensive means that the conceptual model is complete, i.e. that it covers all relevant aspects. However, a model that is too extensive and which contains superfluous matters is not desirable; therefore enterprise ontology focuses on the elements that are significant, by delivering a concise model. The most important property, however, is that the conceptual model of the enterprise only abstracts the essence of an enterprise's organizational activities. This means that it is free from implementation details.

Section 2.2.1 explains the most relevant theory that underlies enterprise ontology: the Ψ-theory. The Ψ-theory consists of four axioms and one theorem [12]. Section 2.2.2 briefly describes two additional theories, which complete the theoretical backing of enterprise ontology.

2.2.1 The Ψ-theory

The central theory that constitutes enterprise ontology is the so-called Ψ-theory. The Greek letter Ψ is pronounced as 'psi', which in this context is the acronym for *performance in social interaction*. The Ψ-theory defines an enterprise as a heterogeneous social system. As any system, an enterprise has a set of states and a set of elements. As a social system, the set of elements are social individuals that perform two kinds of acts: production and coordination. Production acts result in a state change (contributing to achieving the purpose or mission of the enterprise), whereas coordination acts are acts that result in the commitment of a production act. A more elaborate view of the Ψ-theory is achieved through its axioms, which serve as a starting point for deducing and inferring the Ψ-theory [12].

The operation axiom

The operation axiom states that the operation of an enterprise consists of activities and actor roles, which are amounts of authority and responsibility, fulfilled by social individuals, i.e. the actors [12]. In the fulfillment of their actor roles, these actors perform two kinds of acts: *production acts* (P-acts) and *coordination acts* (C-acts). The results of these acts are production facts (P-facts) and coordination facts (C-facts), respectively.

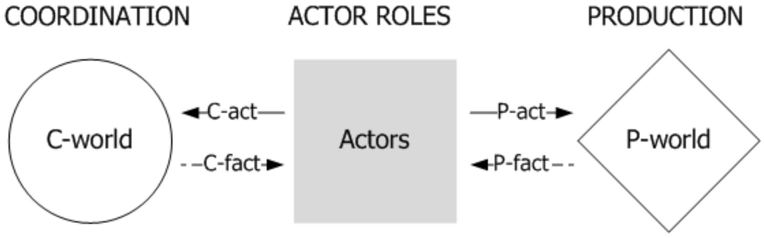


Figure 2.4: Graphical representation of the operation axiom

P-acts are acts that deliver products or services from the enterprise to its environment. Performing a P-act therefore results in something tangible or intangible. Tangible acts are for instance the creation, transportation or storage of a good; an example of an intangible act is a judge sentencing someone in a courtroom, the decision to grant an insurance claim, electing someone as the new President, and so on. Whenever a decision is made as a result of human judgment, a new P-fact is created.

C-acts, on the other hand, do not create anything new. They are however necessary in the preparation of realizing a P-fact. C-acts are performed by one actor, called the *performer*, and directed to another actor, the *addressee.* A part of every C-act is an *intention*, which can be a *request*, *promise*, *statement* or *acceptance.* Connected to this intention, is the *proposition.* This is something that is or would be the case in the production world.

Analogous to the distinction between the two types of acts and facts, two *worlds* are distinguished: the *production world* or P-world, and the *coordination world* or C-world. The state of the P-world is defined as the set of all P-facts that were created over time, and the state of the C-world is defined by all C-facts. The operation axiom is depicted in figure 2.4.

The transaction axiom

The previous section stated that actors within an enterprise perform two kinds of acts: production acts and coordination acts. This section addresses the relationships between these acts. It would seem that for each process, the acts have a totally different sequence. However, as stated on page 14, one of properties of an ontological model, is that it is abstracted to show the *essence* of an enterprise. In the essential, ontological viewpoint, processes within organizations can be considered as paths through some generic coordination pattern. The transaction axiom therefore states that coordination acts a performed in a certain pattern, a so-called *transaction* [12].

Figure 2.5 shows the basic pattern of a transaction. The basic transaction pattern involves two actors roles, the *initiator* on the left side, and the *executor* on the right side.

A transaction consist of three phases. The first phase, the *order phase* (O-phase), start with a request by the initiator, followed by a promise by the executor. In the

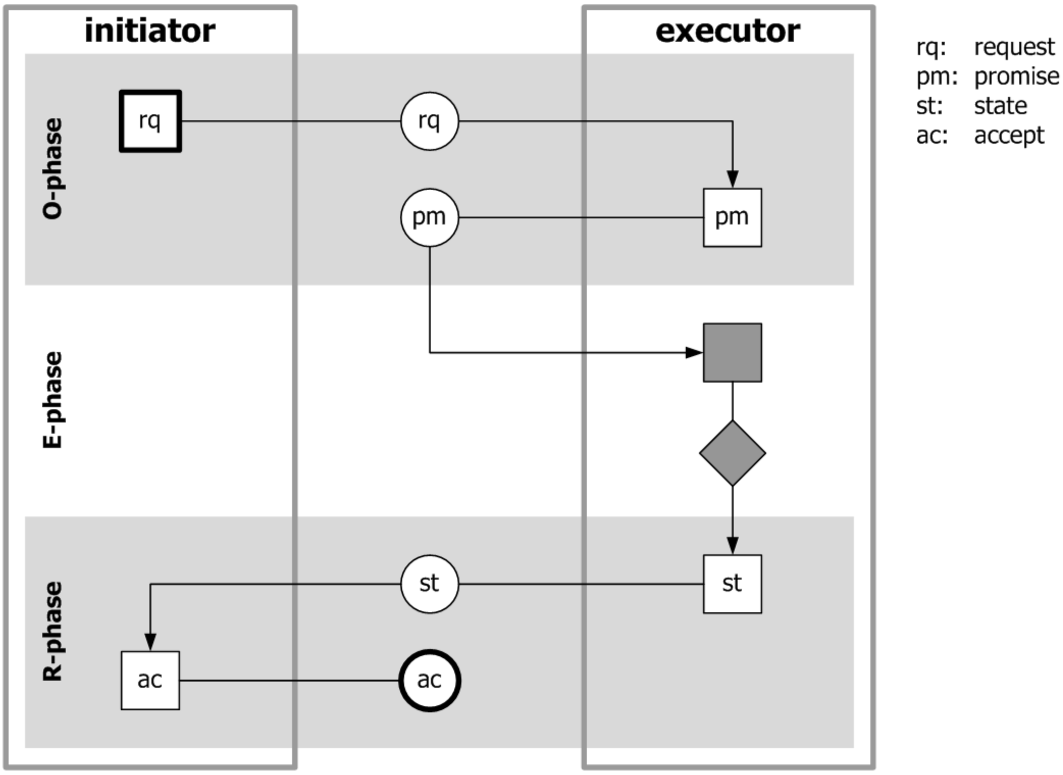


Figure 2.5: The basic pattern of a transaction

execution phase (E-phase), the executor fulfills the promise by actually carrying out the transaction, thereby creating a new P-fact. In the *result phase* (R-phase), the executor of the transaction is stated, after which the initiator accepts the result.

The basic transaction pattern only allows the initiator and executor to agree in each phase of the transaction. In real life, this is not always the case, because the executor might want to decline instead of promise to execute a transaction. Also, the initiator does not necessarily need to always accept the result of a transaction, but may want to decline in some cases. For obvious reasons, the basic transaction pattern is insufficient for all real life scenarios. It is therefore extended into the so-called *standard transaction pattern*, which is shown in figure 2.6.

Although the standard transaction pattern covers a lot more potential scenarios than the basic transaction pattern, it is still lacking a significant type of act: cancellation. There are four cancellation patterns, each of which allows the initiator or the executor of a transaction to revoke an act, resulting in (a part of) the rollback of a transaction. Each cancellation pattern starts with a *cancel* act on which a conditional C-fact is put, connected with a dashed arrow. This means that the cancellation can only be performed if the C-fact exists.

Figure 2.7 shows the cancellation pattern of a request. This pattern can occur whenever the initiator feels remorse about his or her request. If the executor allows

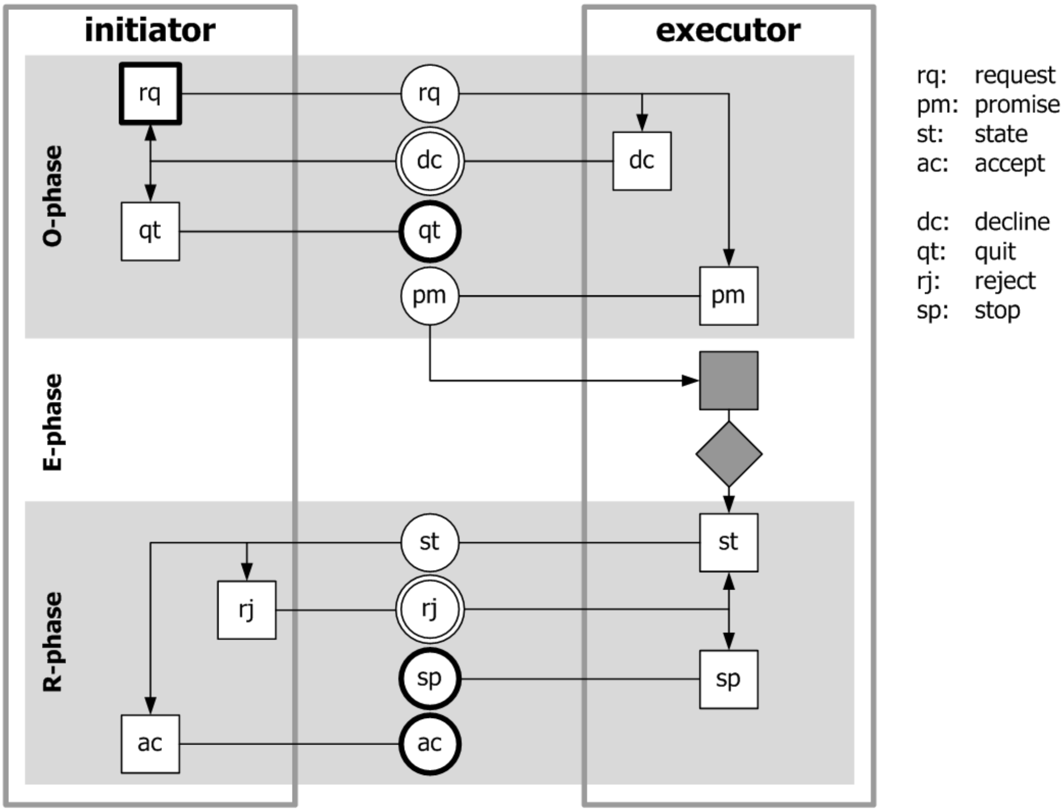


Figure 2.6: The standard pattern of a transaction

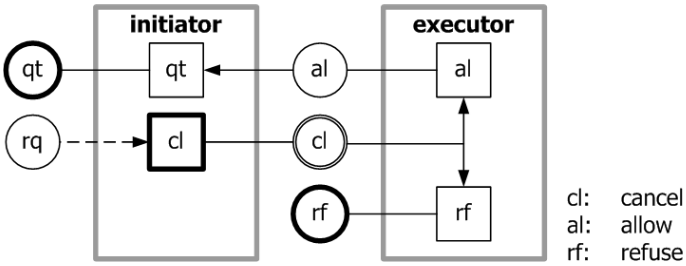


Figure 2.7: Cancellation pattern of a request

(al) this cancellation act (cl), the transaction is quited (qt). However, the executor can block the cancellation by refusing (rf) the cancellation act.

It could also happen that a promise by the executor is canceled (figure 2.8). Now, the initiator can either allow or refuse. However, refusal of a promise cancellation might not help the initiator much, because the initiator does not have any influence on the execution of the transaction.

Figure 2.9 shows the cancellation pattern of a statement. If the initiator allows

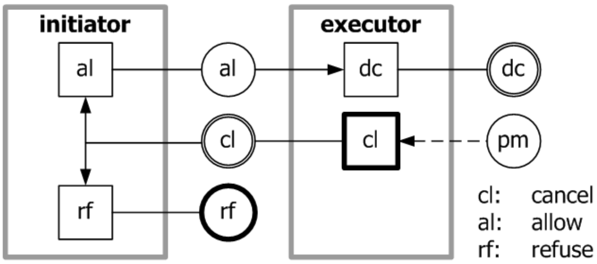


Figure 2.8: Cancellation pattern of a promise

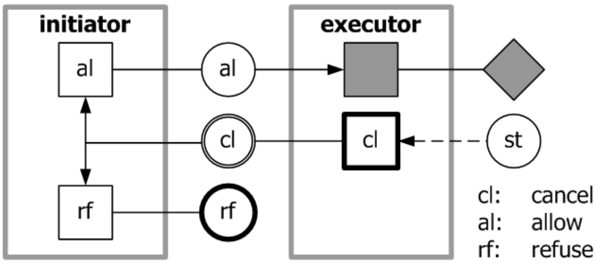


Figure 2.9: Cancellation pattern of a statement

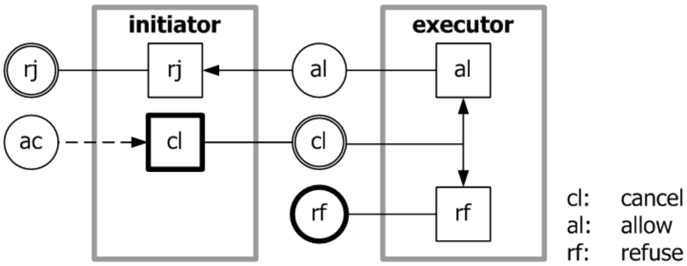


Figure 2.10: Cancellation pattern of an acceptance

this cancellation, the executor is able to redo the production act. However, if the initiator refuses the cancellation, the customer is performing the acceptance act.

If the transaction was successfully completed, the initiator can still perform a cancellation act (figure 2.10). The executor can accept this cancellation act, which ultimately can result in redoing the transaction. If the executor does not accept, there is not much the initiator can do.

In order to cover the complete set of possible scenarios, the standard transaction pattern should be used in combination with the four cancellation patterns. For reasons mentioned in [17], this *complete* pattern may be called *universal*.

The composition axiom

As stated in the previous section, the result of a successful transaction is the creation of a P-fact. Equally, every fact in the P-world is the result of a successful transaction. Some objects, such as a car, are not produced by a single P-fact. Instead, components are produced in a certain sequence, after which these components will be used to construct other, less granular, components. The question is how the transactions for the production of the components, are related to the transaction for the production of the total assembly.

Suppose transaction T1 results in the P-fact of creating a steering wheel. The steering wheel is needed for transaction T2, that results in the P-fact of constructing a bicycle. Logically, T1 has to be completed before T2 can be executed. A solution to this problem can be achieved by adding a dashed arrow from T1's accept state, to the execution act of T2. In this way, T2 has to wait with its execution until T1 is completed (i.e. until T1 is in its accept state).

The existence of sequence within the execution of transactions is the foundation for the composition axiom. It states that every transaction is enclosed in some other transaction, or is a self-activation transaction [12]. This enables the representation of business processes in ontological models, as a business process is a collection of causally related transaction types, such that the initiator is either performed by the actor itself (self-activation) or by an external actor (external activation).

The distinction axiom

The distinction axiom states that the human ability in performing actor roles, can be regarded as a three tier structure: *forma*, *informa* and *performa* [12].

The forma ability deals with the form aspects of communication and information. Both the initiator and the executor of a transaction should be able to understand the C-acts that are performed. In other words, the forma ability is about the syntactic understanding of both actors on either side of the transaction.

The informa ability concerns the content aspects of information and communication. In order to communicate, the initiator should formulate information in a way that the executor can interpret. In other words, the initiator and the executor should semantically be in agreement with each other and share the same thoughts. This is also called intellectual understanding.

Lastly, performa is the actual fulfillment of the mission of the enterprise by the actors, based on the results that can be delivered in the informa layer. It concerns creating something new, which can be something material (products), but also making decisions. The dependencies between the three human abilities in performing a coordination act are shown in figure 2.11.

The organization theorem

The four axioms together constitute the overall goal of the Ψ-theory, which is to extract the essence of an organization from its actual appearance. This section

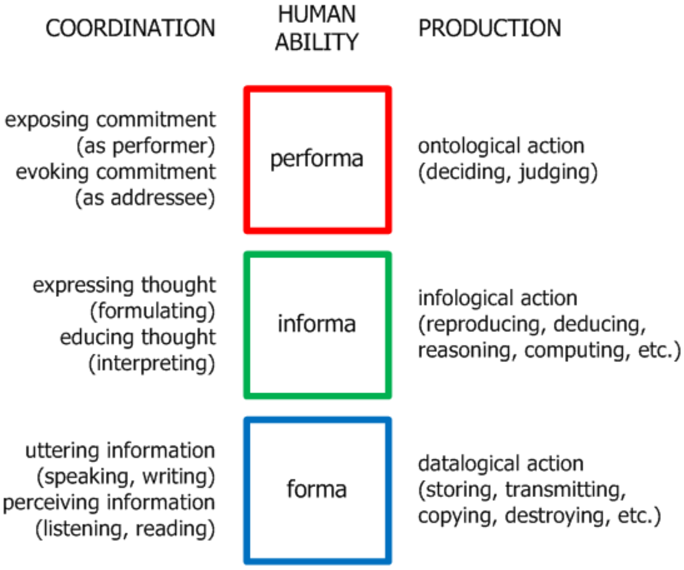


Figure 2.11: Human abilities according the distinction axiom

describes how the benefits of each of the four axioms can be combined to get one consistent, comprehensive, concise and essential model of an enterprise, that can appropriately be regarded as the ontological model of an enterprise.

The organization theorem states that an enterprise can be seen as an organization layered with three heterogeneous tiers [12]. From the top layer to the bottom layer, these are the *business organization* (B-organization), the *intellect organization* (I-organization) and the *document organization* (D-organization). The D-organization and the I-organization support the I-organization and B-organization, respectively, as shown in figure 2.12.

Al the layers in the figure are heterogeneous social systems. This means that they are all alike, as far as the performance of actor roles by social individuals is concerned. However, the layers are different with respect to the production facts that they produce. The B-organization delivers ontological P-facts, the I-organization is infological and the D-organizational creates datalogical P-facts. Higher layers *use* lower layers; lower layers *support* higher layers.

The triangular shape has some interesting characteristics. Firstly, there is nothing above the ontological level, as—in contrast to the other levels—the top width of that layer is equal to zero. Second, the total surfaces of the layers, from top to bottom, are approximately scaled 1:4:7. This represents both the amount of effort on the coordination acts in the subsequent layers, as well as the number of actor roles that exist within the layers.

A very significant distinction is made between the organization *function* and *construction*. There is no intrinsic relation between the two. In other words, the function

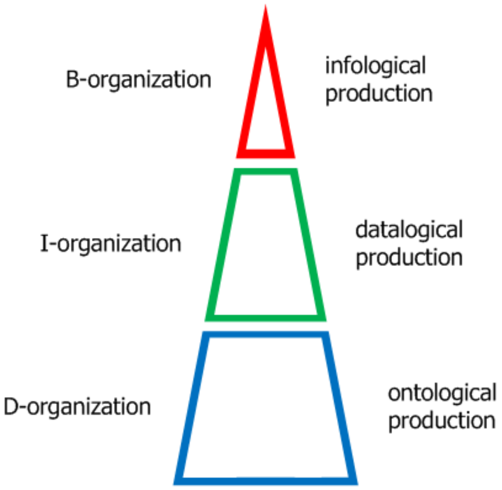


Figure 2.12: Representation of the organization theorem

of the organization is completely independent of its construction. Combined with the notion that layers use and support each other, consider the following example: a treasurer is dividing a bag of money among several divisions of an enterprise. The latter is obviously an ontological P-act. He is not sure which divisions should receive which amount of money. In order to make a good decision, the treasurer requests financial figures, such as the total costs and revenues per division, from the accountant. This is infological, as information on a lower level is processed. These figures are on their turn based on all the financial transactions that were made in a certain period, and therefore the accountant relies on datalogical information. This example shows how a construction of one layer, uses the function of the lower layer.

The organization theorem designates the use of ICT systems within the organization as its *implementation* [18]. Figure 2.13 illustrates the different organizations and ICT systems. ICT systems consist of hardware and software. Software is used to support the different aspect organizations in the enterprise or to take over the acts performed by the actors in these aspect organizations [12]. Software can therefore be classified in the same way as organizations, which entails the following types of applications: B-applications, I-applications and D-applications. Hardware is considered to be the lowest layer supporting the D-application. As

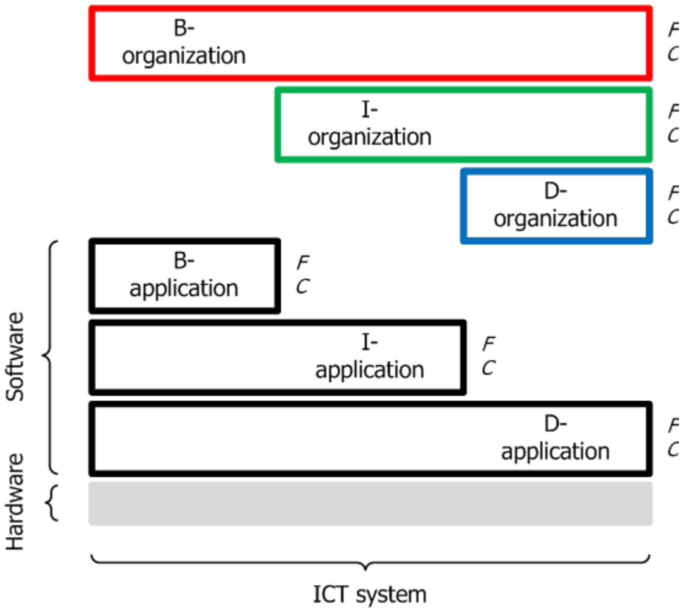


Figure 2.13: Organizations and ICT systems

D-organizations, D-applications concern the storage, transmission or destruction of information. One might think of e-mail systems which enables transfer of digital documents. I-applications use existing information and knowledge in order to obtain new information and knowledge by computation or calculation. The system in the example above is an example of an I-application. B-applications support decision-making within the enterprise. This type of applications is for instance workflow systems. As Figure 2.13 suggests, each underlying ICT systems support the ICT systems above. An I-application can run using several D-applications. A B-application is supported by means of several I-applications [18].

2.2.2 Supplementary theories

In addition to the Ψ-theory, two other theories support the notion of enterprise ontology: the ϕ-theory and the τ-theory. These theories underline the general concept on which enterprise ontology builds.

The Greek symbol ϕ is pronounced as 'FI', which in this context is the acronym for *fact and information*. It provides an understanding of the way data and information are separated and can be applied in all environments, so not just enterprises. [13]

The Greek symbol τ that signifies the τ-theory, is pronounced as 'TAO' and stands for *technology, architecture and ontology*. It provides a profound understanding of the difference between function and construction and how this distinction affects the design process of information systems. [13]

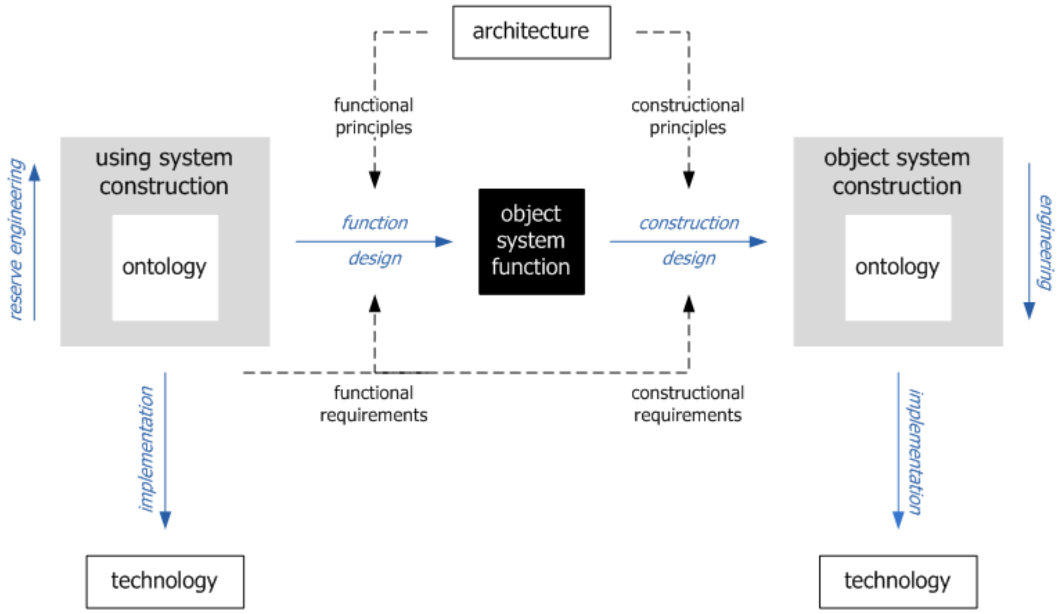


Figure 2.14: Generic system development process

The distinction between function and construction is an important one, as it can be applied in the development of systems of any kind, like information systems. Figure 2.14 provides an overview of the process of developing such a system, referred to as the *object system* (OS), for the sake of supporting another system, the *using system* (US). This so-called *generic system development process* (GSDP) consists of four phases: function design, construction design, engineering and implementation.

Function design starts from the construction of the US and ends with the black-box model of the OS, which contains all functional specifications for the OS to be built. Construction design starts from the specific function of the OS and it ends with the construction of the OS. The next development phase after designing is engineering. Engineering of a system is the activity in which a series of constructional models are produced. Engineering starts from the ontological model and ends with the implementation model. The last phase, implementation, concerns the assignment of technological means to the constructional elements of the implementation model. Thus, once implemented properly, the system can be put into operation.

2.3 Business components identification

The previous section has explained the concept of enterprise ontology and its underlying theory. The field of enterprise ontology is used to abstract an ontological model of an enterprise, that captures the *essence* of its organizational activities [12]. It does not take as-is implementation details into account. For this reason, it is an ideal point of departure for restructuring an enterprise's ICT infrastructure. An ontological model provides all essential information that is necessary in the identification of high level information systems that support the business, so-called *business components* [1]. Furthermore, by using the ontological level (also called 'business' level) as an input for defining these business components, it remains understandable for business people.

This section shows how an ontological model of an enterprise's organization serves as an input for defining business components, by using the *business component identification* (BCI) method. The BCI method's goal is to identify *reusable*, *marketable*, *self-contained*, *reliable* and *manageable* business components.

The identification of business components is the first step in the development of a modern day information system. It uses an ontological model of the organizational activities within an enterprise, and applies a number of steps on this model to capture business components. Key to note is that the derived business components are *not* ICT systems themselves. Rather these are clusters of coinciding functionality. The method tries to find the optimal trade-off between granularity and reusability: if business components become too large, i.e. too much functionality is clustered together, it is hard to reuse parts of functionality. Very small business components are not a good solution either, as that would result in an unmanageable large amount of business components, and the communication between them.

The introduction of this section states that enterprise ontology provides a suitable abstraction of the business model. This is because ontological models fulfill the following requirements necessary for deriving apt business components: *conciseness*, *comprehensiveness*, *consistency* and *essence*[2] [12]. Other techniques, such as Petri nets and activity diagrams, do not distinct between infological and ontological actions. Therefore they include unnecessary implementation details that are undesired in the identification of high level information systems that support *business* processes, not just any other kind of processes [1].

The BCI method works as follows. First, an ontological model should be constructed of the part of the business that is to be supported by the business components. One known methodology for creating such models is the DEMO[3] methodology. The BCI method uses the information objects, process steps and their relations, from the DEMO models. The cardinality of the relationships between process steps expresses how strong the relationship is. If some process steps are always performed subsequently, it is likely that this functionality will be enclosed within one business component. Furthermore, weights are assigned to relationships. For example, the relationship that a process step has with an information object that it creates, and thus *owns*, is much stronger than an optional relationship between two process steps. Once the relationships are identified and weights are assigned, the relationships are

---

[2] these quality requirements are explained on page 14

[3] for more information about this methodology, visit `www.demo.nl`

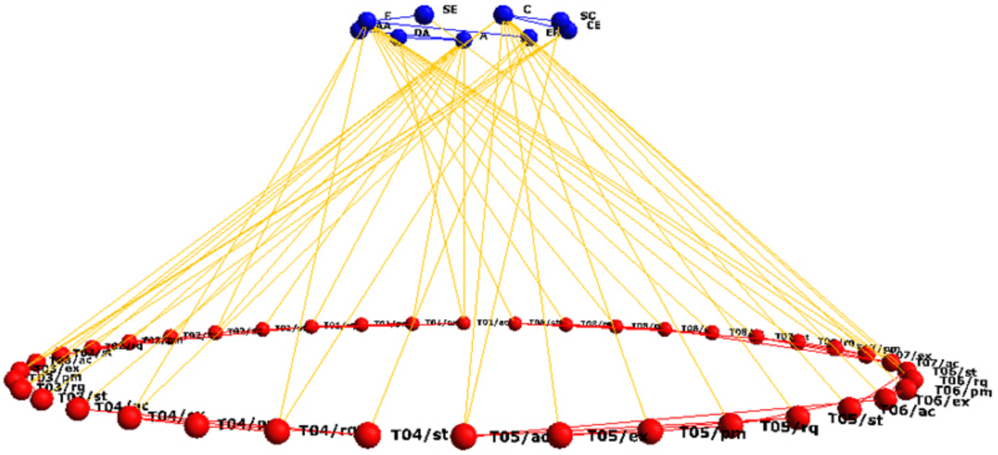


Figure 2.15: Relationships between process steps and information objects

modeled in the BCI method, using a weighted graph [1] (see figure 2.15)

In order to obtain an optimal grouping, an optimization algorithm is used [1]. First, an initial solution is made, which somehow connects each information object and process step. The starting solution is generated using a greedy graph-partitioning algorithm. This solution is further optimized by using the Kernighan and Lin graph-partitioning algorithm. Figure 2.16 shows the results of applying the optimization algorithm on the graph of figure 2.15. This figure was created in a fictional case in October 2007, where an ontological model of an education administration organization was made, after which the BCI method was used to identify five business components [6].

2.4 Conclusions

In the introduction of this document, it was stated that service providers face an extremely competitive environment. New technologies and changing customer demands force service providers to have a well defined strategy and above all effective and efficient business processes. ALU plans to cope with this challenge by achieving economies of scale and synergies in the operations of two large service providers, instead of performing the operations of each one in parallel. A first step in obtaining such synergies is by looking at the business processes.

Using the TM Forum's process framework for the identification of the necessary business processes is key, but insufficient without knowing what is happening

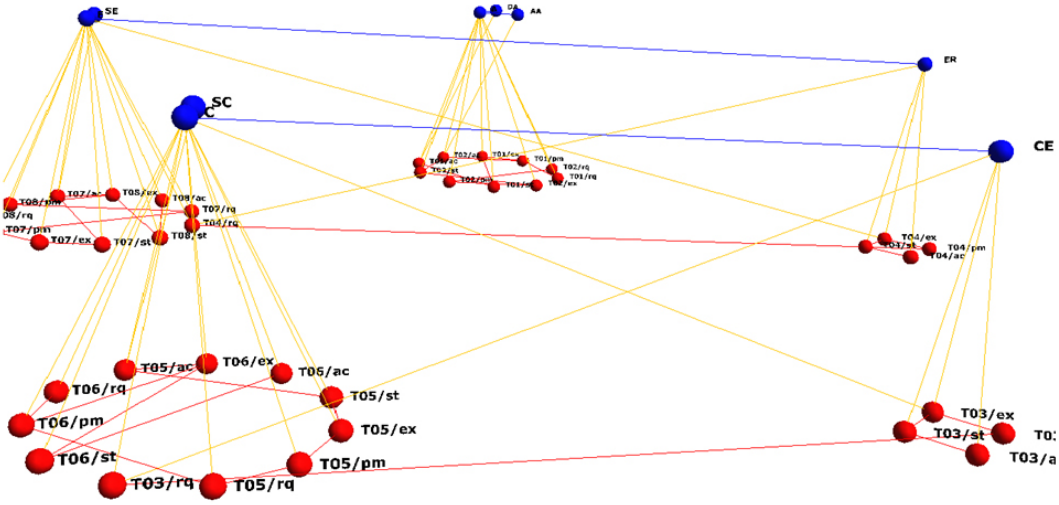


Figure 2.16: Identified business components

on the ontological level. Therefore, enterprise ontology complements the process framework by capturing the essence of common business processes within service providers. Enterprise ontology omits the implementation and is therefore a good point of departure for business process redesign. Furthermore, enterprise ontology fulfills the requirements for using an ontological model as an input in a method for identifying business components, i.e. high-level clusters of functionality that directly relate to ICT services.

As ontology itself does not concern the implementation of systems that support the processes (i.e. anything that happens on the operational level), the BCI method provides the valuable addition of identifying business components. As mentioned before, business components are not ICT systems themselves, but clusters of coherent functionality. The BCI method uses a mathematical optimization algorithm that results in an optimal grouping of business process steps and information objects. As will be demonstrated in a later stage, identifying ICT services that fulfill the identified business component functionality, is relatively easy.

For ALU, the TM Forum applications framework has been a guide for identifying and procuring the necessary applications. However, this is done on the basis of the functional domains, around which the TM Forum solution frameworks evolve. Using ontology to derive the essence of described and actual business processes, in combination with the BCI method, is very promising as—in theory—it provides the most efficient way of utilizing ICT systems.

Chapter 3

Network operations processes

This chapter gives an overview of the processes that are performed at the service desk. The processes are obtained through interviews with employees and documents provided by employees, and are described by means of flowcharts and textual descriptions (section 3.1). This chapter also provides an overview of the ICT systems and services that are used to support the processes (section 3.2). By describing the processes and ICT services, research steps (3) and (4) are performed.

3.1 Network operations processes

When looking at business processes in telecommunications, one widely used framework is clearly predominant: the process framework or *enhanced telecom operations map* (eTOM), being one of the four TM Forum solution frameworks. The framework is used by service providers for defining their business processes and their interrelationships. The process framework covers all core activities of a telecom service provider—fulfillment, assurance and billing—and identifies, for each of those three, processes in the entire supply chain (from customers to suppliers/partners). At ALU, the scope is limited to a subset of this framework, as only the operations and not their customer's end-customer relations and brand management are insourced.

The framework mentioned above is described in more detail in section 2.1.1. The core processes that are used as an input for building ontological models using DEMO[1], are part of the group of activities which are referred to as *assurance*. Among these assurance processes, are several circular processes that monitor the network, i.e. making sure that it is up and running. These processes are mostly supported by computer systems, that carefully monitor anything that happens in the network. Irregular activities are handled by the 1^{st} level support office, also called *front office*, which takes the proper actions. There are five scenarios for which processes are identified:

1. Fault with no customer impact (subsection 3.2.1);

---

[1] Design and Engineering Methodology for Organizations

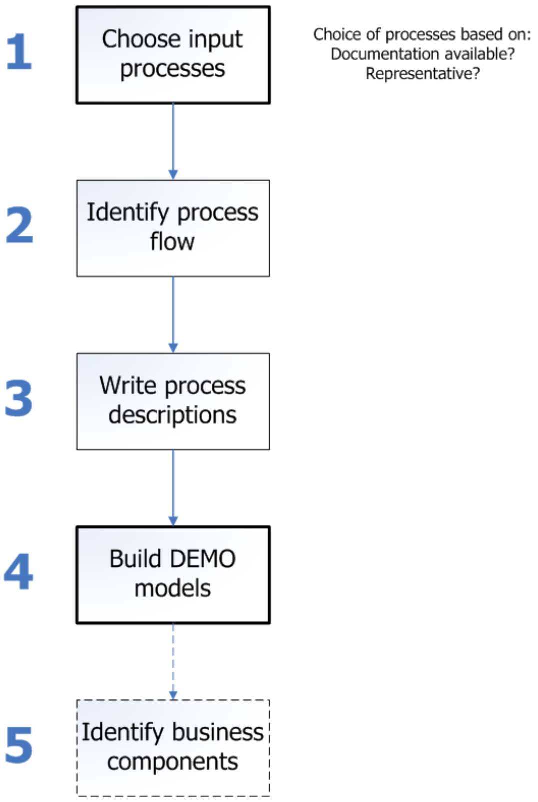


Figure 3.1: Steps towards DEMO models

2. Fault with customer impact (subsection 3.2.2);
3. Fault which the customer has reported (subsection 3.2.3);
4. Major business customer network outage (subsection 3.2.4);
5. Major residential network outage (very similar to subsection 3.2.4, will not be covered in isolation).

3.2 Process flow

Sections 3.2.1 to 3.2.4 describe the processes identified in the previous section, first by giving a textual description, after which the process flow charts are shown. These serve as the input for constructing the ontological models in the next chapter. Figure 3.2 explains the notation used in the charts. The notation used is the ANSI standard flowchart[2].

Multiple processes that are covered in this chapter involve the same authorities or actors. The 1st level support, more commonly known as the *front office*, is often the

---

[2] these models are constructed using Microsoft Visio

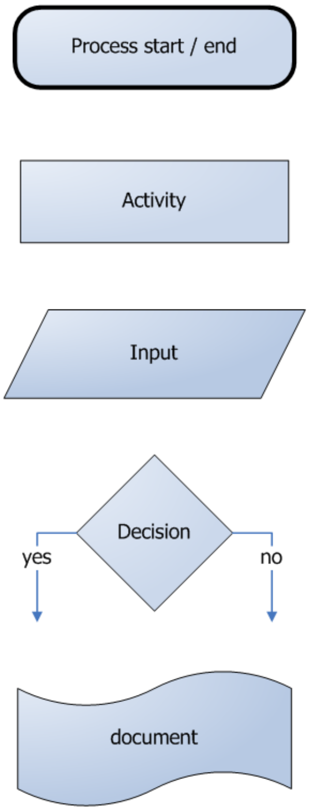


Figure 3.2: Notation used in the process flow charts

starting point in the network operation processes. They provide a service oriented approach, in which their main responsibilities are the 24/7 surveillance and 1^{st} level troubleshooting of network elements and customer services (data, mobile, Internet, voice), fault management (internal and external) and solving problems with the network.

The 2^{nd} level support, or *back office*, is technology oriented. It consists of professionals with certain technological competences. The 2^{nd} levels support is called for in case something needs to be done which requires very specific knowhow and with great proficiency.

The level after the 2^{nd} level (for which there is no name other than 'next level') consists of field services, engineering, ICT, vendors and so on. Note that within next level support, there is no distinction between internal and external parties.

Another collection of entities that are key to the processes described in this section, are the so-called 'trouble tickets'. There are two types of trouble tickets. *Network* trouble tickets (NTTs) are created whenever there is a problem with the network that does not affect the customer directly. For example, a direct connection between two sites might be lost. However, if these sites are somehow connected through other sites, it does not directly influence the service delivered to the customer.

For any fault or liability that directly affects the customer, a *customer* trouble ticket (CTT) is created. By definition, a liability is affecting the customer if it directly decreases the quality of service towards the customer, or if the customer reports a fault at the customer service desk—no matter what the problem is.

While reading the process descriptions, the red-, green- and blue-colored words represent production and coordinations acts and facts in the B-, I- and D-organization, respectively. This is the result of the Performa-Informa-Forma (PIF) analysis, which is the first step of the DEMO methodology. For now, the colors of the words can be ignored, because the results PIF analysis will be applied in the next chapter.

3.2.1 Fault management process: fault with no customer impact

The following process flow description corresponds with the process flow chart in figure 3.3.

A single fault in the network is discovered by one of ALU's automated monitoring systems, which measures, based on the error data, that there is no direct negative influence on the quality of service towards the customer. An employee of the front office (the *monitoring agent*), who constantly monitors the system, requests another employee (the *fault handler*) to take care of the fault. At first, the fault in the network will be handled by the front office (1^{st} level). At this point, a so-called network trouble ticket (NTT) will be created, stating the problem, even though there might not yet be much information available. After filling in the NTT, the front office first checks, based on the number of available people and the knowhow they possess, whether they are able to fix the problem themselves. If they decide to resolve the problem and, after some actions are taken, consider the problem fixed, the NTT will be updated and closed. If they come to the conclusion that they cannot fix the problem, or if they tried to fix it without success, either the back office (2^{nd} level) or 'next' level support will be assigned to fix the problem, which usually concerns problems requiring specific technical expertise.

In a number of cases, the back office is assigned to take over the problem handling. The back office will diagnose the problem further, and if the responsible person cannot fix it (because the problem may be beyond the person's competences), he/she can chose to send a problem solving request to a colleague of the back office, or to someone from the next level support. This could be done either directly by the back office employee, or the employee can contact the front office in order to request for a call-out. If the problem is thought to be solved, the problem is handed back again to the front office.

If the front office directly assigns the problem to next level support, then next level support will take the necessary actions and report back to the front office thereafter.

Problems that were handled by either the back office or next level support, always return to the front office. If the problem is considered fixed, the NTT will be updated and closed. If not, the process of assigning the back office or next level support will continue, until the process eventually reaches the state in which the problem is regarded as 'solved' by the front office.

It is also possible that a back office employee discovers a fault while in the field. In this case, the back office will create the NTT and takes the necessary steps in order to solve the problem. However, the front office is responsible for delivering a quality service and therefore is the only authority that can decide to close tickets.

3.2.2 Fault management process: fault with customer impact

The following process flow description corresponds with the process flow chart in figure 3.4.

A single fault in the network is discovered by one of ALU's automated monitoring systems, which measures, based on the error data, that there is a direct negative influence on the quality of service towards an end-customer of Sunrise. An employee of the front office (the *monitoring agent*), who constantly monitors the system, requests another employee (the *fault handler*) to take care of the fault. The authority of Sunrise responsible for the quality of the network in the eyes of Sunrise's customers (which is the end-customer), is the helpdesk. Soon after the ALU front office has informed the Sunrise helpdesk, the helpdesk verifies the problem at the end-customer. If the end-customer has confirmed the problem, the helpdesk creates a customer trouble ticket (CTT). The CTT is shared among the helpdesk and all ALU support levels. The helpdesk can perform some basic actions that are sufficient for minor problems (for example, "reset the router"). When the helpdesk believes that they have solved the problem, the customer is asked to confirm the repair, and if confirmed, the CTT will be closed. If the helpdesk does not have the confidence that they can solve the problem, or if the helpdesk made an unsuccessful attempt to fix the problem, the ALU front office will be requested to hand over the problem.

The front office first checks, based on the number of available people and the knowhow they possess, whether they are able to fix the problem themselves. If they decide to resolve the problem and, after some actions are taken, consider the problem fixed, the NTT will be updated and closed. If they come to the conclusion that they cannot fix the problem, or if they tried to fix it without success, either the back office (2^{nd} level) or 'next' level support will be assigned to fix the problem, which usually concerns problems requiring specific technical expertise.

In a number of cases, the back office is requested to take over the problem handling. The back office will diagnose the problem further, and if the responsible person cannot fix it (because the problem may be beyond the person's competences), he/she can chose to send a problem solving request to a colleague of the back office, or to someone from the next level support. This could be done either directly by the back office employee, or the employee can contact the front office in order to request for a call-out. If the problem is thought to be solved, the problem is handed back again to the front office.

If the front office directly assigns the problem to next level support, then next level support will take the necessary actions and report back to the front office thereafter.

Problems that were handled by either the back office or next level support, always return to the front office. If the problem is considered fixed, the NTT will be updated and closed. If not, the process of assigning the back office or next level support will continue, until the process eventually reaches the state in which the problem is regarded as 'solved' by the front office.

Key to note is that all have abilities to update the CTT, but the helpdesk is the only authority that can decide to close the CTT. During the whole process, compliance with the SLA is constantly monitored, in order to make sure that urgency is at an appropriate level.

3.2.3 Fault management process: customer reports fault

The following process flow description corresponds with the process flow chart in figure 3.5.

A fault in the network is discovered by the customer and the customer requests the helpdesk to repair the fault. The authority of Sunrise responsible for the quality of the network in the eyes of Sunrise's customers (which is the end-customer), is the helpdesk. The helpdesk creates a customer trouble ticket (CTT). The CTT is shared among the helpdesk and all ALU support levels. The helpdesk can perform some basic actions that are sufficient for minor problems (for example, "reset the router"). When the helpdesk believes that they have solved the problem, the customer is asked to confirm the repair, and if confirmed, the CTT will be closed. If the helpdesk does not have the confidence that they can solve the problem, or if the helpdesk made an unsuccessful attempt to fix the problem, the ALU front office will be requested to hand over the problem.

The front office first checks, based on the number of available people and the knowhow they possess, whether they are able to fix the problem themselves. If they decide to resolve the problem and, after some actions are taken, consider the problem fixed, the NTT will be updated and closed. If they come to the conclusion that they cannot fix the problem, or if they tried to fix it without success, either the back office (2^{nd} level) or 'next' level support will be assigned to fix the problem, which usually concerns problems requiring specific technical expertise.

In a number of cases, the back office is requested to take over the problem handling. The back office will diagnose the problem further, and if the responsible person cannot fix it (because the problem may be beyond the person's competences), he/she can chose to send a problem solving request to a colleague of the back office, or to someone from the next level support. This could be done either directly by the back office employee, or the employee can contact the front office in order to request for a call-out. If the problem is thought to be solved, the problem is handed back again to the front office.

If the front office directly assigns the problem to next level support, then next level support will take the necessary actions and report back to the front office thereafter.

Problems that were handled by either the back office or next level support, always return to the front office. If the problem is considered fixed, the NTT will be updated and closed. If not, the process of assigning the back office or next level support will continue, until the process eventually reaches the state in which the problem is regarded as 'solved' by the front office.

Key to note is that all have abilities to update the CTT, but the helpdesk is the only authority that can decide to close the CTT. During the whole process, compliance with the SLA is constantly monitored, in order to make sure that urgency is at an appropriate level.

3.2.4 Major business customer network outage

The following process flow description corresponds with the process flow chart in figure 3.6.

This process is slightly different from the ones described above. Whereas regular fault management processes run rather frequently, major outages only occur a few times each year, on average. However, processes for major outages are essential, as major outages can result in huge SLA penalties and can severely damage the company's reputation.

Major outages are always discovered by either the customer or by the front office. In case the customer discovers the outage, the customer calls the helpdesk and the helpdesk informs the front office[3]. The first thing to do is to determine, based on the available fault data whether the liability is indeed a major outage or a 'regular' network fault. In the latter case, the fault is resolved by passing through the generic fault management process (subsection 3.2.3) and this major network outage process ends. The remainder of this process flow description applies in case the fault is considered a major outage.

Next, the corporate communications department of ALU is notified. They will inform both the relevant ALU employees, as well as external channels, such as the media. Next to notifying corporate communications, it will be determined whether the major outage is critical. This depends on the customer volume, impact on the revenue (by checking SLA boundaries) and the potential harm to the company's reputation. If the major outage is considered critical, the crisis management process will start in parallel (this process is not documented and goes beyond the responsibility of the service desk). If the major outage is not critical, ALU's technical competence center (consisting of 1^{st} and 2^{nd} level support employees) will come up with another solution until the problem is solved. After each update, both the end-customer and ALU's corporate communications department will be given a heads up of the problem status and next steps.

---

[3] although communication between end-customers and the front office always runs through the helpdesk, the helpdesk is omitted in the flow chart (figure 3.6) for clarity reasons

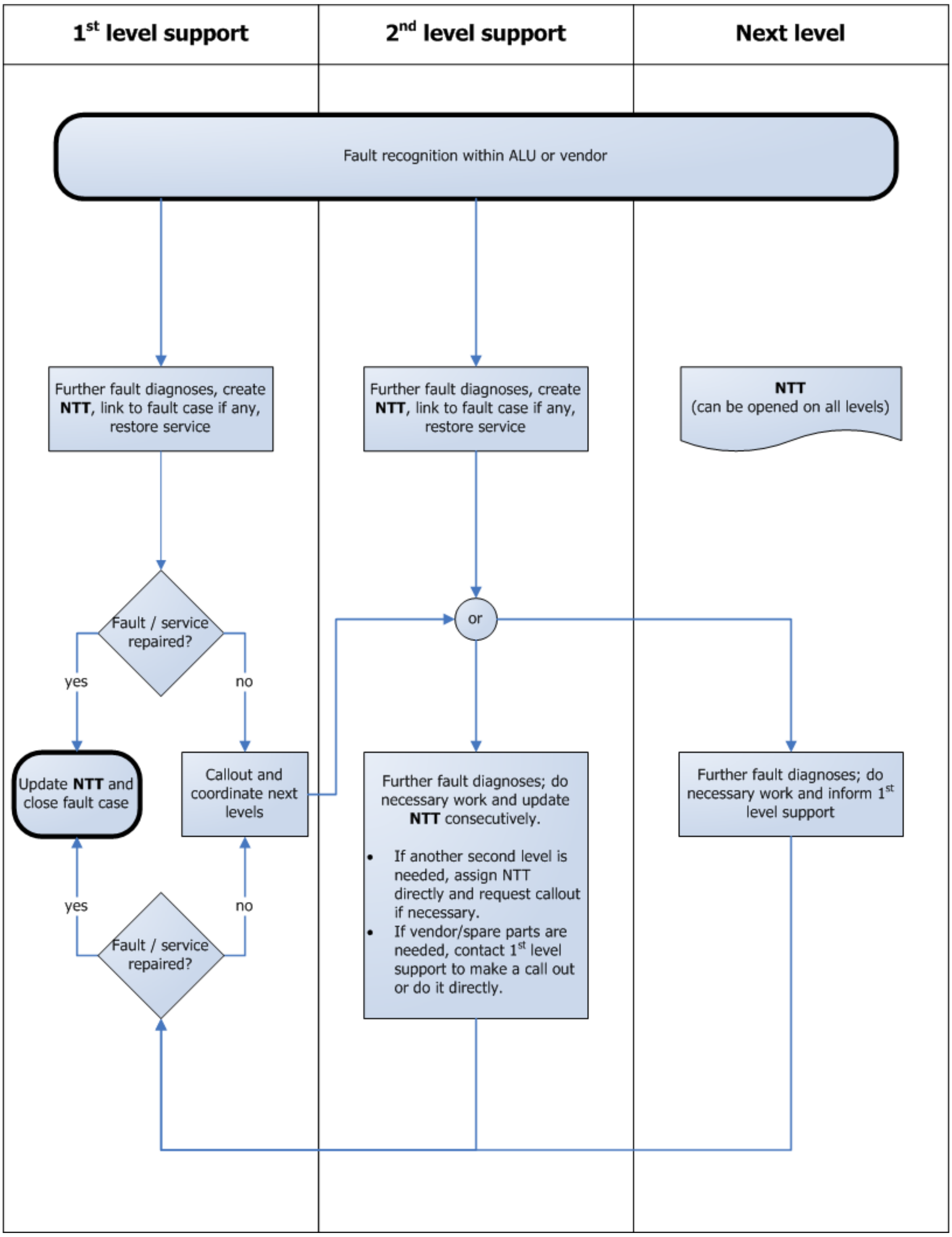


Figure 3.3: Fault management process: fault with no customer impact

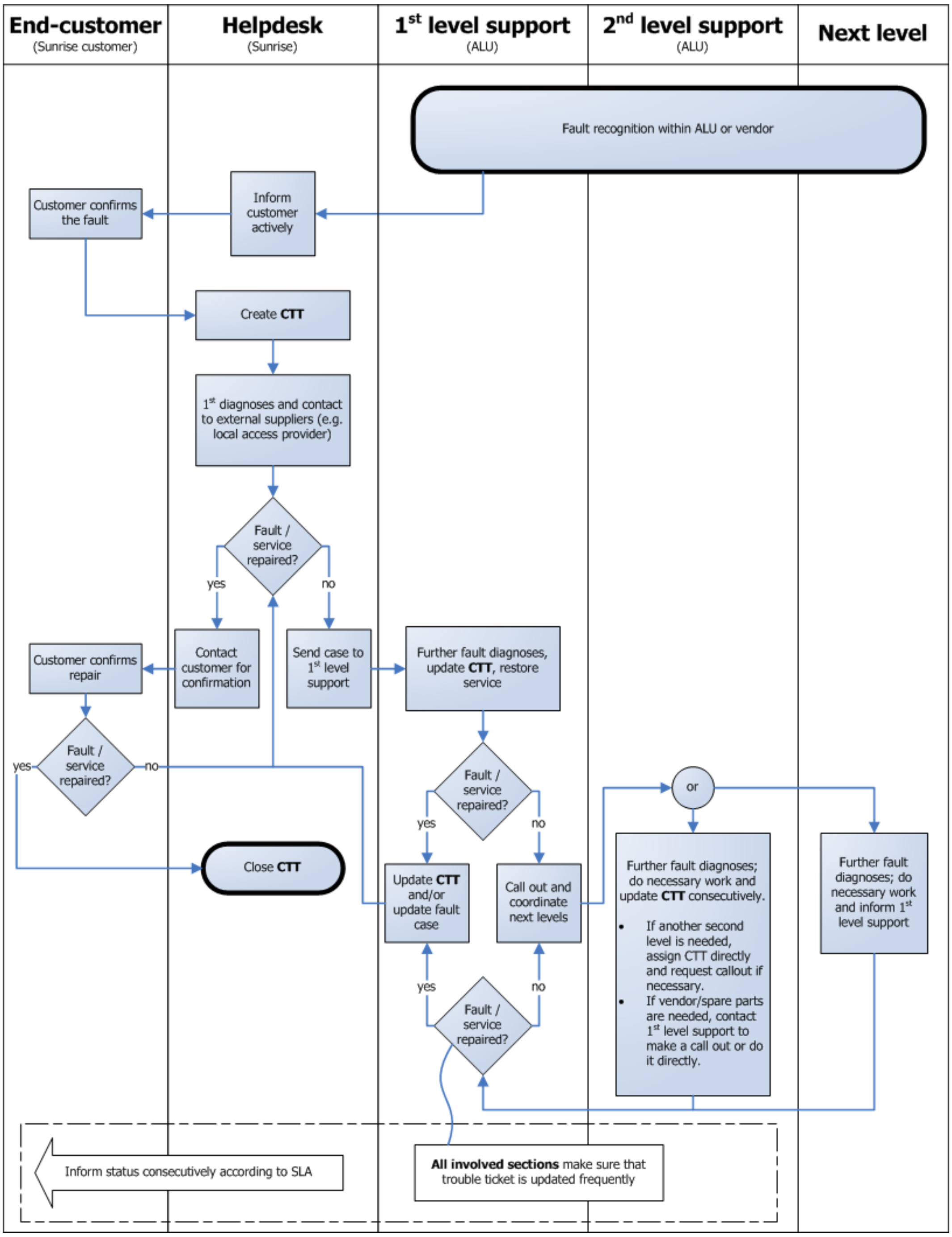


Figure 3.4: Fault management process: fault with customer impact

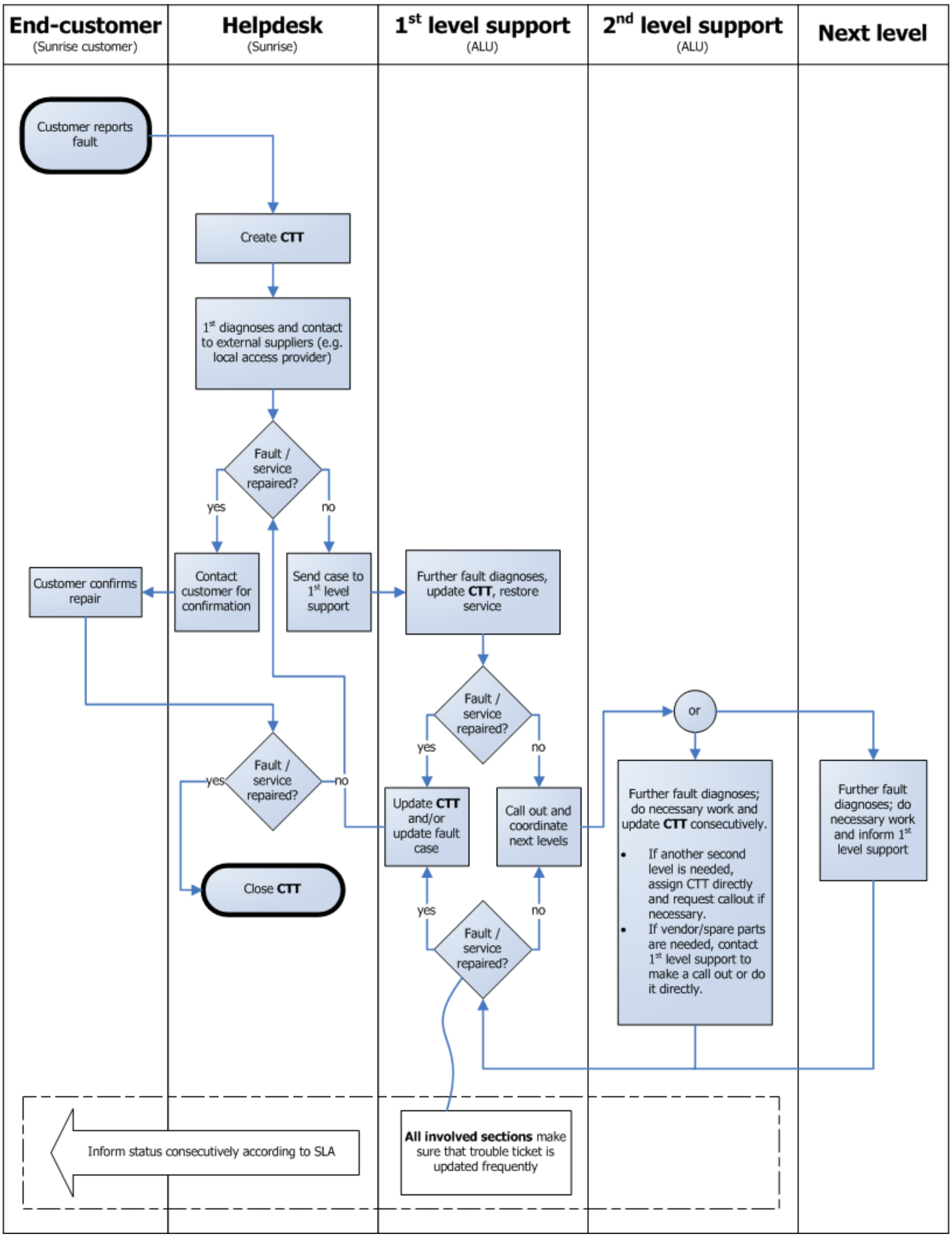


Figure 3.5: Fault management process: customer reports fault

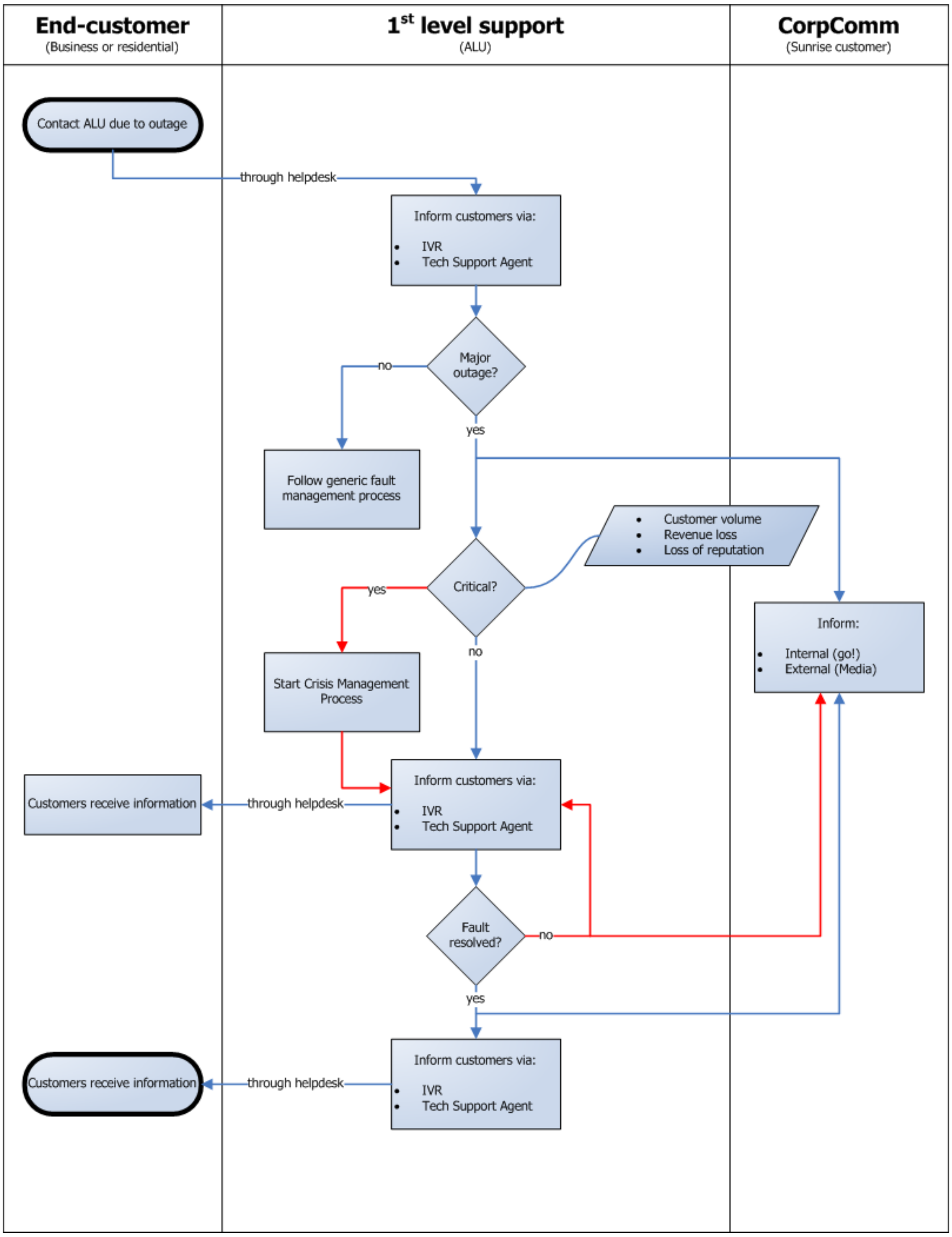


Figure 3.6: Major business customer network outage

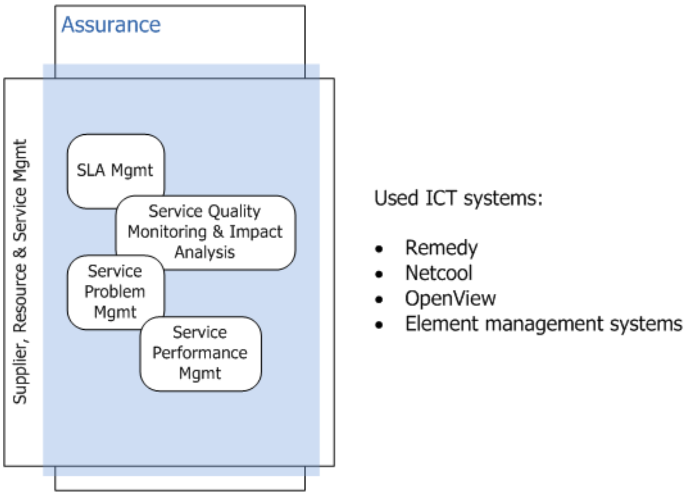


Figure 3.7: Applications used in network monitoring

3.3 Supporting ICT systems

This section focuses on the ICT systems used by ALU for monitoring the telecommunication network for Sunrise. Chapter 2 already introduced the process framework and applications framework, which serve as guidelines in defining business processes and ICT functionality, respectively. Figure 2.3 on page 14 showed the process framework in detail level 1, meaning that it already distinguishes between vertical process groups within network operations. The blue marked area in figure 2.3 denotes the processes that are currently insourced by ALU and which are implemented to serve Sunrise, the second largest TSP in Switzerland. These processes were described in the previous sections.

The service desk at ALU, which hosts the front office, acts as the bridge between resource problems (e.g. network problems) and customer affecting issues. Customer problem resolution, service problem resolution and resource problem resolution can be seen as one distributed set of related applications, built around common data and functions. Each takes a different view of the same data. Figure 3.7 shows a piece of the applications framework, in which applications areas are mapped to ICT systems that ALU uses for their network monitoring activities.

The lowest layer within the network monitoring processes is that of the network and its elements. Elements are pieces of hardware, which could made by ALU as well as other vendors. Each vendor uses its own element management system (EMS). An EMS monitors its attached elements in the network for faults and forwards error messages to Netcool, a system that deals with these faults. An EMS is also used to locate the elements that contain a fault and in 70% of the cases, the fault can directly be repaired from the remote location, i.e. directly from the service desk.

Netcool is an umbrella alarming tool that collects and processes event and alarm information in real-time from many different network data sources (e.g. routers, base stations) and presents a simplified view of this information to operators and administrators. It receives those messages from various EMSs, one for each vendor. It sets one standard for incoming messages, which is the Simple Network Management Protocol (SNMP). Incoming warnings and error messages can be filtered on any kind of field, such as priority, severity, vendor, elements type, and so on. Unlike with an EMS, it is not possible to solve faults in Netcool, as Netcool is only a tool for collection alarms. However, it enables the first step in solving the problem, which is the creation of a trouble ticket (using two mouse clicks in Netcool), which will be further managed by Remedy.

OpenView is a product series of HP, existing of a wide variety of network and system management tools. ALU uses this tool in the same fashion as Netcool, but OpenView is considered as a legacy system. Due to technical and financial reasons, the system is still not completely replaced by Netcool.

The primary purpose of Remedy is to provide a central tracking system, in this case for NTTs and CTTs. Functionality includes sending notifications to groups of support staff members by various means (e-mail, pager) as soon as a NTT/CTT is entered into the system and assigned to them. It is also possible to notify the customer via e-mail when a new trouble ticket is created, and when that ticket is placed in various states such as ‘work in progress’, ‘pending’, ‘resolved’ and ‘closed’. It enables the customer to decide whether or not the problem was resolved, and if not, allows them to easily reopen the ticket, which automatically notifies the support staff of that fact. Furthermore, it allows customers to enter their own trouble tickets into interactive web forms that are forwarded directly to the ALU front office. In order to solve recurrent problems quicker, Remedy stores solutions to problems in a database. Besides creating and tracking trouble tickets, Remedy is used for KPI measurements, which is done in order to determine the level of SLA compliance in a later stage.

The ICT systems and services that are described above are used to support one or more steps within the network monitoring processes. These systems are connected to some extent, but not completely. ‘Old fashioned’ phone calls are still commonly used for communication between the different support levels. Remedy could potentially be able to pass trouble tickets from and to responsible support staff members, but this will always be communicated first over a telephone line. Also callouts for field services are communicated over telephone, as well as the communication between ALU and the Sunrise helpdesk.

In this thesis, it is not only important to know which ICT systems are currently in place to serve the monitoring processes, but also the relationships between the systems, the systems and the staff members and between the staff members themselves. This is because it gives an overview of the ICT integration layer, which allows for interconnectivity across business processes [8]. In a later stage, the existence or lack of integration between these ICT systems is compared with the way in which business components cluster blocks of supporting functionality.

Table 3.1 lists the process steps, gathered from the process descriptions and the flowcharts. Each process step is mapped to the ICT system that supports the process step. Furthermore, the last column denotes the data connection to another ICT system, which (partially) automates the transition between process steps. The steps are mapped to an ICT system as follows:

- Step 1 is done in the background by the EMSs. Resulting errors from this step are passed to either Netcool or OpenView.
- In general, these systems are used to monitor the network (step 2), which basically comprises of monitoring agents who look at a large screen, showing the graphical user interface of Netcool or OpenView. Faults are filtered and categorized first, also done by Netcool or OpenView (step 3).
- Step 4 to 6 include assigning and requesting support staff. The monitoring agent in this case calls the responsible person, informs him and directly communicates the ticket number over telephone. The monitoring agent also assigns the fault to the responsible person within Remedy, the trouble ticketing system.
- Step 7-8 are solely communication acts, using telephone.
- Step 9 and 11 to 14 include diagnosing and resolving faults. Diagnosing for instance includes pointing out the exact location of faults, which is done in the EMS that monitors the network element that contains the fault. As mentioned before, in 70% of the cases, the fault can directly be fixed. For example, a router the be reset from distance within the EMS.
- Step 10 is checking for resources, which is handled over telephone.
- Step 15 to 20 concern trouble tickets, which is almost completely handled by Remedy. The creating of trouble tickets is done directly from Netcool/OpenView, although tickets are always created *within* Remedy.

Table 3.1: Mapping between process steps and ICT systems

#	**Process step**	**ICT system**	**Link**
1	Detecting network fault	EMS	Netcool/OV
2	Monitoring the network	Netcool/OpenView	
3	Measuring error severity	Netcool/OpenView	
4	Requesting fault handler	Telephone and Remedy	
5	BO assigns NLS to solve problem	Telephone and Remedy	
6	FO assigns BO/NLS to solve problem	Telephone and Remedy	
7	BO contacts FO	Telephone	
8	Performing callout	Telephone	
9	FO diagnoses problem	EMS (only if remote)	
10	FO checks for resources to solve problem	Telephone	
11	FO resolves problem	EMS (only if remote)	
12	BO diagnoses problem	EMS (only if remote)	
13	Bo resolves problem	EMS (only if remote)	
14	NLS resolves problem	EMS (only if remote)	
15	Creating NTT	Netcool/OpenView	Remedy
16	Modifying NTT	Remedy	
17	Closing NTT	Remedy	
18	Creating CTT	Netcool/OpenView	Remedy
19	Modifying CTT	Remedy	
20	Closing CTT	Remedy	

Chapter 4

Ontological models

The previous chapters focused on the processes and supporting ICT services for monitoring telecommunication networks at the ALU service desk. It was shown how several industry standards were used, both on the business and the ICT side.

Chapter 3 demonstrated the implemented processes in a flowchart, accompanied by textual descriptions on page 30 to 33, which provided additional information. Note that this text contains several red-, green- and blue-colored segments. These segments represent production and coordinations acts and facts in the B-, I- and D-organization, respectively, and is the result of the Performa-Informa-Forma (PIF) analysis. This analysis will be used throughout this chapter, as a starting point for devising models of the B-, I- and D-organization.

Readers who are not familiar with the theory that supports enterprise ontology, are advised to consult the background chapter (chapter 2), or to review appendix B for a description of the different models of DEMO[1], the methodology used to construct ontological models.

The models for the B-organization are shown and described in section 4.1. Section 4.2 describes the I-organization and finally the D-organization is described in section 4.3. This chapter performs research step (5).

4.1 Modeling the B-organization

In order to model the B-organization, the steps in appendix B were followed. Figure 4.1 shows the different aspects models that are created.

4.1.1 The Interaction Model

The Interaction Model (IAM) is the first model and is expressed in the actor transaction diagram (ATD; shown in figure 4.2) and the corresponding transaction result table (TRT; shown in table 4.1). The ATD and TRT are the result of a series of

---

[1] Design and Engineering Methodology for Organizations

steps that filter only the relevant aspects of business processes and structure those steps in transactions.

The PIF-analysis has resulted in the identification of process steps that cope with the performa ability of human beings. Although the sum of the amount of performa items from the PIF analysis would suggest a large number of transactions, the number of transactions in the TRT is limited to six. This difference is due to two reasons. Firstly, many process steps form a single transaction, which means that the number of transactions is smaller than the number of process steps. Secondly, different process steps which essentially are the same and involve the same actor roles, are also considered part of the same transaction. In this way, multiple alternatives in performing process steps are treated as one, meaning that implementation details are not taken into account.

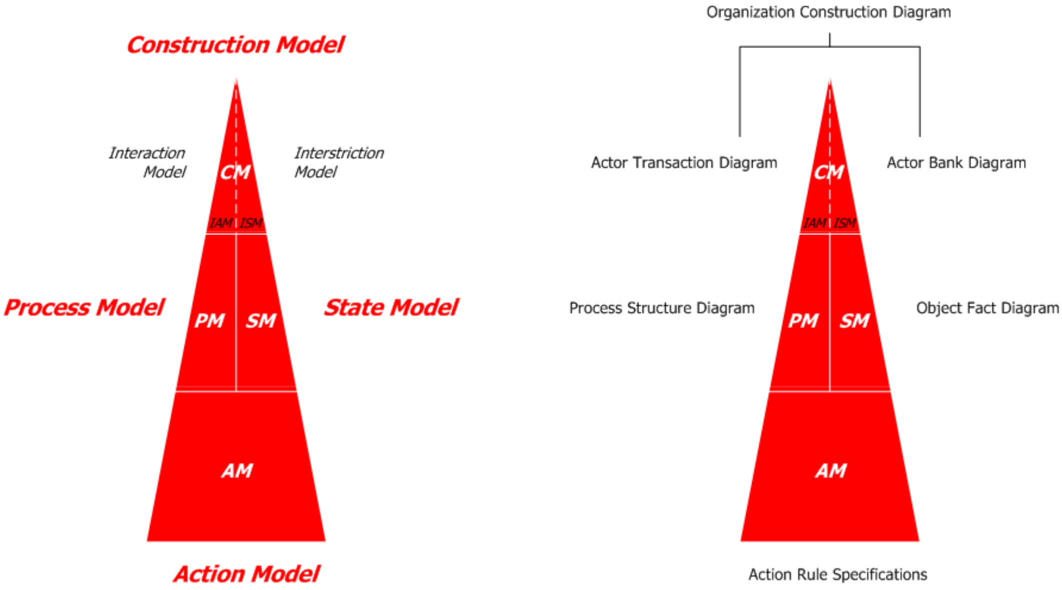


Figure 4.1: Models in the B-organization

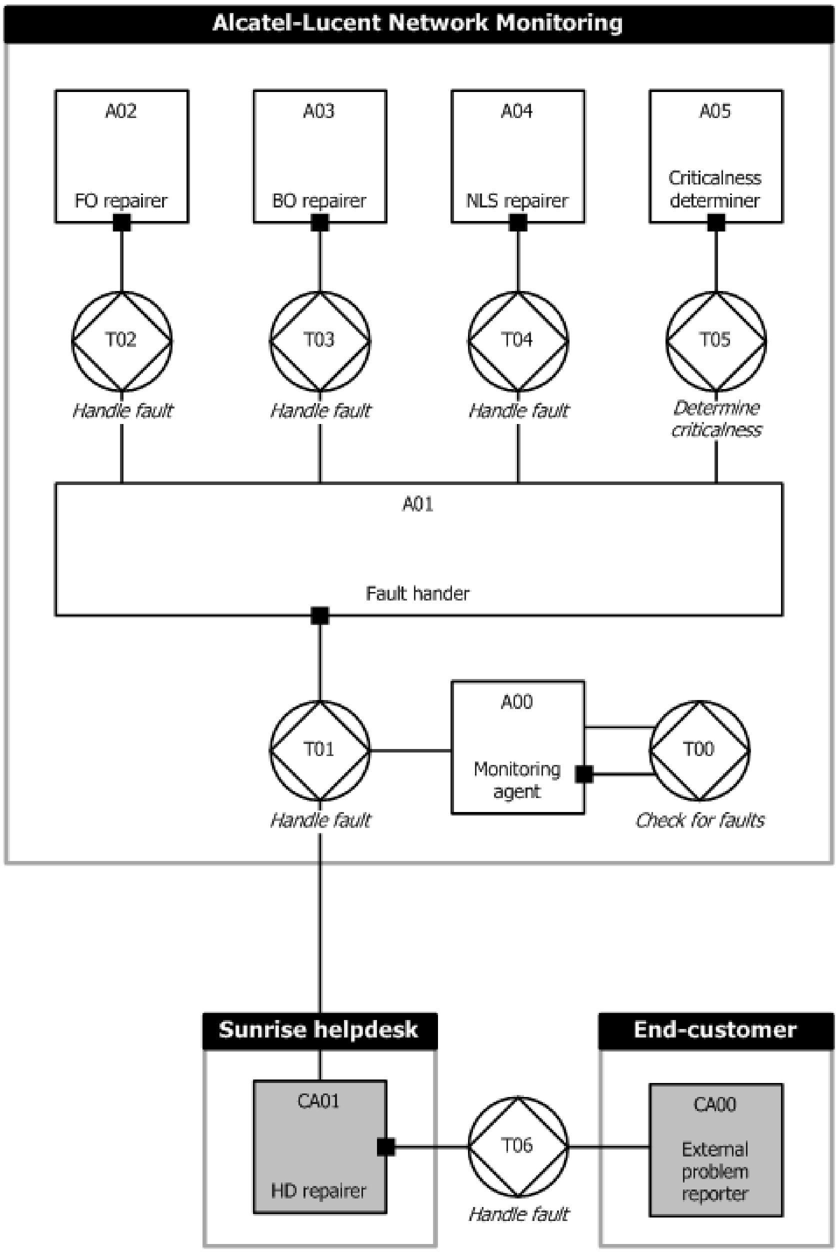


Figure 4.2: (B) Actor Transaction Diagram

Table 4.1: (B) Transaction Result Table

Transaction		**Transaction result**	
BT00	check_for_faults	BR00	The network has been checked for faults F*
BT01	handle_fault	BR01	The fault F has been handled
BT02	fo_handle_fault	BR02	The front office fault FF is handled
BT03	bo_handle_fault	BR03	The back office fault BF is handled
BT04	nls_handle_fault	BR04	The next level support fault NF is repaired
BT05	determine_criticalness	BR05	The criticalness of fault F has been determined
BT06	hd_handle_fault	BR06	The fault F is handled by the helpdesk

The transactions in figure 4.2 cover all the performa (red-colored) items from the process descriptions in chapter 3. The ATD contains seven transactions, numbered T01 to T06:

T0 This is a self-initializing transaction which checks the networks for faults. Although this transactions is performed by an automated monitoring system, it is still regarded as a transaction in the B-organization, because a human being, in this case the monitoring agent, is responsible for the discovery of network faults.

T1 This transaction is the root of handling a fault within the service desk. The transaction is either requested by the monitoring agent who, as a result of T00, discovered a fault in the network, or the transaction is requested by the helpdesk of Sunrise. The latter is the case in two scenarios:

1. The helpdesk of Sunrise has been informed by ALU about a fault with customer impact, but attempts of the helpdesk to solve the fault have failed.
2. Customer has reported a fault to the helpdesk of Sunrise, but attempts of the helpdesk to solve the fault have failed.

T2 Once a fault is assigned to the fault handler (who is a front office employee), the fault handler has to decide whether the front office or another support level should take care of the fault. Transaction T02 is requested within the front office, if they feel that they are capable of dealing with the problem themselves.

T3 If T02 is not requested, the logical choice is to call out for help on the next layer, which is the back office.

T4 This transaction is requested as a last resort to solve the problem, i.e. when both the front and back office have either failed to solve the problem or forwarded the problem to the next level support.

T5 If a discovered fault includes a major outage, it is possibly critical. In this case, the fault handler requests the criticalness determiner to investigate whether the major outage is indeed critical.

T6 If a customer reports a fault to the helpdesk of Sunrise, transaction T06 is performed. This is not relevant for ALU, as this transaction lies completely outside of the large rectangle in figure 4.2, which denotes the service desk of ALU and its operations.

Although there are five distinct processes, there are large similarities between them. For example, the question whether a network fault is discovered by the service desk or by the customer, does hardly have any influence on the way in which the fault is handled. Neither does it matter—on the B-side—whether a *network* or *customer* trouble ticket is created.

4.1.2 The Process Model

The next level of detailing is done in the Process Model (PM), consisting of the Process Structure Diagram (PSD) and the Information Use Table (IUT). The ATD is further specified in the PSD, which is shown in figure 4.3. The transactions from the ATD are divided into process steps. The subset of the the complete transaction pattern is used. This subset comprises more than the basic transaction pattern, as transactions may be rejected. However, the PSD omits cancellations, as these are not common and certainly not applicable in these processes. The PM also contains causal and conditional relationships between transactions. Although the PSD is rather straightforward, some explanation about its structure is given hereunder:

- As mentioned before, T00 is a self-activiting transaction. In the PSD, this is denoted by the circular arrow from T01/rq to itself.
- T01 can be requested by both A00 (monitoring agent) and CA01 (Sunrise helpdesk), which can be seen by the two incoming lines for T01/pm.
- The line from T01/pm to T05/pm, with cardinality 0..1, implies a boolean value, which is whether or not the fault is major.
- The two dashed lines are conditional links for T01/ex, because in case one of transaction T02 to T05 is initiated, T01 can only be executed after a result of the initiated transaction.

Although links between transactions indicate minimum and maximum numbers of next or previous transactions required, it omits the rules that are required to exactly understand the flow of the process. This is done in the Action Model (AM), which is described in the next subsection.

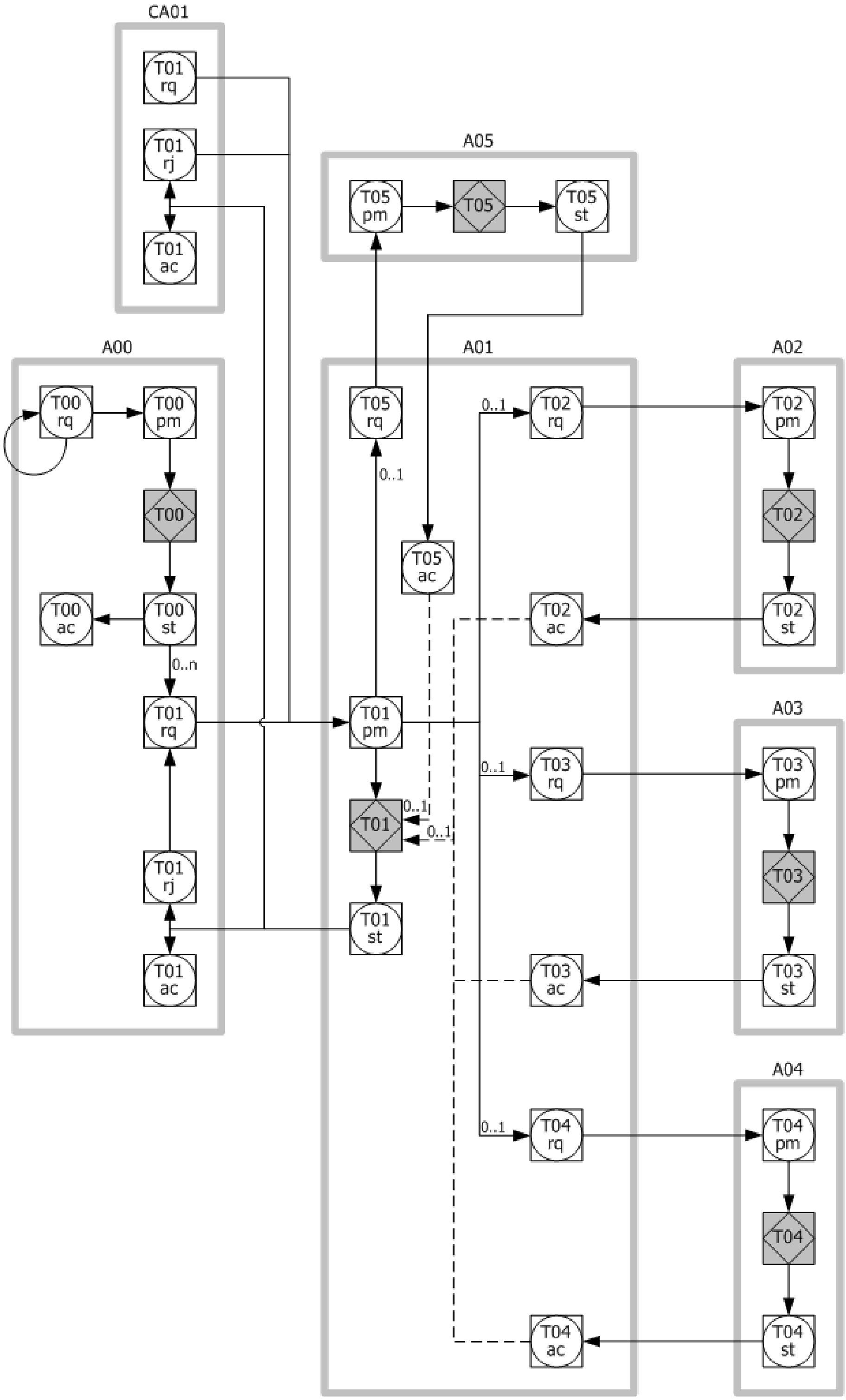


Figure 4.3: (B) Process Structure Diagram

The Information Use Table (IUT) is shown in table 4.2. The IUT specifies for every object class, fact type and result type, in which steps of the PSD instances of it are used. These steps are derived from the AM in subsection 4.1.3. The IUT has been made after the completion of the SM and the AM. It has been shown in this section because it belongs to the PM.

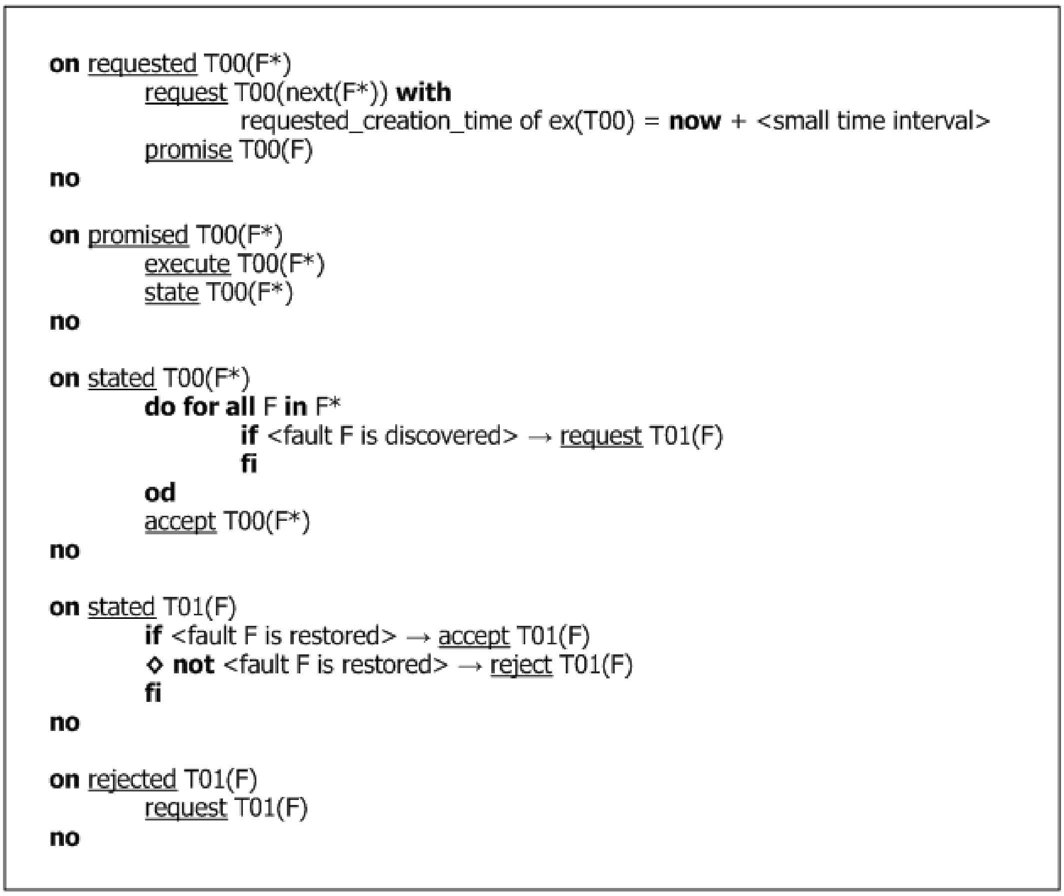
Table 4.2: (B) Information Use Table

object class, fact type or result type	**process steps**
NETWORK ELEMENT	T00/ex, T00/st, T01/ex, T02/ex, T03/ex, T04/ex
operational	T00/st
location	T00/ex
repair_history	T01/ex, T02/ex, T03/ex, T04/ex
FAULT	T00/st, T01/pm, T01/ex, T01/st, T01/ex, T02/ex, T03/ex, T04/ex, T05/ex
downtime	T00/st, T01/st
ticket_number	T01/ex, T02/ex, T03/ex, T04/ex, T05/ex
major_outage	T01/pm
critical	T05/ex
The network has been checked for faults F*	T00/ac
The fault F has been handled	T01/ac
The front office fault FF is handled	T02/ac
The back office fault BF is handled	T03/ac
The next level support fault NF is repaired	T04/ac
The criticalness of fault F has been determined	T05/ac
The fault F is handled by the helpdesk	T06/ac

4.1.3 The Action Model

The AM contains the action rules that serve as guidelines for the actors in dealing with their agenda. It is the second level of detailing of the CM, as it details the steps of the PM. The AM contains all the information of the other models, which are strictly speaking derived from the AM.

The action rules for each actor role are shown in figure 4.4 to 4.7. As mentioned in [12], one should keep in mind that these rules serve only as guidelines for an actor. Sometimes it might be necessary for the actor to deviate from an action rule. The same pseudo algorithmic language for specifying the action rules of an organization is applied, as suggested by Dietz in [12].



```
on requested T00(F*)
        request T00(next(F*)) with
                requested_creation_time of ex(T00) = now + <small time interval>
        promise T00(F)
no

on promised T00(F*)
        execute T00(F*)
        state T00(F*)
no

on stated T00(F*)
        do for all F in F*
                if <fault F is discovered> → request T01(F)
                fi
        od
        accept T00(F*)
no

on stated T01(F)
        if <fault F is restored> → accept T01(F)
        ◇ not <fault F is restored> → reject T01(F)
        fi
no

on rejected T01(F)
        request T01(F)
no
```

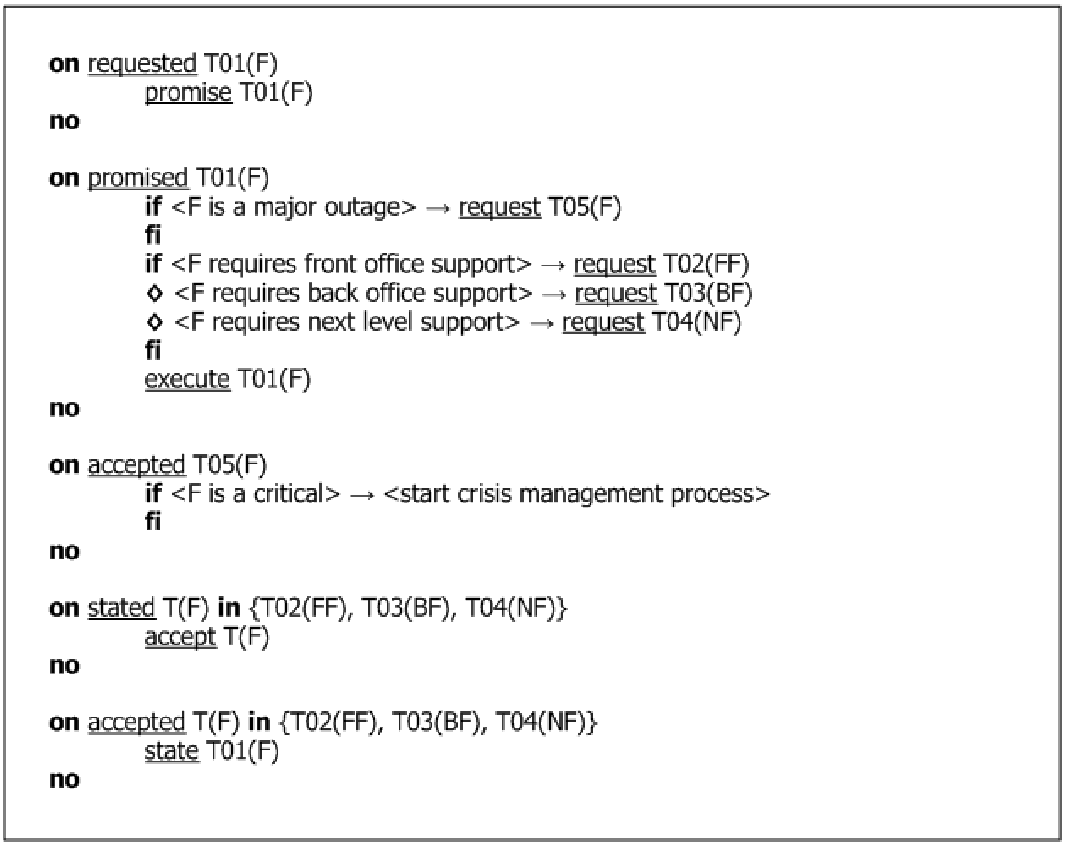
Figure 4.4: (B) Action Rules - A00

Figure 4.4 shows the action rules for actor A00, the monitoring agent. This actor is responsible for monitoring the network for faults. Although this task could be performed by a computer[2], the monitoring agent is responsible for the correct execution of this task. The transaction T00 (check for faults) is requested through self-activation. In action rules, this is realized by starting a new T00 at the time of dealing with the request of the current one. The function *next* is used to provide the next time period. The requested creation time of the P-event is set to now plus an infinitely small time interval. Once T00 has been stated, the monitoring agent has checked the network for the set of possible faults, F*. Now, for each possible fault F in F* that has been discovered, a transaction T01 (handle fault) is requested.

Once a handled fault has been stated (T01-st), the monitoring agent checks if the fault is actually repaired. Repaired faults result in an accept state, whereas

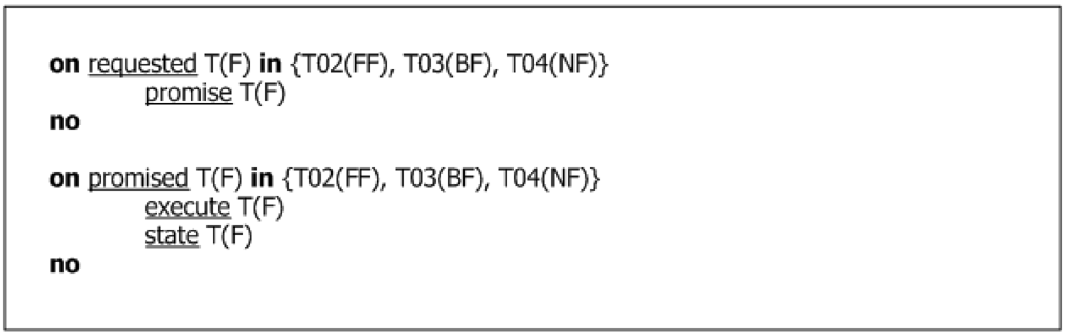
---

[2] in fact, the application Netcool incorporates network monitoring functions



```
on requested T01(F)
        promise T01(F)
no

on promised T01(F)
        if <F is a major outage> → request T05(F)
        fi
        if <F requires front office support> → request T02(FF)
        ◇ <F requires back office support> → request T03(BF)
        ◇ <F requires next level support> → request T04(NF)
        fi
        execute T01(F)
no

on accepted T05(F)
        if <F is a critical> → <start crisis management process>
        fi
no

on stated T(F) in {T02(FF), T03(BF), T04(NF)}
        accept T(F)
no

on accepted T(F) in {T02(FF), T03(BF), T04(NF)}
        state T01(F)
no
```

Figure 4.5: (B) Action Rules - A01



```
on requested T(F) in {T02(FF), T03(BF), T04(NF)}
        promise T(F)
no

on promised T(F) in {T02(FF), T03(BF), T04(NF)}
        execute T(F)
        state T(F)
no
```

Figure 4.6: (B) Action Rules - A02, A03 and A04

not or insufficiently repaired faults will lead to a rejection, after which T01 (handle fault) is requested again for the same fault F.

Discovered faults are dealt with by A01, the fault handler. Figure 4.5 shows the action rules for A01. Each fault which is categorized as a major outage, will be checked for criticalness. If a major outage is indeed critical (there are a number

```
on requested T05(F)
        promise T05(F)
no

on promised T05(F)
        execute T05(F)
        state T05(F)
no
```

Figure 4.7: (B) Action Rules - A05

of criteria used to assess criticalness), a crisis management process is started. This process is not included in the previous aspect models, at it lies beyond the scope of network monitoring. Any fault is sent to the appropriate authority, i.e. one of the transactions T02, T03, T04 will be requested.

Since the action rules for A02, A03 and A04 are similar, figure 4.6 shows to generic action rules that can be applied for all three. The agendum and action in both rules involve transaction T(F), which could be an arbitrary element in the set {T02(F), T03(F), T04(F)}. As also goes for actor A05, who's action rules are shown in figure 4.7, the actions taken for dealing with agenda, directly follow the steps in the basis transaction pattern.

4.1.4 The State Model

The State Model (SM) specifies the state space of the P-world. The SM is expressed using two diagrams: the Object Fact Diagram (OFD) and Object Property List (OPL). Figure 4.8 shows the OFD. The result types from the TRT (figure 4.1 can be found in the OFD along with their connected object classes. Note from the TRT that there is only one central object class: the *fault*. This is not surprising, since the top layer of network monitoring only concerns handling faults. This is because the B-organization is merely an abstraction of the actual implemented processes. On this highest level, the solely existing object class is the fault. However, a fault implicitly can be divided in two sub object classes, namely network elements and errors. A fault is defined as a network element that contains an error. Concluding, the set of faults is a subset of all network elements, i.e. the ones that are erroneous. Because the action rules distinguish between three kinds of faults, depending on the level of support that is concerned with the fault, the specialization of faults is also shown in the OFD. FF stand for front office fault, BF stands for back office fault and NF stands for next level support fault. Having adopted the notion of World Ontology Specification Language (WOSL), which is used as the formal notation of the SM, the object class Fault is a *partition* of all errors.

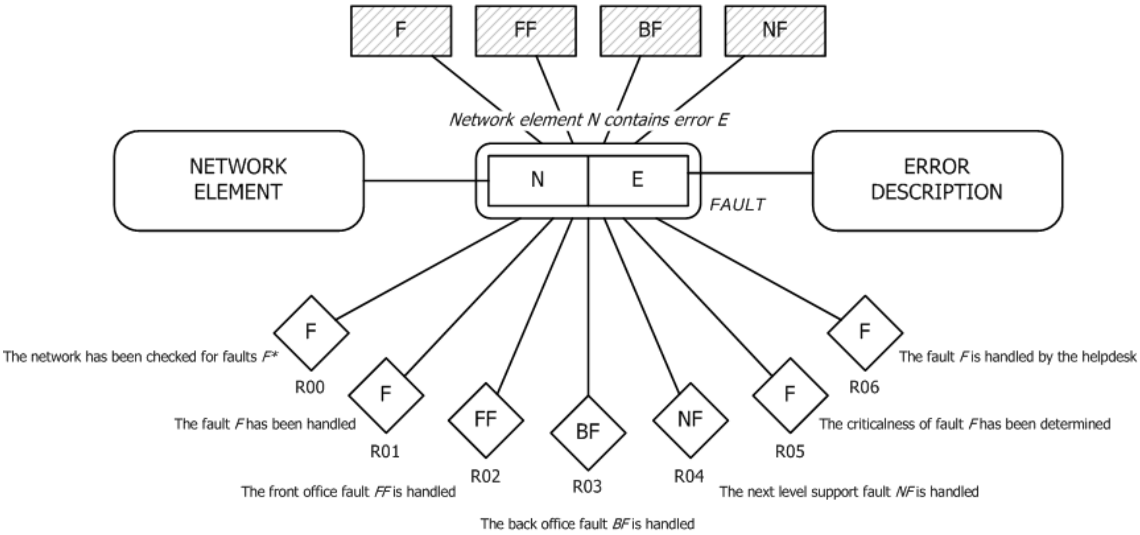


Figure 4.8: (B) Object Fact Diagram

The OPL of the SM contains properties of object classes and their respective scale. The OPL is shown in table 4.3.

Table 4.3: (B) Object Property List

property type	**object class**	**scale**
operational	NETWORK ELEMENT	BOOLEAN
location	NETWORK ELEMENT	DESCRIPTION
repair_history	NETWORK ELEMENT	DESCRIPTION
downtime	FAULT	TIME
ticket_number	FAULT	NUMBER
major_outage	FAULT	BOOLEAN
critical	FAULT	BOOLEAN
severity	NETWORK ERROR	RATIO
customer_effect	NETWORK ERROR	BOOLEAN

4.1.5 The Interstriction Model

Subsection 4.1.1 covered the left-hand side of the CM, as can been seen in figure 4.1. This subsection covers the other part of the CM, which is the Interstriction Model (ISM). The ISM contains the passive influences between actor roles, i.e. taking the existing facts into account (as an actor). It is expressed in the Organization Construction Diagram (OCD) and Bank Contents Table (BCT). The OCD, shown in figure 4.9, almost exactly resembles the ATD, but in addition contains information links and links to production banks (both internal and external).

Production bank CPB08 contains all the error data. These are descriptions of all described errors that could possibly occur within the network. This is static data, as the set of all possible errors is unaffected by the monitoring processes. CPB09 contains everything that needs to be known about the network that is monitored, such as all the sites, its elements and their state. Using information from both CPB08 and CPB09 can be used to monitor the network for faults. Discovered faults are stored in CPB10, which needs to be available with all support levels of the service desk.

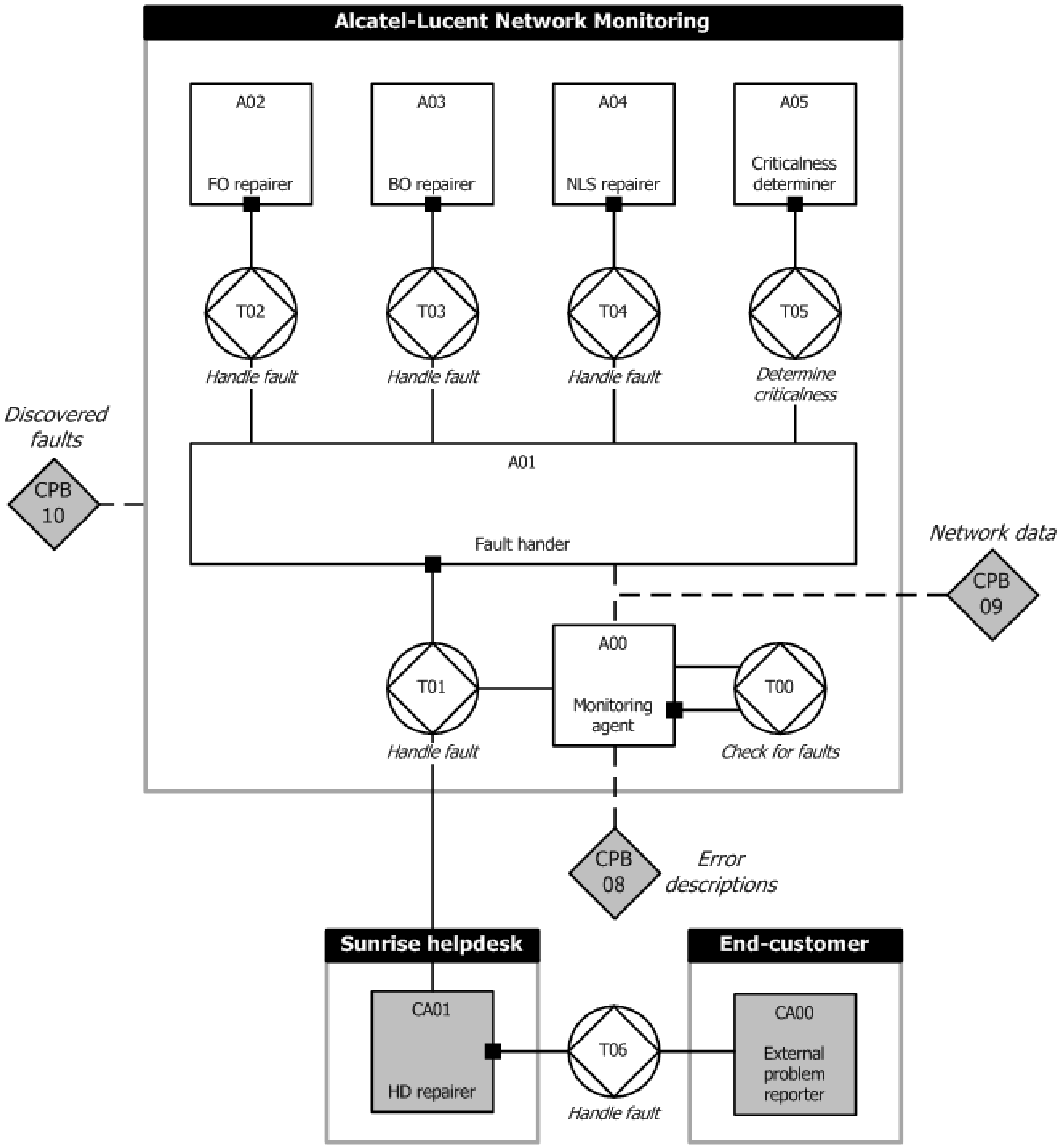


Figure 4.9: (B) Organization Construction Diagram

The BCT, shown in table 4.4, clarifies the information links. The internal production banks and their contents are denoted by PB with their numbering corresponding with their transaction. Next to the internal production banks, a number of external production banks which are used during the execution have been identified as well.

Table 4.4: (B) Bank Contents Table

object class, fact type or result type	**P-bank**
The network has been checked for faults F*	PB01
The fault F has been handled	PB02
The front office fault FF is handled	PB03
The back office fault BF is handled	PB04
The next level support fault NF is repaired	PB05
The criticalness of fault F has been determined	PB06
The fault F is handled by the helpdesk	PB07
NETWORK ERROR	CPB08
severity	
customer_effect	
NETWORK ELEMENT	CPB09
operational	
location	
repair_history	
FAULT	CPB10
downtime	
ticket_number	
major_outage	
critical	

4.2 Modeling the I-organization

The I-organization is partly derived from the B-organization. Therefore, ontological models of the B-organization need to be determined first before the I-organization can be modeled. In the previous section, several diagrams were shown which constitute the design of the ontological model of an enterprise's B-organization. There are four aspect models, of which three models—PM, SM and CM—are derived from one, the AM. The AM is therefore a good starting point for devising the I-organization. By doing so, the resulting I-transactions are considered to be a matter of realization of the B-transactions, because their only purpose is to request and store information required by the B-actors. This is consistent with the view that the function of the I-organization supports the construction of the B-organization. Similarly, the D-transactions are considered to be a matter of realization of I-transactions, as their function is to care for the storage and retrieval, the copying and the transmission of documents containing the rough input data for I-actors.

Based on this information, it can be stated that B-actors initiate I-transactions, which are executed by I-actors, who subsequently can initiate D-transactions that are executed by D-actors. This means that a B-actor uses its performa abilities to execute a B-action and its informa abilities to initialize an I-action. The switching of abilities is achieved when B-actors shape into I-actors before initiating an I-transaction. Once the result of the I-transaction is accepted, the subject shapes back to its original role (in this case the role of a B-actor) and continues with its B-acts. The same holds for the interaction between I-actors and D-actors. A particular subject who possesses of all the three human abilities can therefore fulfill several B-actor roles, I-actor roles and D-actor roles within the same social system [10].

In [11], two basic scenarios regarding the integration aspects between the B-,I- and the D-organizations are identified. These two scenarios are the retrieval and information and the storage of information.

Retrieval of information

Consider the situation that a B-actor requires some information that has to be retrieved from data in the fact bank by means of some computation. In this situation, a B-actor first has to shape into an I-actor and then initiates an I-transaction. For the execution of the I-transaction, the I-actor needs some data from the fact bank. In order to retrieve this data, the I-actor shapes into a D-actor and initiates a D-transaction. Figure 4.10 illustrates this situation. The inclusion of the I-transaction is not always necessary. In that case, the B-actor shapes into an I-actor and directly initiates a D-transaction in order to retrieve some data, by shaping into a D-actor.

Storage of information

Next to retrieving data and information, the B-actor should also be able to store data and information. Figure 4.11 shows the situation where the B-actor first shapes into an I-actor and initiates an I-transaction. The executing I-actor on its turn shapes into a D-actor and stores the original data in a fact bank.

The viewpoint that the *sole* purpose of the I-organization is to support the B-organization is not completely shared throughout this thesis. Although most I-transactions are derived from the AM of the B-organization, there are also I-transactions that do not support B-transactions. It is found that not only B-actions contribute to an organization's goals, but in some cases also I-transactions. The same goes for D-transactions that do not directly support and I- or B-transaction.

The modeling method is as follows. First, based on the AM of the B-organization, the function of the I-organization is determined. The identified functionality is ex-

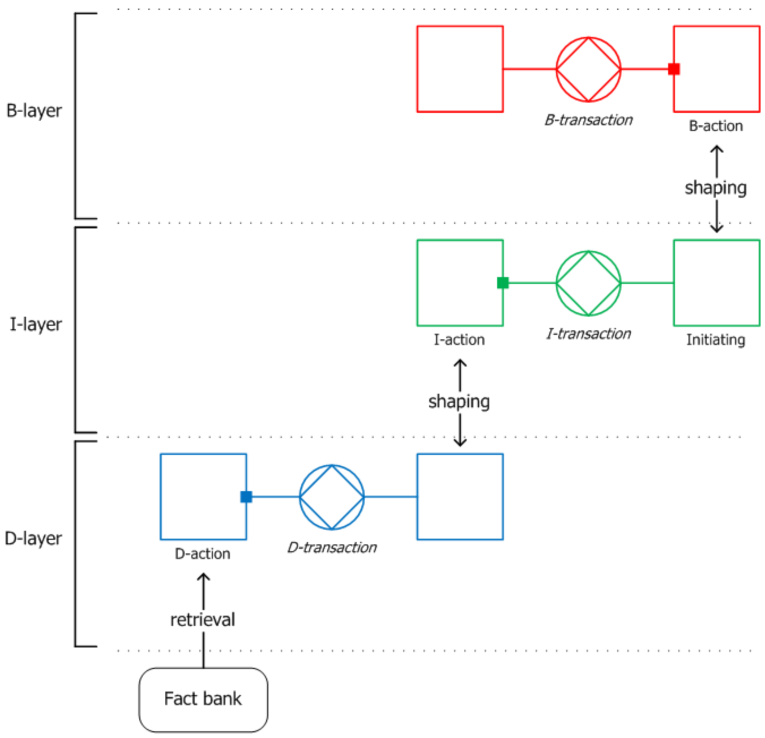


Figure 4.10: Retrieval of information

tended with the results of the PIF-analysis on the textual process descriptions on page 30 to 33. Once this is done, the construction of the I-organization is determined and expressed in the I-ATD. The I-ATD and corresponding TRT are shown in figure 4.12 and table 4.5, respectively.

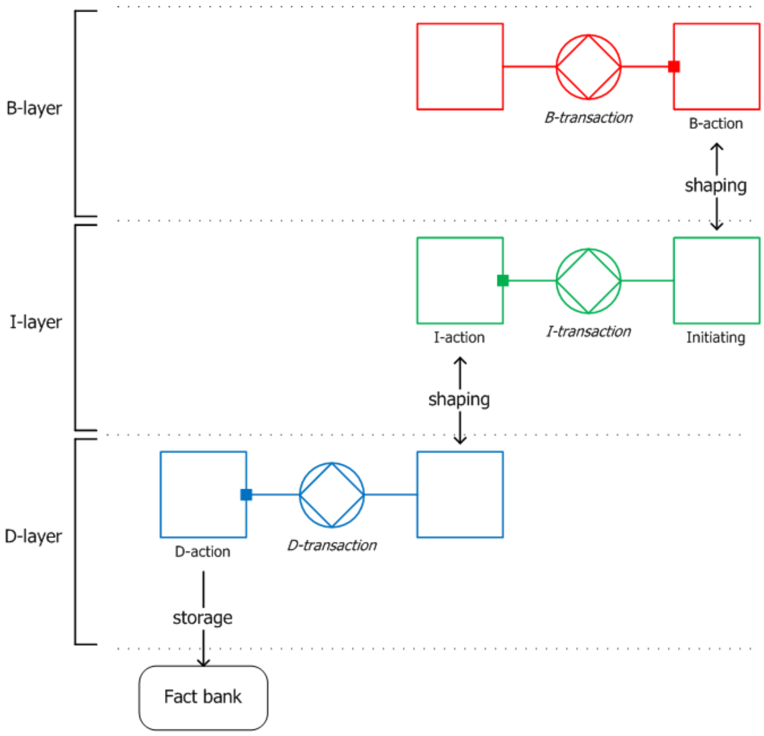


Figure 4.11: Storage of information

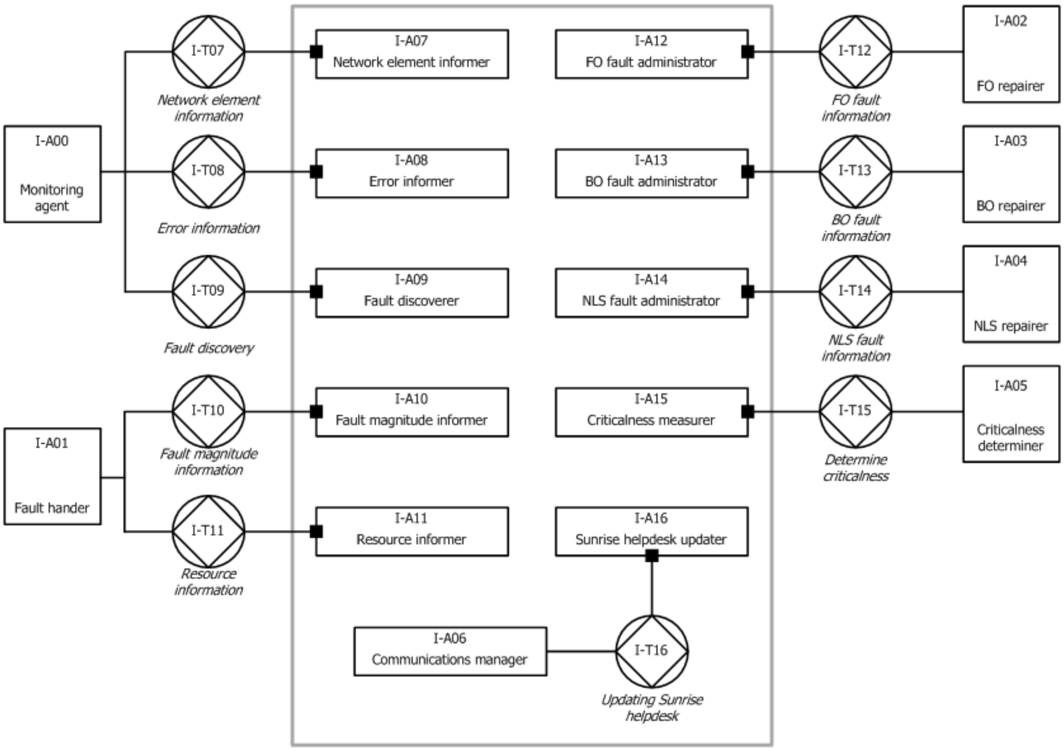


Figure 4.12: (I) Actor Transaction Diagram

Table 4.5: (I) Transaction Result Table

Transaction		**Transaction result**	
IT07	network_element_information	IR07	network element information has been provided
IT08	error_information	IR08	error information has been provided
IT09	fault_discovery	IR09	fault discovery has been performed
IT10	fault_magnitude_information	IR10	fault magnitude information has been provided
IT11	resource_information	IR11	resource information has been provided
IT12	fo_fault_information	IR12	fo fault information has been provided
IT13	bo_fault_information	IR13	bo fault information has been provided
IT14	nls_fault_information	IR14	nls fault information has been provided
IT15	determine_criticalness	IR15	The criticalness has been measured
IT16	updating_sunrise_helpdesk	IR16	The front office has updated the helpdesk

The transaction in figure 4.12 include retrieving data which is required in order to perform the B-transactions in figure 4.2. The actor roles I-A00 to I-A05 correspond to the actor roles of the B-ATD, with the note that these subjects are now shaped into I-actors. In order for the I-actors in the rectangle in figure 4.12 to execute the transactions, they will probably require further D-actions to be taken. This is shown in the next section.

For the I-organization, only the ATD and TRT are given. This is because of three reasons. Firstly, the high degree of abstraction on the actual implemented processes at the service results in aspect models that are too far deviated from reality. Drawing conclusions based on such a high abstraction level is meaningless. Secondly, in order to identify business components in a later stage, the IAM includes all required input for the BCI method. Thirdly, the sequence of process steps is exactly the same as in the basic transaction pattern in figure 2.5.

4.3 Modeling the D-organization

The models for the B- and I-organization have been completed. The combination of both will result in ontological models of the D-organization. Analogous the connection between the B- and I-organization, the data requirements for I-transactions are sufficient to model the D-organization. However, it is also possible for an actor in the B-organization to directly interact with the D-organization without any intermediate I-transactions. D-transactions for these kinds of interactions cannot be identified from the I-organization. Therefore, the B-organization is included during the identification as well.

Recall the organization theorem (section 2.2.1 on page 20). It introduces the three heterogeneous tiers that together constitute an organization. These three tiers are the B-, I- and D-organization, which corresponds with the sequence in which the network monitoring processes are modeled in this chapter. The triangular shape of figure 2.11 has some interesting characteristics. It was stated that the surfaces of the respective organization parts are scaled 1:4:7, which is an approximation of the relative proportions of the set of transactions in each organization. These figures imply that the number of D-transactions is seven times as large as the number of B-transactions. For obvious reasons, the most straightforward D-transactions, those that directly support B- and I-transactions, are not shown in the D-ATD (figure 4.13).

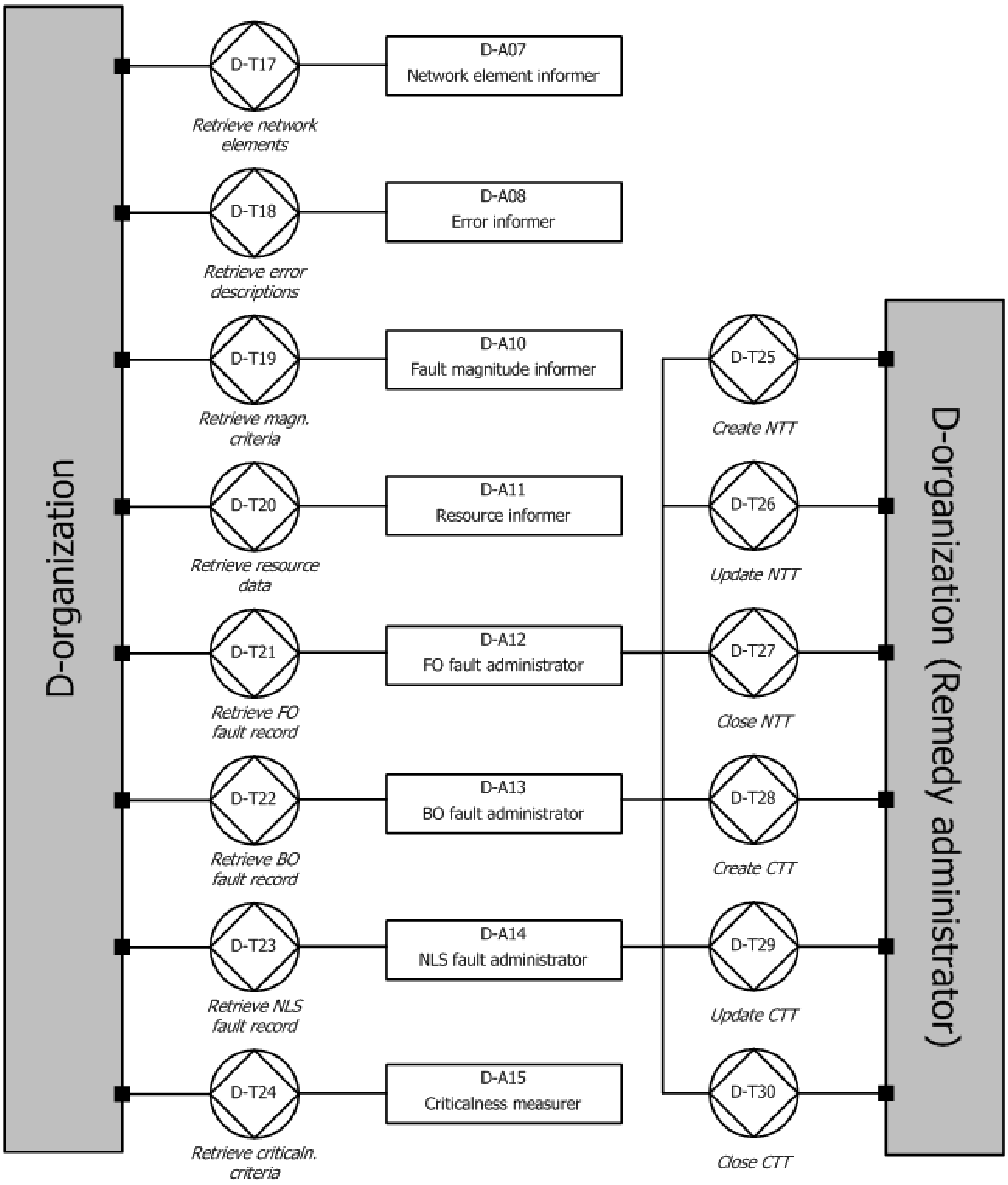


Figure 4.13: (D) Actor Transaction Diagram

Table 4.6: (D) Transaction Result Table

Transaction		**Transaction result**	
DT17	retrieve_network_elements	DR17	Network elements have been retrieved
DT18	retrieve_error_descriptions	DR18	Error descriptions have been retrieved
DT19	retrieve_magnitude_criteria	DR19	Magniture criteria have been retrieved
DT20	retrieve_resource_data	DR20	Resource data have been retrieved
DT21	retrieve_FO_fault_data	DR21	FO fault data have been retrieved
DT22	retrieve_BO_fault_data	DR22	BO fault data have been retrieved
DT23	retrieve_NLS_fault_data	DR23	NLS fault data have been retrieved
DT24	retrieve_criticalness_criteria	DR24	Criticalness criteria have been retrieved
DT25	ctt_create	DR25	Customer trouble ticket CTT has been created
DT26	ctt_update	DR26	Customer trouble ticket CTT has been updated
DT27	ctt_close	DR27	Customer trouble ticket CTT has been closed
DT28	ntt_create	DR28	Customer trouble ticket NTT has been created
DT29	ntt_update	DR29	Customer trouble ticket NTT has been updated
DT30	ntt_close	DR30	Customer trouble ticket NTT has been closed

Chapter 5

Business components

The previous chapter has shown the ontological models of the B-, I- and D-organization. As stated earlier, these models are used as input for the identification of business components and ICT services. This is relevant, as it will present the scientific alternative to two commonly used solution frameworks in telecommunications: the TM Forum's process framework and applications framework. The differences between both approaches will be evaluated in the next chapter.

First, the relationship between business components and services is described in section 5.1. After this, the results of the business component identification (BCI) method are given in section 5.2. This chapter concludes with some final remarks in section 5.3. This chapter performs research steps (6).

5.1 The relationship between components and services

The methodology used to identify services, is based on the generic system development process (GSDP), shown in figure 2.14 on page 23. The first step was to produce ontological models of the B-, I- and D-organization, from which the results have already been presented. Within the GSDP, this is defined as the *construction* of the object system. Previous research, as presented in [3], introduce a framework that can be used in order to identify functional specification for services. This framework, called the *generic service specification framework* (GSSF), identify services based on the transactions from the ontological models. Each transaction maps to one service, which can be further classified as one out of six service types.

The next step is to design the construction of the services. This is done using the BCI method. To this end, the concept of business components and the BCI method are used. A basic description of the BCI method was provided in section 2.3. Having defined as service as (part of) a transaction, it can be stated that business components are clusters of coherent functionality, in which the latter is delivered through one or more services.

5.2 Applying the BCI method

This section applies the BCI method to build a construction model of the services. As mentioned before, the constructional model will consist of a set of business components, that contain process steps within transactions, the services. Furthermore, a business component includes links between other components, from which it is made clear which services they consume and which information objects they create or use. The business component model is created using the BCI method, an optimization algorithm and can be accessed using the BCI-3D tool. This is a Java application that uses the Java3D package in order to generate a graphical representation of business components in a three-dimensional space.

The optimization algorithm of the BCI method has promptly been discussed in section 2.3. It includes three basic steps, presented in the following three subsections.

5.2.1 Denote partitioning preferences

In this step, the relationships between the services are formalized. This is done by looking at the process steps that are implemented by the services and the dependencies between them. The relationship between the process steps and the information objects as defined in the SM of the B-organization are also considered. The result of this step is an initial graph which visualizes the various relationships.

The input for the BCI-3D tool consists of a set of tables which represent the relationships between the various services and the information objects that are identified within the business domain. Three basic kinds of relationships are defined using the following tables:

1. Relationships between information objects (io-io) — Relationships between the information objects as defined in the SM of the B-organization are represented in this table. Information objects are either related-to, part-of or a state-of each other.

2. Relationships between functions (fun-fun) — This table represents the relationships between the various services. These relationships have already been examined by looking at the service parts for composite services. The relationships are classified as standard, optional, conditional or multiple.

3. Relationships between information objects and functions (io-fun) — There are three types of relationships between information objects and process steps, namely create, alter and use. This information is extracted from the IUT and the AM.

The actual input used is shown in appendix C. Using that input has resulted in the initial graphs for the B-, I- and D-organization, shown in figure 5.1, 5.2 and 5.3, respectively.

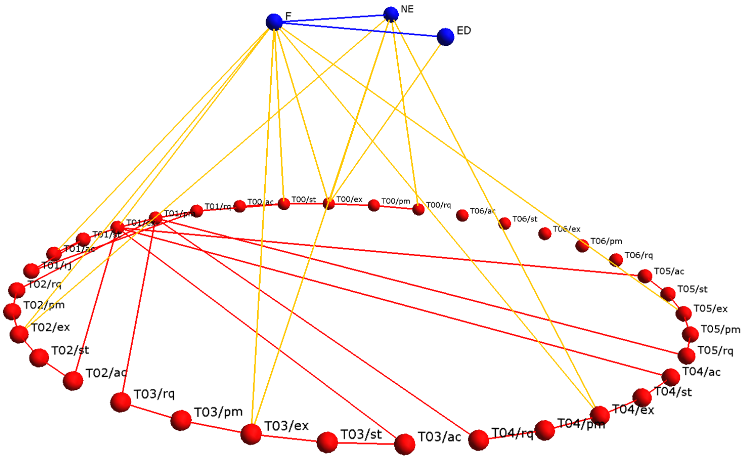


Figure 5.1: (B) Initial graph of the BCI-3D tool

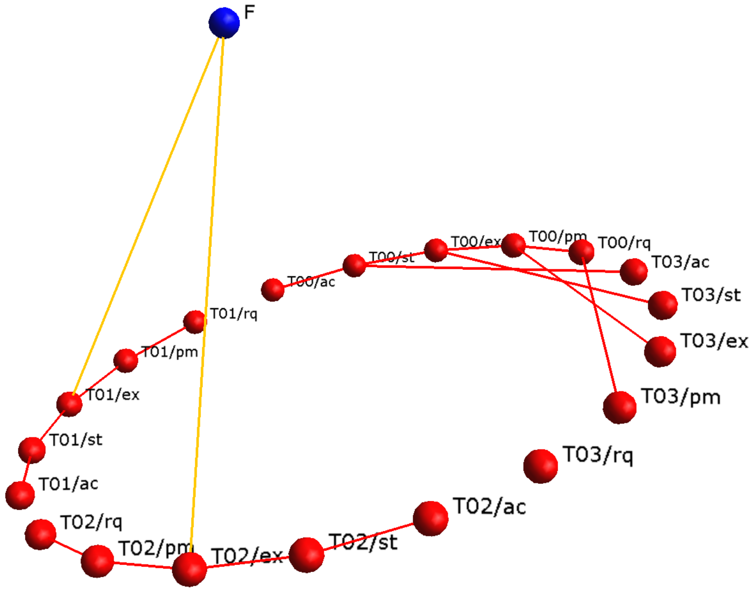


Figure 5.2: (I) Initial graph of the BCI-3D tool

5.2.2 Create optimized partitioning

In this step, the weights to the various relationships (edges in the graph) are assigned, with the goal of grouping process steps and information objects into individual business components. The assignment of weights is based on SOA design principles as described in [16]. The weights are shown in figure 5.4.

The weights are based on the strength of the relationship between two nodes, in which weights of standard and optional relations are in line with their cardinality.

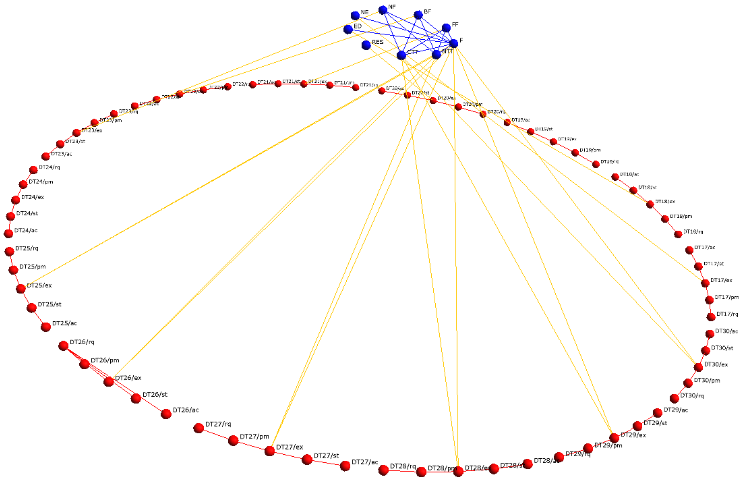
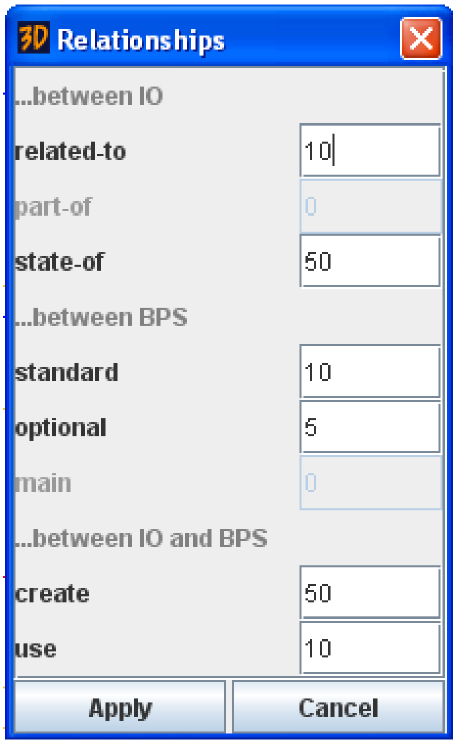


Figure 5.3: (D) Initial graph of the BCI-3D tool



...between IO
related-to: 10
part-of: 0
state-of: 50

...between BPS
standard: 10
optional: 5
main: 0

...between IO and BPS
create: 50
use: 10

Figure 5.4: Assigned weights

The relation between a service which creates (and thus *owns*) a certain object is much stronger than between a service which just uses a certain object. Between information objects, a state-of relationship is much stronger than a related-to relationship, as state-of relationships are between the same type of information objects and therefore use the same interface towards process steps and other information objects. Furthermore, there is a difference between the standard and optional relationships between process steps. This is because standard relationships on average (at least, in this thesis) are exploited twice as much as optional relationships.

5.2.3 Create component specification

During this step, the functionality of the resulting business components from the previous step was specified in terms of the services they implement, the services they consume and the information objects they create or use. An overview of the process steps that are included in each business component is provided in appendix C. Figure 5.5, 5.6 and 5.7 show a graphical representation of the results of performing the Kernighan and Lin graph partitioning algorithm [2] for improving the initial solution. The isolated 'islands' that are visible in the figures are business components. From these figures, it can vaguely been seen which process steps are clustered together. However, because only short names for process steps are visible, the following bullet list describes the generated business components.

- Business components of the B-organization (figure 5.5):

    1. There is only one business component in the B-organization. The explanation for this is that only the network monitoring processes with ALU were chosen in this thesis. Having all the process monitoring facilities in one location, which is the ALU headquarters, it makes sense that all the provided services of the service desk are optimally clustered together in one component. Since there is only one B-component, it includes fault monitoring, fault handling by various support levels and determining whether faults are critical.

- Business components of the I-organization (figure 5.6):

    1. The largest business component contains the core of information retrieval services that are used in order to perform most B-transactions. This includes fault discovery in network elements, fault administration on all support levels and criticalness determination.
    2. Checking whether network elements contain behavior corresponding to that of an error description is a very specific process, because the information required is only used for this purpose. It is therefore not surprising that this resulted in a separate business component for this function.
    3. Checking whether resources are available for the handling of a fault is a process which is not specific for network monitoring and therefore is isolated within a separate business component, as it has no connection with other information objects or process steps.

4. The same goes for informing the helpdesk of Sunrise. This is done in case a fault has a direct (negative) influence on the customer. Like the previous component, there are no connections to other information objects or process steps.

- Business components of the D-organization (figure 5.7):
    1. Fault data retrieval on all support levels, along with NTT and CTT ticket creation (which is done when a fault is assigned to support staff).
    2. Customer trouble ticket management.
    3. Network trouble ticket management.
    4. Network element retrieval.
    5. Error description retrieval.
    6. Resource retrieval, i.e. retrieving data in order to decide whether resources (including staff) are available for solving faults.
    7. Criticalness criteria retrieval.
    8. Magnitude criteria retrieval.

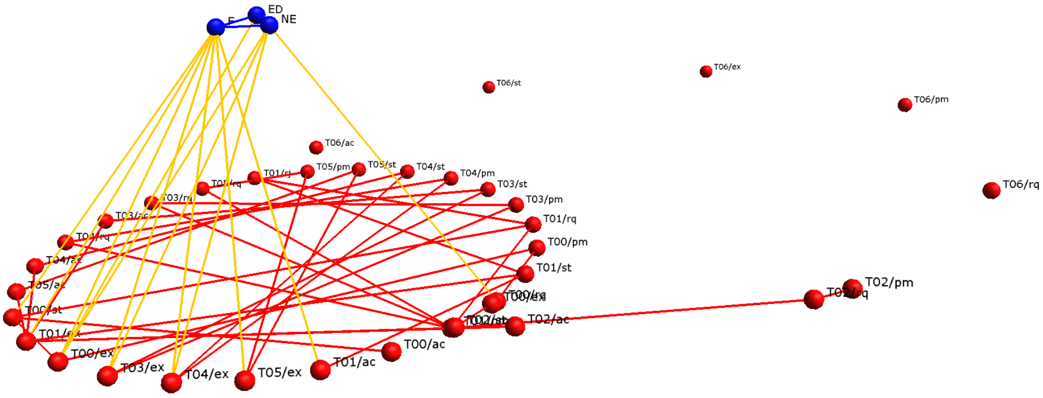


Figure 5.5: (B) Optimized graph of the BCI-3D tool

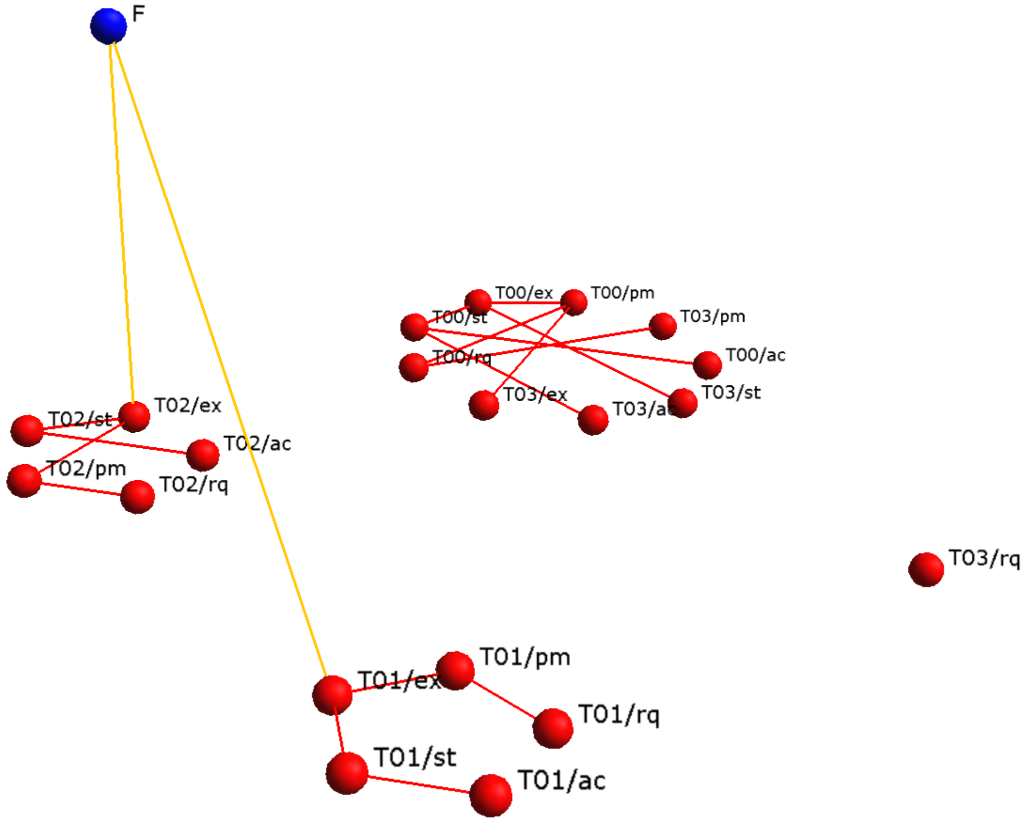


Figure 5.6: (I) Optimized graph of the BCI-3D tool

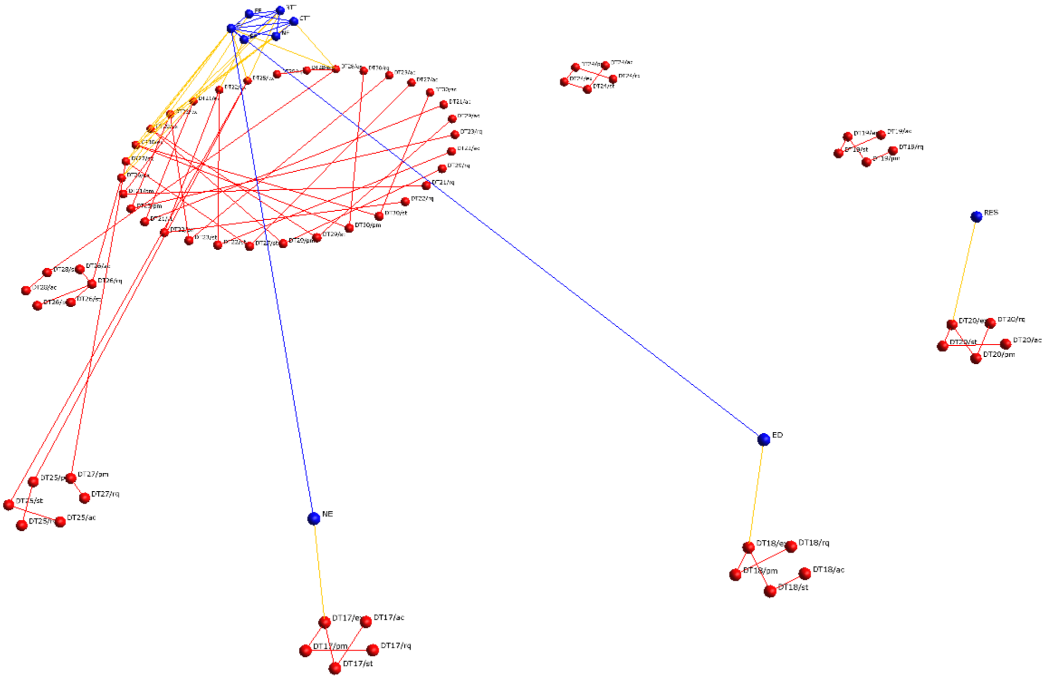


Figure 5.7: (D) Optimized graph of the BCI-3D tool

5.3 Concluding remarks

Running the BCI method has resulted in a number of identified business components for the three organizations in the organization theorem. The results show some interesting aspects:

- The relationship between B-, I- and D-components is the same as the relationship between the B-, I- and D-transactions within the ontological models. The way in which components differentiate themselves from transactions, is the fact that they are clustered in an optimal way.
- The number of business components for the B-, I- and D-organization are 1, 4 and 8, respectively. If compared with the scale 1:4:7 in the organization theorem, it can be seen that these numbers almost match. This is not surprising, as it was already expected that the number of services on lower levels is larger than the number of services on higher levels.
- Altogether, it can be concluded that applying the BCI method using the BCI-3D tool is a straightforward task, in which the results can be foreseen, as they are almost completely dependent on modeling the three organizations.

Chapter 6

Evaluation

This chapter evaluates two approaches for supporting ALU's operate-processes. The first approach will be discussed in section 6.1 and is based on the four solution frameworks, designed by the TeleManagement Forum. The second approach is based on enterprise ontology and the business component identification method. This chapter performs research step (7).

6.1 Industry approach

The industry approach uses the four TM Forum solution frameworks. This thesis and the preceding literature survey identified some strengths and weaknesses. This section reflects on these strengths and weaknesses, first from an industry perspective in general and secondly from the current situation at Alcatel-Lucent in particular.

The frameworks are designed by the TM Forum, which is an industry organization that helps telecom service providers in performing all aspects of their business. Representatives of multiple member companies of the TM Forum help identifying a generic set of principles and guidelines, grouped in four frameworks such that it covers all dimensions of a TSP's organization. The first framework, which is by far the most well known and used within the industry, is the process framework. The separation of the three major process areas, the multiple levels of detail and the focus on functional end-to-end processes (e.g. fulfillment, assurance) as well as each level of the value chain (from customer to supplier), enables different kinds of people to benefit from this framework. It has proven to be a well-thought basis for any (telecom) service provider to identify their business processes, without forgetting any relevant detail. One downside of having such a one-size-fits-all framework, is that it is still rather generic, even on the lowest possible level of detail. Although there are good reasons for this, it still requires a great deal of effort in actually defining the flow of people, assets and processes. Second, the information framework, categorizes a telecom service provider's business entities among the same functional domains as in the process framework. A strength is that it takes everything into account. The applications framework is very extensive, in way that it virtually covers all the functionality that should be fulfilled using applications. However, the focus is merely on the coverage of the complete business needs, but does not contain sufficient information required to procure and implement applications in a cost-effective way.

In general, the TM Forum solution frameworks are very complete and very well thought. The disadvantage of this is that one requires to spend much time to figure out the relationships between those solution frameworks. There are many documents and addenda that describe the frameworks. Most of those documents are exceedingly bulky, and because there are several authors working on these documents, the terminology is often used inconsistently. It would benefit the TM Forum to make an effort in clarifying the relationships between the documents, making sure that terminology is used consistently and in making the frameworks more practical, e.g. by having one single process that anyone could use in order to apply the frameworks for their business needs.

As shown in section 3.3, ALU uses a subset of the services that are listed in the TM Forum applications framework. Although the ICT systems descriptions that were made available for this thesis are quite limited, employees of the service desk declared that there were two drivers for ICT system selection and procurement:

- Scope of offered functionality by a ICT systems
- Procurement costs

The first drivers, which is the extent of which ICT systems cover the ICT functionality required in the service desk, is determined by the years of experience of two program managers at ALU. There is no internal evaluation whether the selected ICT systems are actually consistent with SOA principles as discussed in [16], as the current opinion of the service desk is negative towards changing the current ICT systems landscape. Once a list of required ICT systems is identified, the second driver—procurement costs—determines which one of many comparable products will be used. Unfortunately, due to limited time and scope of this project and because of lack of available information, a benchmark of ALU's ICT effectiveness and efficiency towards their competitors could not be made. This would indicate whether ALU is already on the right track or that they should improve on this point.

The case study in this thesis involved network monitoring processes at the ALU service desk. These are processes that run to make sure that faults in the network are dealt with in the appropriate way. With appropriate is meant that the ALU should operate the network in compliance with the service level agreement between ALU and the customer. There are three categories of ICT systems used that support the network monitoring processes. These are element management systems (EMSs), systems that collect and filter alarm information and a system for tracking trouble tickets.

In theory, this vertical separation (network, applications, people) would work well. However, ALU uses multiple applications in parallel on both the EMS and alarm tool level. The reason for this is that every vendor uses its own EMS, some of which are able to forward messages to one alarming tool, and other EMSs to the other alarming tool.

Another weakness is the lack of workflow management within the service desk. All communication by staff members is done over telephone, which is a time consuming practice.

6.2 Scientific approach

The scientific approach comprises the use of enterprise ontology and business components. This approach is presented as alternative for the current approach taken by ALU, which was described in the previous section.

The modeling methodology of combination between enterprise ontological and the identification of business components uses the generic system development process, shown in figure 2.14 on page 23. From a theoretical point of view, this process is suitable for all systems, including social systems such as ALU. Enterprise ontology has a very deep theoretical background.

Enterprise ontology is a field that has proven itself in practice. There are numerous case studies in which enterprise ontology is successfully used. Looking at these case studies, two common characteristics are observed:

- It does require a deep understanding of the DEMO methodology and preferably also of the underlying theory of enterprise ontology, in order to successfully build ontological models of organizational activities. An insufficiently skilled modeler will build models that deviate from semantical and syntactical correctness. The founding father of enterprise ontology and DEMO, Jan Dietz, has stated that modelers with the same skill set will produce identical models (assuming that they both adopted the full extent of the concept of enterprise ontology). This implies that a modeler follows a predefined set of steps and does not require any decision making abilities. One could even say that a computer program could replace a modeler, if a modeler's judgment could be programmed. It has been found in this thesis that this is only true in theory. In practice, very subtle differences can lead to major changes. A slightly different judgment of the modeler during the PIF-analysis or transaction pattern analysis, will lead to an other set of transactions and eventually different business components. Only modelers with an extensive experience will produce identical models. However, in general, not chance, but different judgment will result in different models.

- Another important factor is the amount of insight into organization activities. This means that the available information should cover a wide range of organizational activities, and that the available information should be sufficiently detailed.

The BCI method is very powerful. It solves an unstructured problem, which is the identification of ICT functionality to support a social system, in a completely structured manner. The Kernighan and Lin graph partitioning algorithm, in combination with the possibilities to assign weights, allow the identified services to comply with SOA principles as described in [16].

The business components generated in this thesis present a logical distribution of services. For the B-organization, there will only be one business component. This covers the entire width of the processes at the service desk. It fills the current gap, which is a layer on top of the ICT systems which serves as a foundation for workflow management.

The business components that are derived for the I-organization, show a similar distribution of functionality as the currect systems at ALU. The concept of 'fault' is however more centralized in the resulting business components than in the current ICT systems. The latter lacks a central place for fault management, as it is sometimes unclear which application provides which details regarding the fault.

In contrast to the current situation at ALU, the set of generated business components provides an optimal blueprint for an application landscape. Because of this optimal functionality distribution, there would be no similar applications running in parallel. Furthermore, it would be better prepared for changing business needs, as each component covers a self-contained piece of functionality.

It should be noted however that the result of the BCI method in a fully mature environment such as ALU will not change the world overnight, because it does not directly map to existing ICT systems on the market. However, in a situation where ICT functionality has to be developed from scratch, it will be the best first step to take.

Chapter 7

Conclusions

This chapter will reflect on the results and draw the conclusions. After this, an overview of the project's recommendations for future work will be given.

7.1 Conclusions

As stated in chapter 1, this thesis evolves around the following research question:

What are the advantages and disadvantages of combining enterprise ontology and the business component identification method in identifying business components and ICT services for supporting ALU's operate-processes, compared to the current approach taken by Alcatel-Lucent, which is based on the TM Forum solution frameworks?

In order to answer the research question above, six research steps were identified, which were described throughout chapters 2 to 6. The first research step was to obtain a clear understanding of the TM Forum solution frameworks. This step is performed by reading scientific papers, looking at industry best practices described in white papers and gaining information from industry experts through interviews and online seminars. It can be concluded that the TM Forum frameworks not only are of great help for TSPs in identifying all the relevant aspects of their business, but that they are absolutely vital for TSPs in order to position itself in the market. The structure of the process framework—which is by far the most prominent framework used—is very spot on as it comes to all the processes that a TSP should carry out in order to be a key player. Having a generic framework for the industry as a whole prevents companies inventing the wheel twice. As it comes to the other framework at which this thesis focuses, the applications framework, there are however many things that can be improved. The applications framework is supposed to be used by TSPs such that they can achieve an effective (i.e. *doing the right thing*) and efficient (i.e. *doing the thing right)* ICT systems landscape. However, the way in which the applications framework is built does not suffice for this goal. The applications framework is merely a list of ICT functionality categorized by topic, but does not take a transparent set of design principles into account. It would benefit the framework to use the set of SOA principles [16] to achieve the aforementioned effectiveness and efficiency. Then again, the applications framework should remain generic enough to be used by multiple service providers. Therefore it should contain an stepwise approach to identify services based on the SOA principles, rather than just listing every possible piece of functionality that could come in hand.

The second research step investigated the concept of enterprise ontology and the identification of business components. There are three theories backing enterprise ontology and there are numerous described cases of enterprise ontology applied in practice. The subject of enterprise ontology was chosen in a preliminary phase of this thesis project, as both the solid theory and some of its publicized applications form a strong motivation for this choice. Furthermore, enterprise ontology is originated at the Information Systems Design group, the group in which this thesis is performed. The outcome of the second research step contains an overview of enterprise ontology and marks some potential benefits for ALU in an early stage, i.e. before using its practical methodology for the construction of the models. The business component identification method is taught to student alongside the subject of enterprise ontology, because the result of applying enterprise ontology enables the identification of business components. The result of enterprise ontology are models of organizational activities that comply with the appropriate set of quality requirements, such that these are the ideal starting point for identifying ICT functionality for business processes. The latter task should be done in an optimal way. The BCI method is considered to give this desired result, as its underlying optimization algorithm is considered to identify groups of coherent ICT services in the most optimal way. The implicit hypothesis that underlies the research question, is that enterprise ontology and the identification of business components improve the current industry approach to some extent.

The third and fourth research step looked into the actual implementation of the TM Forum solutions frameworks based approach at ALU. After having obtained the results of the research done on the TM Forum solution frameworks, the actual processes and supporting ICT functionality at the ALU service desk were identified. The greatest challenge was to obtain enough relevant and sufficient information that could be used to draw viable conclusions. The reasons that this turned out to be a tedious process, were most of all the lack (or rather absence) of documented processes and listings of used ICT applications and services. The blanks had to be filled in by obtaining interviews with service desk employees, which eventually resulted in a number of described processes, from which process charts were drawn (see chapter 3). Furthermore, a concise list used of ICT systems was obtained in the process.

Not surprisingly, the processes that eventually were documented for this thesis, could be mapped on parts of the process framework. Logically, as this thesis focused on the maintenance of telecom networks, the process blocks from the process framework on which a mapping could be made, were included in the ‘assurance’ process vertical. From here on, the mapping of the existing ICT services and the application framework was made. As has been told in a later stage, the reason for which the processes were not documented at all, was because the 'process' owners relied on their own experience in converting high-level process framework blocks into real processes. It was difficult to estimate the extent of efficiency of the current applications, because relevant data (such as revenue vs. ICT costs) was not available. It was also not possible to benchmark with ALU's competitors.

This thesis focused on the network monitoring processes at the ALU service desk. These are processes that run to make sure that faults in the network are dealt with in the appropriate way, i.e. compliant with the agreed SLA. Researching the processes and their supporting ICT systems has shown that the vertical arrangement of functionality (i.e. hardware, applications, people) at the service desk is well structured, but that in practice too many applications run in parallel, although their functionality is often quite similar. There is also no automation within the processes. More specific, a workflow management tool is lacking, which make staff members communicate with each other by telephone almost once for every process step change. This is very time consuming and also inefficient, because it requires a load of employees for handling those calls.

The fifth research step covered the application of scientific methodologies, as an alternative to the currently used industry approach. The information required in order to built the models was already gathered during the third research step, which resulted in a series of flowcharts for distinct processes. First, the aspect models of the B-organization were created. As has been concluded in previous research, the Action Model of the B-organization is a good starting point for devising the I-organization. The I-organization was modeled in a similar approach used for the B-organization. A similar approach was applied to model the D-organization. The validation of the ontological models was done in a meeting with the contact person of the service desk. During the validation session, a lot of time was spent on explaining enterprise ontology concepts rather than focusing on the correctness of the models. This was also the case during meetings with ALU supervisors. Although it was fairly easy to explain the concept in general, it turned out be challenging to explain the way in which the theory could be translated into practice and what results are derived by doing this.

The sixth research step was the identification of business components and ICT services, using the BCI method. This step first analyzed the relationships between the various process steps and information objects they use or create. It can be concluded that nested transactions are present both within one and across multiple aspect organizations. The structure of transactions was used to fill the input tables for the BCI method, which resulted in components that implemented and provided the ICT services in an optimal way. This is done by using various graph partitioning algorithms which produce business components based on the weights assigned to the relationships between the process steps op transactions. The assignment of the weights was done in such a way that a clear distinction remained between the B- and I-organization on one side, and the D-organization on the other. Building on the research done by [4], a mapping from transactions within ontological models to services can be made. From this, it can be determined which services are provided by a single business component.

The seventh and last research steps was to compare both approaches in this thesis. In led to the outcome that the set of generated business components provides an optimal blueprint for the application landscape at ALU. This blueprint omits very similar applications running in parellel, but more important, the top layer business component (the only business component in the B-organization) provides the functionality which is currently missing at ALU: a workflow management tool. It could directly be used as a functional specification for a such a tool, which will bring more automation within the processes of the service desk. ALU will be able to save time and money by changing their current way of working, which includes using telephone calls as much as possible. It can also be concluded that the I- and D-business components bring little improvement to the current ICT system landscape. Using them to change the currently used ICT systems will be a costly operation.

ALU works with pragmatism. Changing the as-is ICT system landscape in this stage will too big of an investment in both time and money, but workflow management is definitely worthwhile to use within the service desk.

7.2 Future work

This section describes some findings of this thesis project that could lead to future research projects.

- Modeling the I- and D-organization using the B-organization as input, is task which is rather sensitive for errors. Since it has been made clear in which ways organizational layers support other (higher) organizational layers, much of this work can be automated. A tool which provides standard building blocks for supporting organizations would decrease the level of sensitivity in the modeling process.
- Since the BCI-3D tool uses the results of the three ontological organizations as input, it would not be completely necessary to manually input the process steps, information objects and the relationships between them. This could be automated as well. In fact, a better integration between ontological models and input for the BCI-3D would not only save the modeler lots of time, but would further decrease the sensitivity for minor changes in a modeler's judgment.
- For ALU in general, it would be beneficial to gain a better overview of their processes, ICT services, quality and costs. Only with sufficiently detailed

information would such a thesis project really show potential benefits in their status quo ICT system landscape. Quality and costs are required to draw fact-based conclusions and possibly enable benchmarks with competitors.

- The telecommunications industry in general, embodied by the TeleManagement Forum, would benefit from more practical-to-use solution frameworks. This means that terminology within documents and addenda should be consistent, documents should be more easily accessible and less bulky. Lastly, a set of SOA principles should be incorporated within all four TM Forum solution frameworks.

Bibliography

[1] A. Albani and J.L.G. Dietz. Benefits of Enterprise Ontology for the development of ICT-based Value Networks. *Software and Data Technologies, Second International Conference, ICSOFT/ENASE 2007, Barcelona, Spain*, July 2007.

[2] A. Albani, S. Overhage, and D. Birkmeier. Towards a Systematic Method for Identifying Business Components. 2008.

[3] A. Albani and L. Terlouw. An Enterprise Ontology-Based Approach to Service Specification. 2009.

[4] A. Albani, L. Terlouw, G.C. Hardjosumarto, and J.L.G. Dietz. Enterprise Ontology Based Service Definition. 2009.

[5] Alcatel-Lucent Switzerland. Vision 2012. PowerPoint presentation, November 2008.

[6] B.J.G. Beelen and J. Drenthen. Identifying Business Components. PowerPoint presentation, October 2007.

[7] E. Brotzmann, M. Andersson, C. Kampfer, and P. Nachtigall. Stella Process & Structures. PowerPoint presentation, November 2008.

[8] H. Buckow and S Rey. Why business needs should shape IT architecture. *McKinsey on Business Technology*, number 19, spring 2010.

[9] CIAO! Research Network. CIAO! mission. `http://www.ciao.tudelft.nl/`.

[10] J. de Jong. Design the ontology of the I-organization of the enterprise.

[11] J. de Jong. Integration aspects between the B/I/D organizations of the enterprise. *CAiSE*, 2009.

[12] J.L.G. Dietz. *Enterprise Ontology: Theory and Methodology.* Springer-Verlag Berlin Heidelberg, 2006.

[13] J.L.G. Dietz. *Is it $\phi\tau\psi$ or bullshit?* 2009.

[14] K. Dilbeck and J. Reilly. TM Forum Introduction and Update. Webinar, November 2008.

[15] M.P.F. dos Santos, W.A. Clarke, and A.L. Nel. The Use of Service Level Agreements in Operational Risk Management to Enhance Telecommunications Business Operations. *Engineering Management Conference, 2006 IEEE International*, September 2006.

[16] T. Erl. *Principles of Service Design.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2007.

[17] J. Habermass. *The Theory of Communicative Action.* Beacon Press, March 1985.

[18] G.C. Hardjosumarto. An Enterprise Ontology based Approach to Service Specification. Master's thesis, Delft University of Technology, 2008.

[19] International Telecommunication Union. ITU-T Recommendation M. 3100. `http://www.itu.int/rec/T-REC-M.3100`, April 2005.

[20] M.B. Kelly. The TeleManagement Forum's Enhanced Telecom Operations Map (eTOM). *Journal of Network and Systems Management*, 11, 2003.

[21] H. Tada, W. Usui, and X.J. Wen. An approach toward implementation of OSS/BSS using NGOSS. *ICCT2003*, 2003.

[22] TeleManagement Forum. Introduction to TM Forum. `http://www.tmforum.org/IntroductiontoTMForum/5749/home.html`.

[23] TeleManagement Forum. The NGOSS approach to Business Solutions, November 2005.

[24] TeleManagement Forum. Business Process Framework (eTOM): Concepts and Principles, November 2008.

[25] TeleManagement Forum. TM Forum Applications Framework, June 2008.

[26] D. Wong, C. Ting, and C. Yeh. From Network Management to Service Management: A Challenge to Telecom Service Providers. *Second International Conference on Innovative Computing, Information and Control 2007 (ICICIC '07)*, September 2007.

Appendix A

Acronyms

ALU	Alcatel-Lucent (Switzerland)
AM	Action Model
ATD	Actor Transaction Diagram
BCI	Business Component Identification
BCT	Bank Contents Table
BO	Back Office
BSS	Business Support System
CIAO	Cooperation & Interoperability - Architecture & Ontology
CM	Construction Model
CPB	Coordination & Production Bank
CTT	Customer Trouble Ticket
DEMO	Design and Engineering Methodology for Organizations
ETOM	Enhanced Telecom Operations Map
FAB	Fulfillment, Assurance, Billing
FACPS	Fault, Accounting, Configuration, Performance and Security
FO	Front Office
GSDP	Generic System Development Process
GSSF	Generic Service Specification Framework
IAM	Interaction Model

ICT	Information and Communication Technology
ISM	Interstriction Model
ITU	International Telecommunication Union
IUT	Information Use Table
KPI	Key Performance Indicator
NGOSS	New Generation Operations Systems and Software
NLS	Next Level Support
NTT	Network Trouble Ticket
OCD	Organization Construction Diagram
OFD	Object Fact Diagram
OPL	Object Property List
OS	Object System
OSR	Operations Support and Readiness
OSS	Operations Support System
PB	Production Bank
PIF	Performa-Informa-Forma
PM	Process Model
PSD	Process Structure Diagram
SLA	Service Level Agreement
SM	State Model
SNMP	Simple Network Management Protocol
SOA	Service Oriented Architecture
TAM	Telecom Applications Map
TM	TeleManagement (Forum)
TRT	Transaction Result Table
TSP	Telecom Service Provider
US	Using System
WOSL	World Ontology Specification Language

Appendix B

DEMO models and steps

This appendix gives and overview of the models and lists the steps that should be performed in order to model the B-organization of an enterprise, in the context of enterprise ontology, using the Design and Engineering Methodology for Organizations (DEMO). Figure B.1 gives an overview of these models.

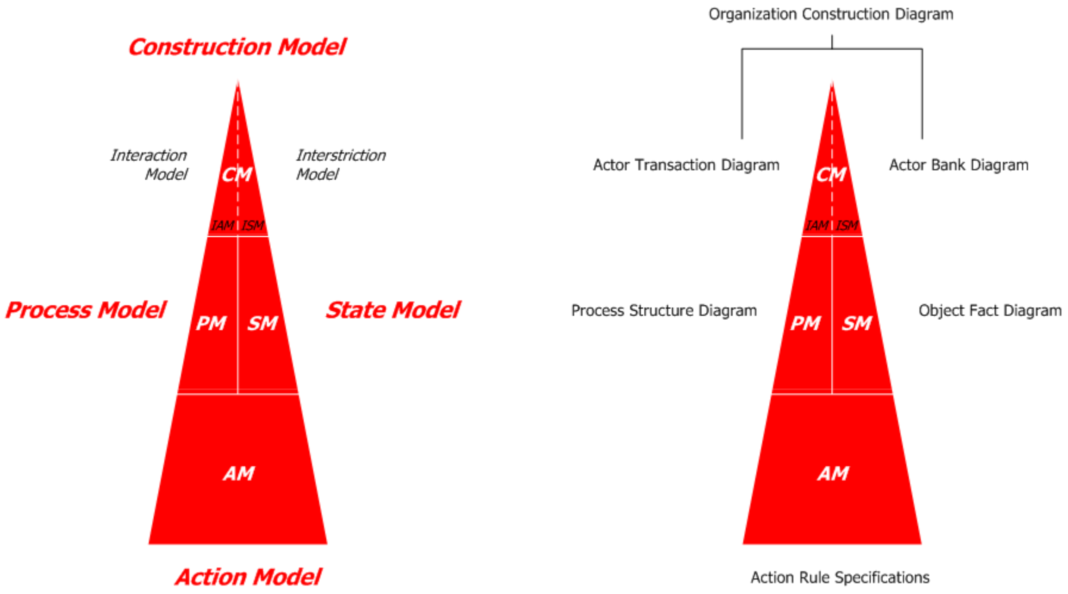


Figure B.1: Models in the B-organization

- Construction Model (CM), consisting of:
  - Interaction Model (IAM) — active influences between actor roles, i.e. the execution of transactions.
  - Interstriction Model (ISM) — passive influences between actor roles, i.e. taking the existing facts into account (as an actor).

- Process Model (PM) — the transactions in the IAM are divided into process steps, which is either the complete transaction pattern or a part of it. The simplest form is the basic transaction pattern. The PM also contains causal and conditional relationships between transactions.
- Action Model (AM) — contains the action rules that serve as guidelines for the actors in dealing with their agenda. It is the second level of detailing of the CM, as it details the steps of the PM. The AM contains all the information of the other models, but is not easily accessible.
- State Model (SM) — specifies the state space of the P-world.

The logical sequence of producing the models above is counterclockwise, starting with the IAM. The following lists shows which sub-models constitute a model:

1. IAM: Actor Transaction Diagram (ATD) and Transaction Result Table (TRT)
2. PM: Process Structure Diagram (PSD)
3. AM: Action Rule Specification (ARS)
4. SM: Object Fact Diagram (OFD) and Object Property List (OPL)
5. Complete the PM: Information Use Table (IUT)
6. ISM: Organization Construction Diagram (OCD) and Bank Contents Table (BCT)

Now that the models and their respective sub-models have been listed, the required set of steps are the following:

1. Performa-Informa-Forma analysis
2. Coordination-Actors-Production analysis (solitary for performa items)
3. Transaction pattern analysis — results so far structured in transactions
4. Result structure analysis — identify sequence of transactions
5. Construction analysis — initiator and executor for transactions are identified
6. Organization synthesis — divide among organization and environment

Appendix C

BCI input and results

C.1 BCI input data

The following tables show the input used in the BCI-3D tool. This input consists of function steps, information objects and the relationships between steps and information objects.

Table C.1: (B) Process steps

Function	**Shortcut**
request fault checking	T00/rq
promise fault checking	T00/pm
produce fault checking	T00/ex
state fault checking	T00/st
accept fault checking	T00/ac
request fault handling	T01/rq
promise fault handling	T01/pm
produce fault handling	T01/ex
state fault handling	T01/st
accept fault handling	T01/ac
reject fault handling	T01/rj
request fo fault handling	T02/rq
promise fo fault handling	T02/pm
produce fo fault handling	T02/ex
state fo fault handling	T02/st
accept fo fault handling	T02/ac
request bo fault handling	T03/rq
promise bo fault handling	T03/pm
produce bo fault handling	T03/ex
state bo fault handling	T03/st
accept bo fault handling	T03/ac
request nls fault handling	T04/rq
promise nls fault handling	T04/pm
produce nls fault handling	T04/ex
state nls fault handling	T04/st
accept nls fault handling	T04/ac
request criticalness determining	T05/rq
promise criticalness determining	T05/pm
produce criticalness determining	T05/ex
state criticalness determining	T05/st
accept criticalness determining	T05/ac

Table C.2: (B) Relations: fun-fun

Fun-fun	**Standard**	**Optional**
request fault checking	T00/rq,T00/pm	
promise fault checking	T00/ex	
produce fault checking	T00/st	
state fault checking	T00/ac	T01/rq
accept fault checking		
request fault handling	T01/pm	
promise fault handling	T01/ex	T02/rq, T03/rq, T04/rq, T05/rq
produce fault handling	T01/st	
state fault handling		T01/ac,T01/rj
accept fault handling		
reject fault handling	T01/rq	
request fo fault handling	T02/pm	
promise fo fault handling	T02/ex	
produce fo fault handling	T02/st	
state fo fault handling	T02/ac	
accept fo fault handling		T01/ex
request bo fault handling	T03/pm	
promise bo fault handling	T03/ex	
produce bo fault handling	T03/st	
state bo fault handling	T03/ac	
accept bo fault handling		T01/ex
request nls fault handling	T04/pm	
promise nls fault handling	T04/ex	
produce nls fault handling	T04/st	
state nls fault handling	T04/ac	
accept nls fault handling		T01/ex
request criticalness determining	T05/pm	
promise criticalness determining	T05/ex	
produce criticalness determining	T05/st	
state criticalness determining	T05/ac	
accept criticalness determining		T01/ex

Table C.3: (B) Information objects

IO	**Shortcut**
Network Element	NE
Fault	F
FO fault	FF
BO fault	BF
NLS fault	NF
Error Description	ED

Table C.4: (B) Relations: io-io

IO-IO	**Related to**	**State of**
Network Element		
Fault	NE,ED	
FO fault		F
BO fault		F
NLS fault		F
Error Description		

Table C.5: (B) Relations: io-fun

IO-fun	**Create**	**Use**
Network Element		T00/rq, T00/ex, T02/ex, T03/ex, T04/ex
Fault	T00/st	T01/ex, T00/ex, T05/ex, T01/ac
FO fault	T02/rq	T02/ex
BO fault	T03/rq	T03/ex
NLS fault	T04/rq	T04/ex
Error Description		T00/ex

Table C.6: (I) Process steps (1)

Function	**Shortcut**
request network element information	IT07/rq
promise network element information	IT07/pm
produce network element information	IT07/ex
state network element information	IT07/st
accept network element information	IT07/ac
request error information	IT08/rq
promise error information	IT08/pm
produce error information	IT08/ex
state error information	IT08/st
accept error information	IT08/ac
request fault discovery	IT09/rq
promise fault discovery	IT09/pm
produce fault discovery	IT09/ex
state fault discovery	IT09/st
accept fault discovery	IT09/ac
request fault magnitude information	IT10/rq
promise fault magnitude information	IT10/pm
produce fault magnitude information	IT10/ex
state fault magnitude information	IT10/st
accept fault magnitude information	IT10/ac
request resource information	IT11/rq
promise resource information	IT11/pm
produce resource information	IT11/ex
state resource information	IT11/st
accept resource information	IT11/ac
request fo fault information	IT12/rq
promise fo fault information	IT12/pm
produce fo fault information	IT12/ex
state fo fault information	IT12/st
accept fo fault information	IT12/ac

Table C.7: (I) Process steps (2)

Function	**Shortcut**
request bo fault information	IT13/rq
promise bo fault information	IT13/pm
produce bo fault information	IT13/ex
state bo fault information	IT13/st
accept bo fault information	IT13/ac
request nls fault information	IT14/rq
promise nls fault information	IT14/pm
produce nls fault information	IT14/ex
state nls fault information	IT14/st
accept nls fault information	IT14/ac
request determining criticalness	IT15/rq
promise determining criticalness	IT15/pm
produce determining criticalness	IT15/ex
state determining criticalness	IT15/st
accept determining criticalness	IT15/ac
request updating sunrise heldesk	IT16/rq
promise updating sunrise heldesk	IT16/pm
produce updating sunrise heldesk	IT16/ex
state updating sunrise heldesk	IT16/st
accept updating sunrise heldesk	IT16/ac

Table C.8: (I) Relations: fun-fun (1)

Fun-fun	**Standard**
request network element information	IT07/pm
promise network element information	IT07/ex
produce network element information	IT07/st
state network element information	IT07/ac
accept network element information	
request error information	IT08/pm
promise error information	IT08/ex
produce error information	IT08/st
state error information	IT08/ac
accept error information	
request fault discovery	IT09/pm
promise fault discovery	IT09/ex
produce fault discovery	IT09/st
state fault discovery	IT09/ac
accept fault discovery	
request fault magnitude information	IT10/pm
promise fault magnitude information	IT10/ex
produce fault magnitude information	IT10/st
state fault magnitude information	IT10/ac
accept fault magnitude information	
request resource information	IT11/pm
promise resource information	IT11/ex
produce resource information	IT11/st
state resource information	IT11/ac
accept resource information	
request fo fault information	IT12/pm
promise fo fault information	IT12/ex
produce fo fault information	IT12/st
state fo fault information	IT12/ac
accept fo fault information	

Table C.9: (I) Relations: fun-fun (2)

Fun-fun	**Standard**
request bo fault information	IT13/pm
promise bo fault information	IT13/ex
produce bo fault information	IT13/st
state bo fault information	IT13/ac
accept bo fault information	
request nls fault information	IT14/pm
promise nls fault information	IT14/ex
produce nls fault information	IT14/st
state nls fault information	IT14/ac
accept nls fault information	
request determining criticalness	IT15/pm
promise determining criticalness	IT15/ex
produce determining criticalness	IT15/st
state determining criticalness	IT15/ac
accept determining criticalness	
request updating sunrise heldesk	IT16/pm
promise updating sunrise heldesk	IT16/ex
produce updating sunrise heldesk	IT16/st
state updating sunrise heldesk	IT16/ac
accept updating sunrise heldesk	

Table C.10: (I) Information objects

IO	**Shortcut**
Fault	F
FO fault	FF
BO fault	BF
NLS fault	NF
Network Element	NE
Error Description	ED
Resource Information	RES

Table C.11: (I) Relations: io-io

IO-IO	**Related to**	**State of**
Fault	NE,ED	
FO fault		F
BO fault		F
NLS fault		F
Network Element		
Error Description		
Resource Information		

Table C.12: (I) Relations: io-fun

IO-fun	**Create**	**Use**
Fault	IT09/st	IT09/ex, IT10/ex, IT15/ex
FO fault	IT12/pm	
BO fault	IT13/pm	
NLS fault	IT14/pm	
Network Element		IT07/ex
Error Description		IT08/ex
Resource Information		IT11/ex

Table C.13: (D) Process steps (1)

Function	**Shortcut**
request network elements retrieval	DT17/rq
promise network elements retrieval	DT17/pm
produce network elements retrieval	DT17/ex
state network elements retrieval	DT17/st
accept network elements retrieval	DT17/ac
request error descriptions retrieval	DT18/rq
promise error descriptions retrieval	DT18/pm
produce error descriptions retrieval	DT18/ex
state error descriptions retrieval	DT18/st
accept error descriptions retrieval	DT18/ac
request magnitude criteria retrieval	DT19/rq
promise magnitude criteria retrieval	DT19/pm
produce magnitude criteria retrieval	DT19/ex
state magnitude criteria retrieval	DT19/st
accept magnitude criteria retrieval	DT19/ac
request resource retrieval	DT20/rq
promise resource retrieval	DT20/pm
produce resource retrieval	DT20/ex
state resource retrieval	DT20/st
accept resource retrieval	DT20/ac
request FO fault data retrieval	DT21/rq
promise FO fault data retrieval	DT21/pm
produce FO fault data retrieval	DT21/ex
state FO fault data retrieval	DT21/st
accept FO fault data retrieval	DT21/ac
request BO fault data retrieval	DT22/rq
promise BO fault data retrieval	DT22/pm
produce BO fault data retrieval	DT22/ex
state BO fault data retrieval	DT22/st
accept BO fault data retrieval	DT22/ac
request NLS fault data retrieval	DT23/rq
promise NLS fault data retrieval	DT23/pm
produce NLS fault data retrieval	DT23/ex
state NLS fault data retrieval	DT23/st
accept NLS fault data retrieval	DT23/ac

Table C.14: (D) Process steps (2)

Function	**Shortcut**
request criticalness criteria retrieval	DT24/rq
promise criticalness criteria retrieval	DT24/pm
produce criticalness criteria retrieval	DT24/ex
state criticalness criteria retrieval	DT24/st
accept criticalness criteria retrieval	DT24/ac
request creating CTT	DT25/rq
promise creating CTT	DT25/pm
produce creating CTT	DT25/ex
state creating CTT	DT25/st
accept creating CTT	DT25/ac
state updating CTT	DT26/rq
state updating CTT	DT26/pm
state updating CTT	DT26/ex
state updating CTT	DT26/st
state updating CTT	DT26/ac
request closing CTT	DT27/rq
promise closing CTT	DT27/pm
produce closing CTT	DT27/ex
state closing CTT	DT27/st
accept closing CTT	DT27/ac
request creating NTT	DT28/rq
promise creating NTT	DT28/pm
produce creating NTT	DT28/ex
state creating NTT	DT28/st
accept creating NTT	DT28/ac
request updating NTT	DT29/rq
promise updating NTT	DT29/pm
produce updating NTT	DT29/ex
state updating NTT	DT29/st
accept updating NTT	DT29/ac
request closing NTT	DT30/rq
promise closing NTT	DT30/pm
produce closing NTT	DT30/ex
state closing NTT	DT30/st
accept closing NTT	DT30/ac

Table C.15: (D) Relations: fun-fun (1)

Fun-fun	**Standard**
request network elements retrieval	DT17/pm
promise network elements retrieval	DT17/ex
produce network elements retrieval	DT17/st
state network elements retrieval	DT17/ac
accept network elements retrieval	
request error descriptions retrieval	DT18/pm
promise error descriptions retrieval	DT18/ex
produce error descriptions retrieval	DT18/st
state error descriptions retrieval	DT18/ac
accept error descriptions retrieval	
request magnitude criteria retrieval	DT19/pm
promise magnitude criteria retrieval	DT19/ex
produce magnitude criteria retrieval	DT19/st
state magnitude criteria retrieval	DT19/ac
accept magnitude criteria retrieval	
request resource retrieval	DT20/pm
promise resource retrieval	DT20/ex
produce resource retrieval	DT20/st
state resource retrieval	DT20/ac
accept resource retrieval	
request FO fault data retrieval	DT21/pm
promise FO fault data retrieval	DT21/ex
produce FO fault data retrieval	DT21/st
state FO fault data retrieval	DT21/ac
accept FO fault data retrieval	
request BO fault data retrieval	DT22/pm
promise BO fault data retrieval	DT22/ex
produce BO fault data retrieval	DT22/st
state BO fault data retrieval	DT22/ac
accept BO fault data retrieval	
request NLS fault data retrieval	DT23/pm
promise NLS fault data retrieval	DT23/ex
produce NLS fault data retrieval	DT23/st
state NLS fault data retrieval	DT23/ac
accept NLS fault data retrieval	

Table C.16: (D) Relations: fun-fun (2)

Fun-fun	**Standard**
request criticalness criteria retrieval	DT24/pm
promise criticalness criteria retrieval	DT24/ex
produce criticalness criteria retrieval	DT24/st
state criticalness criteria retrieval	DT24/ac
accept criticalness criteria retrieval	
request creating CTT	DT25/pm
promise creating CTT	DT25/ex
produce creating CTT	DT25/st
state creating CTT	DT25/ac
accept creating CTT	
state updating CTT	DT26/pm
state updating CTT	DT26/ex
state updating CTT	DT26/st
state updating CTT	DT26/ac
state updating CTT	
request closing CTT	DT27/pm
promise closing CTT	DT27/ex
produce closing CTT	DT27/st
state closing CTT	DT27/ac
accept closing CTT	
request creating NTT	DT28/pm
promise creating NTT	DT28/ex
produce creating NTT	DT28/st
state creating NTT	DT28/ac
accept creating NTT	
request updating NTT	DT29/pm
promise updating NTT	DT29/ex
produce updating NTT	DT29/st
state updating NTT	DT29/ac
accept updating NTT	
request closing NTT	DT30/pm
promise closing NTT	DT30/ex
produce closing NTT	DT30/st
state closing NTT	DT30/ac
accept closing NTT	

Table C.17: (D) Information objects

IO	**Shortcut**
Fault	F
FO fault	FF
BO fault	BF
NLS fault	NF
Network Element	NE
Error Description	ED
Resource Information	RES

Table C.18: (D) Relations: io-io

IO-IO	**Related to**	**State of**
Fault	NE,ED	
FO fault		F
BO fault		F
NLS fault		F
Network Element		
Error Description		
Resource Information		

Table C.19: (D) Relations: io-fun

IO-fun	**Create**	**Use**
Fault	IT09/st	IT09/ex, IT10/ex, IT15/ex
FO fault	IT12/pm	
BO fault	IT13/pm	
NLS fault	IT14/pm	
Network Element		IT07/ex
Error Description		IT08/ex
Resource Information		IT11/ex

C.2 Optimization results

This section shows the optimization results, generated by the BCI-3D tool.

C.2.1 B-organization

```
##### Optimization Results (2010-7-21 12:18) #####


### General Information ###
- 1 Internal Components (containing 6 IOs and 31 BPSs)
- 0 External Components (containing 0 IOs and 0 BPSs)
Costs of this Partitioning: 0.0
Normalized Partitioning Costs: 0.0


### Heuristics ###
Starting Heuristic: StartPartitionGreedy
Improvement Heuristic: Kernighan-Lin


### Parameter Settings ###
IO-IO
- related-to: 10
- part-of: 0
- state-of: 50
BPS-BPS
- standard: 10
- optional: 5
- main: 0
BPS-IO
- create: 50
- use: 10
ACT


### Components ###
Component 1:
- FO fault (FF)
- Fault (F)
- BO fault (BF)
- NLS fault (NF)
- state fault checking (T00/st)
- request fo fault handling (T02/rq)
```

```
- request nls fault handling (T04/rq)
- request bo fault handling (T03/rq)
- Network Element (NE)
- produce fault checking (T00/ex)
- produce fault handling (T01/ex)
- produce criticalness determining (T05/ex)
- Error Description (ED)
- accept fault handling (T01/ac)
- produce nls fault handling (T04/ex)
- produce fo fault handling (T02/ex)
- produce bo fault handling (T03/ex)
- request fault checking (T00/rq)
- promise fault handling (T01/pm)
- promise fo fault handling (T02/pm)
- promise bo fault handling (T03/pm)
- accept fault checking (T00/ac)
- promise nls fault handling (T04/pm)
- promise fault checking (T00/pm)
- state fault handling (T01/st)
- request fault handling (T01/rq)
- state fo fault handling (T02/st)
- state bo fault handling (T03/st)
- state nls fault handling (T04/st)
- state criticalness determining (T05/st)
- promise criticalness determining (T05/pm)
- reject fault handling (T01/rj)
- accept bo fault handling (T03/ac)
- accept nls fault handling (T04/ac)
- request criticalness determining (T05/rq)
- accept criticalness determining (T05/ac)
- accept fo fault handling (T02/ac)
```

C.2.2 I-organization

```
##### Optimization Results (2010-7-21 13:41) #####


### General Information ###
- 4 Internal Components (containing 7 IOs and 50 BPSs)
- 0 External Components (containing 0 IOs and 0 BPSs)
Costs of this Partitioning: 10.0
Normalized Partitioning Costs: 0.012048192771084338
```

```
### Heuristics ###
Starting Heuristic: StartPartitionGreedy
Improvement Heuristic: Kernighan-Lin


### Parameter Settings ###
IO-IO
- related-to: 10
- part-of: 0
- state-of: 50
BPS-BPS
- standard: 10
- optional: 0
- main: 0
BPS-IO
- create: 50
- use: 10
ACT


### Components ###
Component 1:
- FO fault (FF)
- Fault (F)
- BO fault (BF)
- NLS fault (NF)
- state fault discovery (IT09/st)
- promise fo fault information (IT12/pm)
- promise bo fault information (IT13/pm)
- promise nls fault information (IT14/pm)
- produce determining criticalness (IT15/ex)
- produce fault discovery (IT09/ex)
- produce fault magnitude information (IT10/ex)
- Network Element (NE)
- Error Description (ED)
- produce fo fault information (IT12/ex)
- produce bo fault information (IT13/ex)
- produce nls fault information (IT14/ex)
- request bo fault information (IT13/rq)
- request nls fault information (IT14/rq)
- accept fault discovery (IT09/ac)
- request fo fault information (IT12/rq)
- produce network element information (IT07/ex)
```

- promise fault discovery (IT09/pm)
- promise determining criticalness (IT15/pm)
- promise fault magnitude information (IT10/pm)
- promise network element information (IT07/pm)
- state network element information (IT07/st)
- state determining criticalness (IT15/st)
- state fault magnitude information (IT10/st)
- state fo fault information (IT12/st)
- state nls fault information (IT14/st)
- state bo fault information (IT13/st)
- accept bo fault information (IT13/ac)
- accept fo fault information (IT12/ac)
- accept network element information (IT07/ac)
- request determining criticalness (IT15/rq)
- request fault magnitude information (IT10/rq)
- accept determining criticalness (IT15/ac)
- request network element information (IT07/rq)
- accept nls fault information (IT14/ac)
- accept fault magnitude information (IT10/ac)
- request fault discovery (IT09/rq) Component 2:
- promise error information (IT08/pm)
- produce error information (IT08/ex)
- state error information (IT08/st)
- accept error information (IT08/ac)
- request error information (IT08/rq)

Component 3:
- produce resource information (IT11/ex)
- state resource information (IT11/st)
- promise resource information (IT11/pm)
- Resource Information (RES)
- request resource information (IT11/rq)
- accept resource information (IT11/ac)

Component 4:
- produce updating sunrise heldesk (IT16/ex)
- state updating sunrise heldesk (IT16/st)
- promise updating sunrise heldesk (IT16/pm)
- request updating sunrise heldesk (IT16/rq)
- accept updating sunrise heldesk (IT16/ac)

C.2.3 D-organization

Optimization Results (2004-5-17 3:29)

```
### General Information ###
- 8 Internal Components (containing 9 IOs and 70 BPSs)
- 0 External Components (containing 0 IOs and 0 BPSs)
Costs of this Partitioning: 70.0
Normalized Partitioning Costs: 0.051094890510948905


### Heuristics ###
Starting Heuristic: StartPartitionGreedy
Improvement Heuristic: Kernighan-Lin


### Parameter Settings ###
IO-IO
- related-to: 10
- part-of: 0
- state-of: 50
BPS-BPS
- standard: 10
- optional: 0
- main: 0
BPS-IO
- create: 50
- use: 10
ACT


### Components ###
Component 1:
- FO fault (FF)
- Fault (F)
- BO fault (BF)
- NLS fault (NF)
- produce BO fault data retrieval (DT22/ex)
- produce FO fault data retrieval (DT21/ex)
- produce NLS fault data retrieval (DT23/ex)
- produce updating NTT (DT29/ex)
- produce closing NTT (DT30/ex)
- produce closing CTT (DT27/ex)
- state updating CTT (DT26/ex)
- promise FO fault data retrieval (DT21/pm)
- promise NLS fault data retrieval (DT23/pm)
- state FO fault data retrieval (DT21/st)
```

- promise BO fault data retrieval (DT22/pm)
- state NLS fault data retrieval (DT23/st)
- state BO fault data retrieval (DT22/st)
- state closing CTT (DT27/st)
- promise updating NTT (DT29/pm)
- state updating NTT (DT29/st)
- promise closing NTT (DT30/pm)
- state closing NTT (DT30/st)
- request BO fault data retrieval (DT22/rq)
- request FO fault data retrieval (DT21/rq)
- request updating NTT (DT29/rq)
- accept BO fault data retrieval (DT22/ac)
- request NLS fault data retrieval (DT23/rq)
- accept updating NTT (DT29/ac)
- accept FO fault data retrieval (DT21/ac)
- accept closing NTT (DT30/ac)
- accept closing CTT (DT27/ac)
- accept NLS fault data retrieval (DT23/ac)
- request closing NTT (DT30/rq)
- Customer trouble ticket (CTT)
- produce creating NTT (DT28/ex)
- promise creating NTT (DT28/pm)
- request creating NTT (DT28/rq)
- Network trouble ticket (NTT)
- produce creating CTT (DT25/ex)

Component 2:
- state creating NTT (DT28/st)
- accept creating NTT (DT28/ac)
- state updating CTT (DT26/pm)
- state updating CTT (DT26/st)
- state updating CTT (DT26/rq)
- state updating CTT (DT26/ac)

Component 3:
- promise creating CTT (DT25/pm)
- state creating CTT (DT25/st)
- request creating CTT (DT25/rq)
- accept creating CTT (DT25/ac)
- request closing CTT (DT27/rq)
- promise closing CTT (DT27/pm)

Component 4:
- Network Element (NE)
- produce network elements retrieval (DT17/ex)
- promise network elements retrieval (DT17/pm)
- state network elements retrieval (DT17/st)

- request network elements retrieval (DT17/rq)
- accept network elements retrieval (DT17/ac)

Component 5:
- Error Description (ED)
- produce error descriptions retrieval (DT18/ex)
- promise error descriptions retrieval (DT18/pm)
- state error descriptions retrieval (DT18/st)
- accept error descriptions retrieval (DT18/ac)
- request error descriptions retrieval (DT18/rq)

Component 6:
- Resource Information (RES)
- produce resource retrieval (DT20/ex)
- state resource retrieval (DT20/st)
- promise resource retrieval (DT20/pm)
- accept resource retrieval (DT20/ac)
- request resource retrieval (DT20/rq)

Component 7:
- produce magnitude criteria retrieval (DT19/ex)
- state magnitude criteria retrieval (DT19/st)
- promise magnitude criteria retrieval (DT19/pm)
- request magnitude criteria retrieval (DT19/rq)
- accept magnitude criteria retrieval (DT19/ac)

Component 8:
- promise criticalness criteria retrieval (DT24/pm)
- produce criticalness criteria retrieval (DT24/ex)
- state criticalness criteria retrieval (DT24/st)
- request criticalness criteria retrieval (DT24/rq)
- accept criticalness criteria retrieval (DT24/ac)