A Framework for Analyzing and Monitoring the Impact of Dependencies on Quality

Doctoral Dissertation by

Olav Skjelkvåle Ligaarden

Submitted to the Faculty of Mathematics and Natural Sciences at the University of Oslo in partial fulfillment of the requirements for the degree Ph.D. in Computer Science

October 2012
Abstract

In today’s society, we are dependent on a number of services provided by interconnected systems. These services may be anything from electricity to services provided by social media platforms. Interconnected systems are challenging to analyze from a quality perspective in general and from a security perspective in particular. The systems depend on each other through services. Thus, the quality of services provided by one system is often directly linked to the quality of services provided by another. Moreover, the systems may be under different managerial control and within different jurisdictions, and the systems may evolve rapidly in a manner that may be difficult to predict. All of this makes it challenging to assess risk to the quality of services.

In this thesis we present a framework for analyzing and monitoring the impact of dependencies on quality. More specifically, the framework should be used in the context of interconnected systems to analyze and monitor the impact of service dependencies on quality of services. The framework is the result of the integration of three artifacts: (1) a method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators; (2) a method for capturing and monitoring the impact of service dependencies on the quality of provided services; and (3) an architectural pattern for constructing enterprise level monitoring tools based on indicators. The three artifacts may be viewed as contributions on their own, since they can be used independently of each other. In addition, the thesis contributes in terms of two industrial case studies: (1) an empirical study on trust-based decisions in interconnected systems; and (2) an empirical study on the design of indicators for monitoring risk. The industrial case studies have mainly been carried out to support the development of the artifacts, but since the industrial case studies also provide insight into issues of a more general nature, they may be seen as contributions on their own.
Abstract
First of all, I would like to express my deepest gratitude to my main supervisor Ketil Stølen for all his encouragement and involvement. I thank him for sharing his extensive knowledge and passion for science, for all the interesting discussions, for everything I have learnt from him, and for all the useful advice and feedback he has provided on my work. The numerous hours he has spent on supervising my work, far exceeds what can reasonably be expected from any supervisor. He has been an excellent supervisor, and I will without doubt recommend Ketil Stølen as a supervisor to anyone who wants to pursue a master’s degree or a PhD within his fields of expertise.

I also wish to express my gratitude to my second supervisor Birger Møller-Pedersen for all the valuable advice, feedback, and support I have received from him during his participation in the DIGIT project.

I am grateful to Atle Refsdal who co-authored all of the papers in this thesis. I would like to thank him for the collaboration and for all the feedback, support, and advice he has provided.

I have also co-authored papers with Tormod Vaksvik Håvaldsrud, Mass Soldal Lund, Fredrik Seehusen, Jon Ølnes, and Per Myrseth. I wish to thank them all for their collaboration and support.

I would also like to thank guest scientists in the DIGIT and EMERGENCY projects: Fabio Massacci, Christian Damsgaard Jensen, Audun Jøssang, late William Winsborough, Jonas Landgren, and Isabelle Simplot-Ryl. I am also thankful to the members of the DIGIT advisory board. Both the guest scientists and the members of the advisory board have on a number of occasions provided useful feedback and advice.

I have on many occasions presented my work to Aida Omerovic, André Alexander-Hauge, Erik Gosta Nilsson, Amelia Karahasancvic, Bjørnar Solhaug, Gyrd Brændeland, and Heidi Elisabeth Iuell Dahl. Thanks for the many interesting discussions and for all the useful feedback and advice you have provided.

My thanks also go to the people of the Department of Networked Systems and Services at SINTEF ICT and research director Bjørn Skjellaug for providing a good work environment. I would also like to thank the many members of the administrative and academic staff at the University of Oslo for helping me whenever I needed assistance.

I would like to thank all my friends for taking my mind off work, for all the joyful time we spend together, and for your support. I am also grateful for all the love and support I have received from my parents, Lars Ligaarden and Tove Randi Skjelkvåle, my brother Lars Anton Skjelkvåle Ligaarden, and my sister Ingeborg Skjelkvåle Ligaarden. I would also like to thank my sister for reading and commenting on my thesis.

The research on which this thesis reports has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER and NESSoS projects, both funded from the European Community’s Seventh Frame-
work Programme (FP7/2007-2013) under grant agreements FP7-216917 and FP7-256980, respectively.
List of original publications


   The first version of this paper received a **best paper award** at the First International Conference on Business Intelligence and Technology (BUSTECH’2011).


The publications 1–5 are available as Chapters 9–13 in Part II of this thesis. In the case of publications 1, 2, and 5, we have included the full technical reports which are extended, slightly revised versions of the published papers.
List of original publications
Contents

Abstract iii
Acknowledgements v
List of original publications vii
Contents ix
List of figures xiii

I Overview 1

1 Introduction 3
1.1 Objective .................................. 4
1.2 Contribution ................................. 4
1.3 Organization ................................ 6

2 Problem characterization 9
2.1 Conceptual clarification .................... 9
2.1.1 System and service ......................... 9
2.1.2 Quality and quality of service .............. 9
2.1.3 Service dependency ........................ 10
2.1.4 Indicator and metric ....................... 10
2.1.5 Trust ..................................... 10
2.1.6 Architectural pattern ..................... 11
2.2 Success criteria ................................ 11
2.2.1 Framework for analyzing and monitoring the impact of dependencies on quality . .. 12
2.2.2 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators . .. 12
2.2.3 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services . .. 13
2.2.4 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators . .. 13
## 3 Research method
- 3.1 The standing of computer science ........................................ 15
- 3.2 Classical versus technology research .................................... 16
- 3.3 Strategies for evaluation ...................................................... 18
- 3.4 Our research method .......................................................... 19

## 4 State of the art
- 4.1 Indicators ............................................................................ 24
  - 4.1.1 Measurement of the fulfillment of business objectives .......... 24
  - 4.1.2 Specification of the design and deployment of indicators ...... 24
  - 4.1.3 Validation of indicators .................................................. 25
- 4.2 Dependencies ...................................................................... 26
  - 4.2.1 Modeling and analysis of dependencies in general .......... 26
  - 4.2.2 Modeling and analysis of service dependencies .............. 27
- 4.3 Risk analysis ....................................................................... 28
  - 4.3.1 Asset-based risk analysis methods ................................ 28
  - 4.3.2 Analysis of risk in the context of dependencies .......... 28
  - 4.3.3 Validation of expert judgments in risk analysis ............ 29
  - 4.3.4 Indicator-based risk analysis ......................................... 30
- 4.4 Monitoring .......................................................................... 30
  - 4.4.1 Monitoring of quality ..................................................... 30
  - 4.4.2 Dynamic risk monitoring .............................................. 31
  - 4.4.3 Patterns and the implementation of monitoring tools ...... 32
- 4.5 Trust .................................................................................... 33
  - 4.5.1 Estimating the trustworthiness of external services ........ 33
  - 4.5.2 Trust-based decisions in interconnected systems .......... 34

## 5 Summary of contribution
- 5.1 The ValidKI Framework ....................................................... 38
- 5.2 Overview of artifacts .......................................................... 40
  - 5.2.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators .......... 40
  - 5.2.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services .......... 41
  - 5.2.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators .......... 42
- 5.3 Overview of industrial case studies ..................................... 43
  - 5.3.1 Empirical study on trust-based decisions in interconnected systems .......... 43
  - 5.3.2 Empirical study on the design of indicators for monitoring risk . 44

## 6 Overview of research papers
- 6.1 Paper A: ValidKI: A method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators .......... 47
- 6.2 Paper B: Using indicators to monitor risk in interconnected systems: How to capture and measure the impact of service dependencies on the quality of provided services .......... 48
7 Discussion

7.1 Fulfillment of the success criteria

7.1.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

7.1.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

7.1.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

7.1.4 Framework for analyzing and monitoring the impact of dependencies on quality

7.2 How the artifacts relate to and extend the state of the art

7.2.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

7.2.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

7.2.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

8 Conclusion

8.1 What has been achieved

8.2 Directions for future work

Bibliography

II Research Papers

9 Paper A: Validki: A method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

10 Paper B: Using indicators to monitor risk in interconnected systems:
   How to capture and measure the impact of service dependencies on the quality of provided services

11 Paper C: An architectural pattern for enterprise level monitoring tools

12 Paper D: Experiences from using a UML-based method for trust analysis in an industrial project on electronic procurement
13 Paper E: Experiences from using indicators to validate expert judgments in security risk analysis 319
List of Figures

3.1 The main steps of the method for technology research (adopted from [1]) 17
3.2 The main steps of the method for classical research (adopted from [1]) . 18
3.3 Our research method (adopted from [2] and modified) . . . . . . . . . . 20
3.4 How the artifacts and the framework are related to the empirical studies 22

5.1 Relations between the ValidKI Framework and Artifacts 1–3 . . . . . . 38
5.2 The ValidKI Framework . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
Part I

Overview
Chapter 1

Introduction

Over time we have gradually become more and more dependent on services that are provided by interconnected systems. Such systems may, for instance, be found in power grids or in the Internet. They provide services like electricity, communication services, etc. The importance of the quality of these services in general and the security in particular is not new. In 1988, the Morris worm [3] infected about 4% of the approximate Internet population of 60000 computers. Much has however changed since the early days of the Internet. Since then, the Internet and other networks have increased much in size and become more interconnected, while incidents with respect to interconnected systems have increased both with respect to number and severity [4]. Moreover, the realization of the Internet of Things [5] will lead to even more interconnected networks. The Internet of Things refers to a world where physical objects and beings have virtual components that can produce and consume services. Such extreme interconnection will in particular result in new challenges for the security of services [6].

Despite of the importance of quality in general, security in particular, and the interconnected systems that surround us, there is often a lack of understanding of the interconnections’ potential effect on quality of services. On January 25, 2003, the David-Besse nuclear power plant in Ohio was infected by a SQL slammer worm [7] due to a network connection that circumvented the firewall. The infection resulted in the internal network being overloaded, which again resulted in the unavailability of crucial control systems for about five hours. Although the operators were burdened by these losses, the plant was not affected because of analogue backup systems which remained unaffected. Later the same year, the so-called “Northeast blackout” [8] occurred. This blackout left 50 million people in North America without electrical power and affected other critical infrastructures such as transportation, communication, and water supply. A major contributing factor to the incident was a software bug, while the severe consequences were the result of the many interconnections.

Interconnected systems are often so-called system of systems (SoS). An SoS may be thought of as a kind of “super system” comprising a set of interconnected systems that work together towards some common goal. The common goal may be as simple as enabling all the individual systems to achieve their capabilities, or to construct a set of new capabilities not achievable by the individual systems alone.

Interconnected systems, such as SoS, are challenging from a quality perspective for the following reasons:

1. The services provided by one system may rely on services provided by other
Introduction

systems, resulting in so-called service dependencies. Changes in the quality attributes of one service may easily cause the quality attributes of its dependent services to change as well. This means that in order to capture the impact of risk to quality of services provided by one system, we not only need to capture the risks arising in the system in question, but also risks which are solely or partially due to dependencies on other services.

2. The systems may be under different managerial control and within different jurisdictions. For the systems that are outside our control, we have limited knowledge of their risks, structure, and behavior. Thus, we need means for capturing the impact of service dependencies involving systems for which we have insufficient information on risk.

3. Such a large number of systems, controlled and operated by different parties, evolve rapidly in a manner that may be difficult to predict. Thus, there is a need for updating the risk picture as the interconnected systems change with values based on observable properties of the interconnected systems. It is however not trivial to achieve this. The data from which the updates are calculated may be associated with many different sources of uncertainty. Thus, the validity of the data must be taken into account in order to ensure the correctness of the risk picture.

Traditional approaches to risk analysis such as [9–11] lack capabilities for addressing these challenges in a satisfactory manner. Moreover, a critical infrastructure may often be thought of as a set of interconnected systems that interact by the use of services. Hence, research on critical infrastructure protection is relevant in this thesis. In [12–14], state of the art on critical infrastructure protection are presented. With respect to the above mentioned challenges, the approaches are either not relevant at all or they lack capabilities for addressing them in a satisfactory manner. Based on all of this, we see the need for new artifacts for addressing the above mentioned challenges.

1.1 Objective

The purpose of this thesis has been to develop a framework for analyzing and monitoring the impact of dependencies on quality. More specifically, the framework should be useful in the context of interconnected systems to analyze and monitor the impact of service dependencies on quality of services. The overall objective has been to: develop a framework that is:

1. well-suited to analyze the impact of service dependencies on quality of services;

2. well-suited to support the set-up of monitoring of the impact of service dependencies on quality of services; and

3. applicable in an industrial context within acceptable effort.

1.2 Contribution

The framework is the result of the development and integration of three artifacts. Since the artifacts may be employed and used in practice independently of each other, they
may be seen as contributions on their own. Besides contributing in terms of new artifacts, the thesis also contributes in terms of empirical results from two industrial case studies. The industrial case studies were mainly carried out to support the development of the artifacts, but since they also provide insight into issues of a more general nature, they may be seen as contributions on their own. The main contributions of this thesis are: (1) a method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators; (2) a method for capturing and monitoring the impact of service dependencies on the quality of provided services; (3) an architectural pattern for constructing enterprise level monitoring tools based on indicators; (4) an empirical study on trust-based decisions in interconnected systems; and (5) an empirical study on the design of indicators for monitoring risk.

1. **Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators:** The method takes business objectives focusing on quality as input, and delivers valid indicators as output. By valid indicators we mean that the indicators measure to what extent the business or relevant part thereof fulfills the business objectives. The method also results in deployment and design specifications for the different indicators. The deployment specifications document how sensors for gathering the data needed in the calculation of the indicators should be deployed in the relevant part of business, while the design specifications document how the indicators should be calculated based on data provided by the sensors. These specifications may be used to implement ICT-supported monitoring of the indicators.

Analysts will manage the application of the method and document its results, while domain experts will participate during the application of the method. The domain experts are supposed to communicate their knowledge in such a way that correct models are achieved.

2. **Method for capturing and monitoring the impact of service dependencies on the quality of provided services:** The method is used for capturing the impact of service dependencies on risk to the quality of provided services in interconnected systems, and for setting up monitoring of selected risks by the use of indicators for the purpose of providing a dynamic risk picture for the provided services. The result of applying the method is a risk picture that captures the impact of services dependencies on the quality of the provided services. The risk picture is parameterized by indicators, each defined by design and deployment specifications. These specifications may be used in the implementation of a risk monitor.

Analysts will manage the application of the method and document its results, while domain experts will participate during the application of the method. The domain experts are supposed to communicate their knowledge in such a way that correct models are achieved.

3. **Architectural pattern for constructing enterprise level monitoring tools based on indicators:** The pattern serves as a basis for constructing enterprise level monitoring tools based on indicators. These are tools that: collect low-level indicators from the ICT infrastructure or similar; aggregate the low-level indicators into high-level indicators, useful at the enterprise level; and present the high-level
indicators in a way that is understandable to the intended users. The pattern structures an enterprise level monitoring tool into a set of components, and it captures features that are general to a broad class of enterprise level monitoring tools.

The architectural pattern will be employed by developers of ICT-based enterprise level monitoring tools.

4. **Empirical study on trust-based decisions in interconnected systems:** The empirical study was conducted as part of an industrial project focusing on the use of a UML-based trust analysis method to model and analyze a public eProcurement system (used by public authorities to award contracts to economic operators). This system makes use of a Validation Authority (VA) service for validating electronic IDs and digital signatures. The goal of the trust analysis was to obtain a better understanding of the potential usefulness of a VA service for supporting trust-based decisions in systems which rely on electronically signed documents. The study gave strong indications that the trust analysis method is feasible in practice.

5. **Empirical study on the design of indicators for monitoring risk:** The empirical study was integrated in a commercial security risk analysis conducted in 2010. In this analysis, indicators were designed for the purpose of validating likelihood estimates obtained from expert judgments. The main result from the empirical study was the identification of several challenges related to the design of indicators for monitoring security risks.

### 1.3 Organization

The thesis is structured into two main parts. Part I provides the context and an overall view of the work, while Part II contains the research papers. Each of the papers is self-contained and can therefore be read separately. We have structured Part I into eight chapters:

**Chapter 1 – Introduction** provides the background and motivation for the thesis, brief explanations of the objective and contributions, and the structure of the thesis.

**Chapter 2 – Problem characterization** clarifies the interpretation of core concepts used throughout the thesis and refines the overall objective into success criteria that the framework and the three artifacts must fulfill.

**Chapter 3 – Research method** presents the research method used in the thesis work.

**Chapter 4 – State of the art** provides an overview of work related to the research presented in the thesis.

**Chapter 5 – Summary of contribution** presents the framework and provides an overview of the five contributions.
Chapter 6 – Overview of research papers provides an overview of the papers resulting from the research.

Chapter 7 – Discussion discusses to what extent the success criteria has been fulfilled and how our artifacts relate to and extend the state of the art.

Chapter 8 – Conclusion summarizes the work and discusses different directions for future work.
Chapter 2

Problem characterization

In Chapter 1 we presented the overall motivation and objective for our research. In this chapter we refine this objective into a set of success criteria. In Section 2.1 we clarify the interpretation of core concepts used throughout the thesis. In Section 2.2 we present success criteria that should be fulfilled in order to successfully accomplish the research objective. Section 2.2 is divided into four sub-sections. In Section 2.2.1 we present success criteria for the framework for analyzing and monitoring the impact of dependencies on quality, while in Sections 2.2.2–2.2.4 we present success criteria for the three artifacts.

2.1 Conceptual clarification

This section characterizes the main terminology used throughout the thesis.

2.1.1 System and service

As already explained in Chapter 1, the framework should be useful in the context of interconnected systems. Both computerized and non-computerized systems are addressed in this thesis. In our context, a system is characterized by a set of components which interact and operate as a whole. Based on this, we define system as “a group of interacting, interrelated, or interdependent elements forming a complex whole” [15].

In our context, systems are interconnected if they interact by the use of services. The different systems act as providers and/or consumers of services, where each service represents the exchange of some commodity (electricity, information, etc.). Moreover, we limit each service to have one provider and one consumer. Based on the above, we end up with the following definition for service: “A service is provided by a system and consumed by a system, and it represents the exchange of some commodity.”

2.1.2 Quality and quality of service

The framework has a strong focus on quality and quality of service. Generally, quality is concerned with the degree to which relevant non-functional requirements are fulfilled. In [16], quality is defined as “the degree to which a system, component, or process meets specified requirements,” while [17] defines quality as “the ability of a product, service, system, component, or process to meet customer or user needs, expectations, or requirements.” Based on the two definitions above, we define quality as “the degree to which
a system, service, component, or process meets specified non-functional requirements.” Moreover, based on this definition we define quality of service as “the degree to which a service meets specified non-functional requirements.”

2.1.3 Service dependency

As already explained in Chapter 1, the framework has been developed to analyze and monitor the impact of service dependencies on quality of services. A service may require other services in order to be provided with the required quality. A service dependency describes a relationship between a service provided by a system and services this system requires from its environment to provide the service in question. In this thesis, we consider a service to be dependent on other services if “a change in the quality of the latter may lead to a change in the quality of the former.”

2.1.4 Indicator and metric

In this thesis, the impact of service dependencies on quality of services is monitored by the use of indicators. Hammond et al. defines indicator as “something that provides a clue to a matter of larger significance or makes perceptible a trend or phenomenon that is not immediately detectable” [18]. For example, a drop in barometric pressure may signal a coming storm, while an unexpected rise in the traffic load of a web server may signal a denial of service attack in progress. Thus, the significance of an indicator extends beyond what is actually measured to a larger phenomenon of interest.

Indicators are closely related to metrics. In [17], metric is defined as “a quantitative measure of the degree to which a system, component, or process possesses a given attribute,” while it defines attribute as “the specific characteristic of the entity being measured.” For the web server mentioned above, an example of an attribute may be availability. An availability metric may again act as an indicator for denial of service attacks, if we compare the metric with a baseline or expected result [19]. As we can see, metrics are not that different from indicators. For that reason, indicators and metrics are often used interchangeably in the literature.

It should also be noticed that indicators are often referred to as key indicators in a business context. Here, the key indicators are used to measure to what extent business objectives/goals are fulfilled. In Paper A (presented in Chapter 9), we refer to indicator as key indicator, while in the rest of the thesis we only use the term indicator.

2.1.5 Trust

In the context of the thesis, trust is relevant for reasoning about third-party service dependencies. Inspired by [20,21], [22] defines trust as “the subjective probability by which an actor (the trustor) expects that another entity (the trustee) performs a given transition on which its welfare depends.” In other words, trust is the belief of a trustor that a trustee will perform a specific transaction on which the welfare of the trustor depends. For instance, the operator of a system may have a certain amount of trust in the ability of an operator of another system to deliver a service according to requirements. The level of trust may vary from 0 (complete distrust) to 1 (complete trust).
2.1.6 Architectural pattern

In this thesis, we present an architectural pattern for constructing enterprise level monitoring tools based on indicators. A pattern captures the essence of solving a recurring problem. It is a description or template for how to solve a problem that can be used in many different situations. In software engineering, patterns are best known as design patterns [23]. These patterns are used for solving recurring design problems. In [17], design pattern is defined as “a description of the problem and the essence of its solution to enable the solution to be reused in different settings.” In this thesis we focus on architectural patterns [24]. The difference between a design pattern and an architectural pattern is that design patterns describe design solutions at the object/class level, while architectural patterns describe design solutions at the architectural level, e.g., design of components and their relationships.

2.2 Success criteria

In Chapter 1 we outlined the problem area that motivates the work of this thesis, and we argued that there is a need for a framework for the analysis and monitoring of the impact of service dependencies on quality of services in the context of interconnected systems. As explained in Chapter 1, the overall objective has been to: develop a framework that is:

1. well-suited to analyze the impact of service dependencies on quality of services;

2. well-suited to support the set-up of monitoring of the impact of service dependencies on quality of services; and

3. applicable in an industrial context within acceptable effort.

The framework is the result of the development and integration of three artifacts:

1. Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

2. Method for capturing and monitoring the impact of service dependencies on the quality of provided services

3. Architectural pattern for constructing enterprise level monitoring tools based on indicators

As already explained in Chapter 1, each of the three artifacts may be viewed as contributions on their own. In Chapter 5, we relate the three artifacts to each other, and we summarize their contributions to the overall objective. The main hypothesis for each of the artifacts is that it fulfills its intended purpose (as elaborated below), and that it is feasible to be used by its intended users. As explained in Chapter 1, the artifacts target the following different types of user groups:

- The target group of Artifact 1 is the analyst and the domain experts.
- The target group of Artifact 2 is the analyst and the domain experts.
Problem characterization

• The target group of Artifact 3 is the developer of ICT-based enterprise level monitoring tools.

For the framework and each of the artifacts, we have identified a set of success criteria that the framework/artifact should fulfill. These are presented in Sections 2.2.1–2.2.4.

2.2.1 Framework for analyzing and monitoring the impact of dependencies on quality

The framework is the result of integrating the three artifacts. The purpose of the framework is to: (1) analyze the impact of service dependencies on quality of services; and (2) support the set-up of monitoring of the impact of service dependencies on quality of services. Hence, the following overall success criterion:

**Success criterion 1** *The framework fulfills its intended purpose.*

2.2.2 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

The purpose of Artifact 1 is to facilitate the design and assessment of indicators to be used in ICT-supported monitoring of the fulfillment of business objectives focusing on quality.

**Success criterion 2** *The application of the method results in indicators that measure correctly to what extent the business objectives received as input are fulfilled.*

Indicators are not suitable for measuring the fulfillment of business objectives if they cannot measure correctly to what extent the business objectives are fulfilled. The use of such indicators may lead to bad business decisions, which again can harm the company. To ensure that suitable indicators are designed, we need to evaluate whether the designed indicators measure correctly to what extent the business objectives are fulfilled.

**Success criterion 3** *The application of the method results in specifications of indicators that are well-suited for ICT-based monitoring.*

The main output from the application of the method is specifications of the indicators. These specifications are to be used to implement ICT-based monitoring of the indicators. Thus, the specifications need to be well-suited for implementing ICT-based monitoring.

**Success criterion 4** *The method is applicable in an industrial context within acceptable effort.*
In order for the method to be useful, it must be possible to apply it in an industrial context within acceptable effort.

### 2.2.3 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

The purpose of Artifact 2 is to facilitate the capture of the impact of service dependencies on risk to the quality of provided services in interconnected systems, and to facilitate the set-up of monitoring of selected risks by the use of indicators for the purpose of providing a dynamic risk picture for the provided services.

**Success criterion 5** *The application of the method results in specifications of indicators that correctly capture and measure the impact of service dependencies on the quality of provided services.*

The main output from the application of the method is specifications of indicators for monitoring risk to quality of provided services. These specifications need to correctly capture and measure the impact of service dependencies on the quality of provided services in order to be useful.

**Success criterion 6** *The application of the method results in specifications for the deployment and design of indicators that are sufficient for setting up risk monitoring based on indicators.*

In order to provide a dynamic risk picture for the provided services, the method must result in specifications for the deployment and design of indicators that are sufficient for setting up risk monitoring based on indicators.

**Success criterion 7** *The method is applicable in an industrial context within acceptable effort.*

In order for the method to be useful, it must be possible to apply it in an industrial context within acceptable effort.

### 2.2.4 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

The purpose of Artifact 3 is to serve as a basis for implementing enterprise level monitoring tools based on indicators.

**Success criterion 8** *The architectural pattern serves as a basis for building monitoring tools based on indicators within a wide range of domains and enterprises.*

It should be possible to use the architectural pattern as a basis for building monitoring tools based on indicators within a wide range of domains and enterprises. In order for this to be possible, the pattern must capture features that are general to a
Problem characterization

broad class of enterprise level monitoring tools.

**Success criterion 9** *The architectural pattern facilitates modularity and reuse.*

We have identified two desirable properties for monitoring tools resulting from the application the architectural pattern. The first desirable property is that the application of the pattern should result in tools that are modular, while the second desirable property is that it should be possible to build tools that can be reused when building other monitoring tools.
Chapter 3

Research method

In this chapter we present our research method. We start by three introductory sections. In Section 3.1 we discuss the standing of computer science as a science and describe its overall research method. In Section 3.2 we relate classical and technology research as defined by Solheim and Stølen. In Section 3.3 we describe different strategies to evaluate the research results. Thereafter, in Section 3.4 we go on to describe the research method used in this thesis. We emphasis in particular the role of empirical studies in the invention and evaluation of the three artifacts.

3.1 The standing of computer science

Computer science is a relatively young discipline compared to classical sciences such as mathematics, physics, astronomy, etc., which date back to the ancient Greeks. Moreover, computer science has a unique standing with respect to theory and practice compared to the classical sciences. While the focal point of classical sciences is more on the what than how, many advances in the history of computer science have been driven by explaining how something can be achieved through the interaction between theory and the technology that realizes it [25,26]. For this reason, [25] states that the science and engineering aspects of computer science are much closer than in many other disciplines.

With computer science being such a young discipline and closely connected to engineering, its qualification as a science has been widely debated [25,27–30]. Abelson and Sussman [30] claim that computer science is not a science and that its significance has little to do with computers. The authors do not explain what qualifies as a science or why computer science does not qualify as one. They do however relate computation to mathematics by describing mathematics as the framework for dealing precisely with notions of what is, while they describe computation as the framework for dealing precisely with notions of how to.

Similarly, Brooks [29] claims that computer science is not a science but merely an engineering discipline. According to Brooks, computer scientists are engineers since they study in order to build things such as computers, algorithms, software systems, etc. The scientist, on the other hand, is concerned with the discovery of facts and laws. The scientist does only build things that are needed for supporting his/hers studies.

Even though we find some objections to the classification of computer science as a science, there is a widely established agreement that computer science is in fact a
Research method

science, because of its many similarities with the classical sciences. Denning [27] describes computer science as the blend of mathematics, science, and engineering. Moreover, Denning criticizes the ones who object to computer science being a science on the grounds that man-made objects (technologies) are studied. According to Denning, computer science is about studying natural and artificial information processes. To study these processes, computer science relies on the same method as the classical sciences; namely the scientific method, or what we often refer to as the hypothetico-deductive method: (1) recognition of a theoretical problem; (2) proposal of conjectured solution to the problem (hypothesis); (3) critical testing of the hypothesis; and (4) conclusion: retain hypothesis if test is positive, otherwise reject hypothesis as falsified and possibly devise new tests/hypotheses/problems [31]. According to Denning, there are many examples of the usage of the scientific method within computer science. One example is that software engineering researchers hypothesize models for how programming is done and how effects arise. These models go through testing where the researchers seek to understand which models work well and how to use them to create better programs with fewer defects.

Tichy [32] shares Denning’s view on computer science being the study of information processes. According to Tichy, the applicability of the scientific method is as relevant in computer science as it is in for instance physics. It does not make any difference that the subject of inquiry is information instead of energy or matter. In both cases we would need to observe a phenomenon, formulate explanations and theories, and test them.

3.2 Classical versus technology research

Solheim and Stølen [1] distinguish between classical and technology research. For both classical and technology research we start with an overall hypothesis on the form “$B$ solves the problem $A$.” In the case of classical research, $A$ is the need for forming a new or improved theory about a real-world phenomenon, while $B$ is the new theory. The real-world phenomenon addressed may take place in nature, space, the human body, society, etc. In its simplicity, classical research is concerned with seeking new knowledge about the real world. On the other hand, in technology research $A$ is the need for a new or improved human-made object, i.e., artifact, while $B$ is this artifact. The technology researcher strives to manufacture artifacts which are better than those that already exist. The result of conducting technology research may for instance be a new or improved material, medicine, algorithm, software engineering method, etc.

Solheim and Stølen [1] argue that despite the differences in the problems addressed and the solutions sought after, classical and technology research have a lot in common and follow the same principal steps for finding a solution to the research problem. Moreover, Solheim and Stølen claim that technology research should be conducted in accordance with the hypothetico-deductive method of classical research. In technology research, the development of an artifact is motivated by a need. The overall hypothesis of technology research is that the artifact satisfies the need. A common way to evaluate the satisfaction of the overall hypothesis is to formulate a set of predictions. The falsification of a prediction results in the rejection the overall hypothesis. Hence, the predictions serve as a basis for gathering evidence on the validity of the overall hypothesis.
Technology research, like classical research, is driven by an iterative process [1]. In Figures 3.1 and 3.2 (both adopted from [1]) we have summarized the processes of technology research and classical research. The main steps of the technology research method are as follows:

1. **Problem analysis** – The researcher identifies a potential need for a new or improved artifact by interacting with potential users and other stakeholders. During this step, the researcher expresses the satisfaction of the potential need through a set of success criteria.

2. **Innovation** – The researcher tries to manufacture an artifact that satisfies the success criteria. The overall hypothesis is that the artifact satisfies the success criteria, or more precisely the potential need.

3. **Evaluation** – Based on the success criteria, the researcher formulate a set of predictions about the artifact and evaluate whether these predictions come true. If the evaluation results in a positive outcome, then the researcher may argue that the artifact satisfies the potential need.

Moreover, the main steps of the classical research method are as follows:

1. **Problem analysis** – The researcher identifies a need for a new or better theory. The need is either due to the lack of a theory or a deviation between present theory and reality.

2. **Innovation** – The researcher suggests a new explanation. The researcher works according to the overall hypothesis that the new explanation agrees with reality.

3. **Evaluation** – The researcher checks whether the hypothesis is true by performing observations. Based on the hypothesis, the researcher formulates predictions and checks whether the predictions come true. If the observations verify the predictions, the researcher can argue that the new explanation agrees with reality.

Technology research is similar to technology development. In both technology research and technology development, artifacts are developed to satisfy potential needs. What distinguishes technology research from technology development is that: (1) the
Research method

Figure 3.2: The main steps of the method for classical research (adopted from [1])

development of the artifact results in new knowledge; (2) the new knowledge is of interest to others; and (3) the development of the artifact is documented in such a way that it enables others to repeat and verify the development of the artifact [1].

Technology research is also closely related to design science. This paradigm and the behavioral-science paradigm characterize much of the research in the information systems discipline [33]. Moreover, the paradigm has its roots in engineering and the sciences of the artificial [34]. In information systems research, design science is used to create and evaluate IT artifacts intended to solve problems related to some aspect of the design of an information system. Design science differs from routine design or system development by contributing with new knowledge in the form of foundations and methodologies for design.

3.3 Strategies for evaluation

As mentioned in Section 3.2, evaluation is to find out if the predictions are true. Regardless of whether classical or technology research is being conducted, different strategies may be used to gather the evidence necessary to test the predictions. According to McGrath [35], when you gather evidence, you are always trying to maximize three things: (A) generality – results that are valid across populations; (B) precision – precise measurements; and (C) realism – the evaluation is performed in environments similar to reality. Different strategies all have their strengths and weaknesses with respect to A, B, and C. According to McGrath, the eight most common strategies are:

- **Field studies** refer to efforts to make direct observations of ongoing systems, while interfering with the systems as little as possible.

- **Laboratory experiments** are attempts from a researcher to observe systems in a context where the researcher may control and isolate the variables whose effects are to be examined.

- **Field experiments** are field studies with one major difference; the deliberate manipulation of the variables whose effects are to be examined.

- **Experimental simulations** are conducted in a laboratory setting. The researcher makes an effort to create a system that is like some class of natural occurring systems. The system is artificial in the sense that it is only created for the study.
• **Sample survey**, or just survey, are efforts to obtain information from a broad and carefully selected group of actors. The information is often given in the form of verbal responses to a set of questions.

• **Judgment studies**, or what may also be referred to as **qualitative interviews**, are efforts to obtain information from a small set of actors. The information obtained tends to be more precise than information obtained from sample surveys, but cannot be generalized in the same way.

• **Formal theory** is a theoretical approach where evidence is gathered by the use of argumentation based on logical reasoning.

• **Computer simulations** is another theoretical approach where attempts are made to model a specific real life system or class of systems.

None of the strategies is able to maximize *A*, *B*, and *C* simultaneously. Laboratory experiments score high on precision, while field studies have the greatest realism. Moreover, formal theory and sample surveys deliver the greatest generality. The solution must therefore be to choose several strategies that complement each other. When choosing strategies to evaluate the predictions, the researcher needs to consider a number of aspects [1]:

- **Is the strategy feasible?** Time, cost, and the availability of resources, such as individuals to participate in the evaluation, are important aspects to consider when selecting a strategy.

- **How to ensure that a measurement really measures the property it is supposed to measure?** The property to be measured needs to be isolated, and different factors that may influence the measurement need to be accounted for. The researcher also needs to take into account the nature of the property, i.e., whether it is qualitative, quantitative, or formal, when selecting the strategy to be used.

- **What is needed to falsify the prediction?** The strategy selected is nothing worth if it cannot cause the prediction to be rejected.

### 3.4 Our research method

The overall objective of the thesis has been to develop a framework for analyzing the impact of service dependencies on quality of services, and to support the set-up of monitoring of the impact of service dependencies on quality of services. As previously mentioned, the framework integrates three artifacts. Each of the three artifacts also serves a purpose outside the context of the framework. Thus, the artifacts have been developed in such a way that they can be employed and used in practice independently of each other. To develop the artifacts, the technology research method described in Section 3.2 was applied. The result of applying this highly iterative method was that the framework, artifacts, and success criteria were constantly improved as the evaluation provided us with new insight into the research problems.

Figure 3.3 describes the process that was followed in our thesis work. The figure also shows how this process relates to the process depicted in Figure 3.1. For each artifact, we first identified success criteria. Both the usage of the artifact in the framework and
its independent usage were taken into account when identifying the success criteria. The next step was to conduct industrial case studies. Besides providing valuable input to the development of the framework and the artifacts, the two case studies may also be seen as contributions on their own, since they provide insight into issues of a more general nature. The following industrial case studies were conducted:

- **Study 1:** Empirical study on trust-based decisions in interconnected systems
- **Study 2:** Empirical study on the design of indicators for monitoring risk

**Study 1** and **Study 2** were conducted for the purpose of identifying challenges with respect to the use of trust-based decisions in interconnected systems and design of indicators for monitoring risk, respectively. In the innovation phase, each artifact was developed with respect to the success criteria. And finally, the artifact was evaluated with respect to the success criteria.

The following three artifacts have been developed by following the process in Figure 3.3:
• **Artifact 1**: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

• **Artifact 2**: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

• **Artifact 3**: Architectural pattern for constructing enterprise level monitoring tools based on indicators

The three artifacts have been checked against the literature. To evaluate **Artifact 1** and **Artifact 2**, the following two case studies were conducted:

• **Study 3**: Case study for evaluating the method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

• **Study 4**: Case study for evaluating the method for capturing and monitoring the impact of service dependencies on the quality of provided services

Both **Study 3** and **Study 4** are the results of development of large, realistic examples for the purpose of evaluating the success criteria associated with **Artifact 1** and **Artifact 2**, respectively. Both case studies cover all the steps of the two methods from start to finish. Thus, the entire method was considered in both cases.

**Artifact 3** (the architectural pattern) is a generalization of the architecture of the CORAS risk monitor [36] that was developed in the MASTER\(^1\) [37] research project. The artifact has been evaluated based on experiences from the MASTER project, and by arguing for its ability to serve as a basis for building monitoring tools based on indicators within a wide range of domains and enterprises.

Figure 3.4 shows how the artifacts and the framework are related to the four case studies. The two industrial case studies, **Study 1** and **Study 2**, identified challenges that have been taken into consideration when identifying the challenges to be addressed by the artifacts and the framework. In particular, the main result from **Study 1** is the need for reasoning about trust in interconnected systems. This challenge has been addressed by **Artifact 2**. The main results from **Study 2** is the need for valid indicators when monitoring risk and domain knowledge for systems with dependencies. The former challenge has been addressed by **Artifact 1**, while the latter result has mainly been used in the development of **Artifact 2**. As already explained, **Study 3** and **Study 4** have been used to evaluate **Artifact 1** and **Artifact 2**, respectively. On the other hand, the development of the two case studies has also resulted in new challenges being identified. These challenges have been addressed during the development of the two artifacts.

---

\(^1\)According to [37], the MASTER (Managing Assurance Security and Trust for sERvices) project aimed at “providing methodologies and infrastructures that facilitate the monitoring, enforcement, and audit of quantifiable indicators on the security of a business process, and that provide manageable assurance of the security levels, trust levels and regulatory compliance of highly dynamic service-oriented architecture in centralized, distributed (multidomain), and outsourcing contexts.”
Research method

Motivates the need for trust

New challenges to be addressed

Solutions to challenges to be evaluated

Domain knowledge for systems with dependencies

Motivates the need for valid indicators

Figure 3.4: How the artifacts and the framework are related to the empirical studies
Chapter 4

State of the art

In this chapter we provide an overview of the state of the art of relevance to the contributions of this thesis. In the following we motivate the structure of this chapter.

The main result of Artifact 1 is a method for specifying the design and deployment of indicators to be used in ICT-supported monitoring of the fulfillment of business objectives focusing on quality. In Section 4.1.1 we present state of the art of relevance to measuring the fulfillment of business objectives. Section 4.1.2 presents state of the art related to the specification of design and deployment of indicators, while state of the art related to the validation of indicators is presented in Section 4.1.3. Moreover, state of the art of relevance to the monitoring of quality is presented in Section 4.4.1. In all four sections we put particular emphasis on the issue of quality.

The main result of Artifact 2 is a method for capturing the impact of service dependencies on risk to quality of provided services in interconnected systems. This includes facilitating monitoring of selected risks by the use of indicators to offer a dynamic risk picture for the provided services. The method relies on modeling and analysis of service dependencies in order to capture their impact on risk. In Sections 4.2.1 and 4.2.2 we provide an overview of state of the art related to the modeling and analysis of dependencies in general and service dependencies in particular, respectively. The method also provides a solution for estimating the trustworthiness of services provided by third-parties. Related state of the art is presented in Section 4.5.1. To capture the impact of service dependencies on risk to quality, the method employs an asset-based risk analysis approach. In Section 4.3.1 we present approaches of this kind, while in Section 4.3.2 we direct our attention to state of the art of relevance to the analysis of risk in the context of dependencies in general and service dependencies in particular. As already explained, the method is also used for setting up monitoring of selected risks to quality. Thus, in Section 4.4.2 we discuss state of the art of relevance to dynamic risk monitoring.

The main result of Artifact 3 is an architectural pattern that serves as a basis for implementing enterprise level monitoring tools based on indicators. In Section 4.4.3 we present state of the art related to patterns and the implementation of monitoring tools.

The first industrial case study investigates the use of trust to reason about the behavior of systems/actors in cases where the behavior of these systems/actors may result in risks and/or opportunities. In Section 4.5.2 we discuss state of the art related to trust-based decisions in interconnected systems.

The second industrial case study investigates the use of indicators to validate likeli-
hood estimates based on expert judgments in security risk analysis. In Section 4.3.3 we present state of the art related to the validation of expert judgments used in risk analysis, while in Section 4.3.4 we focus on state of the art related to the use of indicators in risk analysis.

4.1 Indicators

4.1.1 Measurement of the fulfillment of business objectives

There exist numerous approaches for measuring business performance. Some of these are presented in [38]. Regardless of the approach being used, the organization must translate their business objectives/goals into a set of key performance indicators in order to measure performance. An approach that is widely used [39] is balanced scorecard [40]. This approach translates the company’s vision into four financial and non-financial perspectives. For each perspective, a set of business objectives (strategic goals) and their corresponding key performance indicators are identified. However, the implementation of a balanced scorecard is not necessarily straight forward. In [41], Neely and Bourne identify several reasons for the failure of measurement initiatives such as balanced scorecards. One problem is that the identified measures do not measure fulfillment of the business objectives, while another problem is that measures are identified without putting much thought into how the data must be extracted in order to compute the measures.

There exist a number of frameworks and best practice approaches that are supported by metrics/indicators for measuring the achievement of goals. One example is COBIT (Control Objectives for Information and related Technology) [42], which is a framework for IT management and IT governance. The framework provides an IT governance model that helps in delivering value from IT and understanding and managing the risks associated with IT. In the governance model, business goals are aligned with IT goals, while metrics, in the form of leading and lagging indicators [43], and maturity models are used to measure the achievement of the IT goals. Another example is Val IT [44], which is a framework that is closely aligned with and that complements COBIT. It consists of a set of guiding principles and a number of processes and best practices for measuring, monitoring, and optimizing the realization of business value from investment in IT.

ITIL [45–49] (Information Technology Infrastructure Library) provides best practice guidance for IT service management for the purpose of aligning IT services with business needs. The current version of ITIL (version 3) provides a holistic perspective of the full life cycle of services by covering the entire IT organization and all supporting components needed to deliver services to customers. Under ITIL guidelines, services are designed to be measurable. To support measurement, ITIL comes with a number of metrics/indicators.

4.1.2 Specification of the design and deployment of indicators

A number of standards and guides provide guidance on the design of indicators/metrics. The ISO/IEC 27004 [50] standard provides guidance on the development and use of measures and measurement for assessing information security management systems and controls, as specified in ISO/IEC 27001 [51]. The appendix of the standard also
suggests security metrics which have been selected to align with ISO/IEC 27002 [52]. Moreover, the NIST Performance Measurement Guide for Information Security [53] provides guidance on the development, selection, and implementation of suitable measures for information security. It also comes with a number of candidate measures for measuring information security.

The Goal-Question-Metric [54,55] (GQM) is an approach for measuring the achievement of goals. Even though GQM originated as an approach for measuring achievement in software development, it can also be used in other contexts where the purpose is to measure achievement of goals. In GQM, business goals are used to drive the identification of measurement goals. These goals do not necessarily measure the fulfillment of the business goals, but they should always measure something that is of interest to the business. Each measurement goal is refined into questions, while metrics are defined for answering the questions. The data provided by the metrics are interpreted and analyzed with respect to the measurement goal in order to conclude whether it is achieved or not.

An approach that is closely related to GQM is the Goal-Question-Indicator-Measurement [56,57] (GQ(I)M) approach. In this approach, we start by identifying business goals that we break down to manageable sub-goals. The approach ends with a plan for implementing measures and indicators that support the goals.

In [58], Popova and Sharpanskykh present a framework for modeling performance indicators within a general organization modeling framework. Two of the main contributions of this framework are the formalization of the concept of a performance indicator, and the formalization of the relationships between performance indicators. In [59], the same authors present a formal framework for modeling goals based on performance indicators. This formal framework is also used within the general organization modeling framework mentioned above. To enable evaluation of organizational performance, the framework defines mechanisms for establishing goal satisfaction. By using the frameworks presented in [58,59], goal and performance indicator structures can be modeled. The goal structure is given in the form of a hierarchy. It shows how different goals are related and how goals are refined into sub-goals. It also defines how goal satisfaction should be propagated through the hierarchy. On the other hand, the performance indicator structure shows the different relationships that exist between the performance indicators. It may for instance show that one indicator is the aggregation of other indicators, or that the value of one indicator influences the value of another. In order to identify goals and performance indicators and to create the two structures, both frameworks rely on organizational documents and expert knowledge. The frameworks also provide mechanisms for checking consistency of and correspondence between the goal and the performance indicator structures.

### 4.1.3 Validation of indicators

No specific method, beyond reviews, is specified for validating whether the correct questions/sub-goals and metrics/indicators have been identified by the use of GQM or GQ(I)M. In the case of GQM, the different kinds of data used for computing the metrics should be checked for correctness, completeness, and consistency [55].

In the software engineering literature, the validity of metrics that measure attributes of software products is an important topic (see e.g., [60–62]). Even though this literature targets software engineering, it is still relevant to different extents in other domains.
that also focus on the validity of measurements/metrics/indicators. Even though the validity of software engineering metrics have received a lot of attention, no agreement have been reached upon what constitutes a valid metric [63]. In [63], Meneely et al. present a systematic literature review of papers focusing on validation of software engineering metrics. The literature review began with 2288 papers, which were later reduced to 20 papers. From these 20 papers, the authors extracted and categorized 47 unique validation criteria. The authors argue that metric researchers and developers should select criteria based on the intended usage of the metric when validating it.

A number of the software metrics validation approaches advocate the use of measurement theory [64–66] in the validation (see e.g., [67–69]). Measurement theory is a branch of applied mathematics that is useful in measurement and data analysis. The fundamental idea of this theory is that there is a difference between measurements and the attribute being measured. Thus, in order to draw conclusions about the attribute, there is a need to understand the nature of the correspondence between the attribute and the measurements. In [70], Morali and Wieringa present an approach that relies on measurement theory for the validation of indicators. More specifically, the approach uses measurement theory to validate the meaningfulness of IT security risk indicators.

Measurement theory has been criticized of being too rigid and restrictive in a practical measurement setting. Briand et al. [68] advocate a pragmatic approach to measurement theory in software engineering. The authors show that even if their approach may lead to violations of the strict prescriptions and proscriptions of measurement theory, the consequences are small compared to the benefits. Another approach that takes a pragmatic approach to measurement theory is [69]. Here, the authors propose a framework for evaluating software metrics. The applicability of the framework is demonstrated by applying it on a bug count metric.

4.2 Dependencies

4.2.1 Modeling and analysis of dependencies in general

Dependencies are often modeled by the use of directed graphs. In these graphs, vertices represent different entities, while edges represent dependencies between the entities. The direction of an edge specifies the direction of the dependency. There exist a number of graph-based approaches that can be used in the modeling of dependencies. One example is semantic networks [71], which is a technique for knowledge representation. The main idea behind semantic networks is that knowledge is often best understood as a set of concepts that are related to each other. A semantic network is given in the form of a graph, where vertices represent concepts, while directed labeled edges connect the concepts. The meaning of a concept is defined by its semantic relations to other concepts. With a semantic network we can show which concepts that depend on each other and why.

In [72], Cox et al. present an approach to dependency analysis that relies on conceptual graphs [73], which is a formalism for knowledge representation. In [72], the conceptual graphs are used to represent, characterize, and analyze dependencies between the entities of a model. Cox et al. claim that the use of conceptual graphs makes it easier to model dependencies at multiple levels of detail and when only partial information is available.
Another approach to dependency modeling and analysis is block diagrams [74,75]. These diagrams are often used in reliability assessments. The diagrams show how components of a system are parallel or serial, and thereby identify possible weak points in the form of dependencies.

### 4.2.2 Modeling and analysis of service dependencies

There exist a number of approaches that focus on the modeling and analysis of service dependencies and their applications in different domains. In [76], Ensel and Keller present an XML-based model for specifying service dependencies in distributed systems. The purpose of the model is to facilitate information sharing between the different systems. In [77], a UML-based service dependency model for ad hoc collaborative systems is presented. In such systems, different computing devices participate in collaborative applications by using and offering services to each other. Besides handling static aspects of service dependencies, the modeling approach also handles dynamic aspects of service dependencies due to the possibility of devices coming and going. Another approach that also takes dynamic aspects of service dependencies into account is presented in [78]. This approach provides a service dependency classification for system management analysis. It traces the flow of dependency information from the design to the run time stages of services. It relies on two types of models: a function model that defines generic service dependencies, and a structural model containing detailed information on the software components realizing the services.

In [79], Ding and Sha present a dependency algebra, consisting of a formal theoretical framework and a prototype toolkit, for service dependency management in real-time systems. The algebra can be used to ensure that critical components only use and do not depend on services provided by non-critical components. By ensuring that critical services do not depend on non-critical services, the system will not be brought down by minor failures in non-critical services. The algebra also offers a metric for measuring the strength of dependencies.

There are also approaches that address service dependencies with respect to security. Such approaches take into account that the dependencies have security implications, i.e., that a change in the security state of a component providing a service implies a change in the security state of the component requiring the service. One such approach is [80]. This approach is used for constructing formal models of services dependencies in information systems. The constructed dependency models are to be used in security policy-based management. More precisely, the dependency models are used to find enforcement points for security rules, which then support countermeasure deployment, and for computing the impact of attacks and countermeasures that propagate over the information system. A related approach is [81], which presents a framework for risk assessment of information infrastructures. The framework generalizes the notion of dependency with respect to security attributes, such as confidentiality, integrity, and availability. These security dependencies are used to describe relationships between components, to discover attack strategies, and to define risk mitigation plans.

Service dependencies are also used in fault analysis [82] and dependability analysis [83], as well as in analyses targeting critical infrastructures. A number of the approaches that address critical infrastructures use graph-based models to model and analyze service dependencies (see e.g., [84–87]). Moreover, a number of the approaches focus primarily on the consequences of infrastructure services not being provided. One
such approach is [86]. This approach is used to create models of infrastructure systems and their interactions. The models are used in computer simulations where the main purpose is to investigate how the functionality of infrastructure systems and interconnections react to different attack scenarios (“what if” scenarios where one or two systems are removed), and how mechanisms for strengthening the underlying dependency graph can be used.

4.3 Risk analysis

4.3.1 Asset-based risk analysis methods

According to [88], risk analysis is the process to comprehend the nature of risk and to determine the level of risk, whereby a risk is the effect of uncertainty on objectives. For an organization, this effect of uncertainty on objectives may be the result of internal and/or external factors, while the effect may be either positive or negative, since uncertainty is a deviation from the expected.

An asset is something to which a party assigns value and hence for which the party requires protection. Asset-based risk analysis methods focus on identifying the assets that require protection, and then assess how they may be protected from threats and risks. There exist a number of well-established methods for asset-based risk analysis. Examples include: Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [9]; CCTA Risk Analysis and Management Method (CRAMM) [10]; (Microsoft) Threat Modeling [89]; and CORAS [11]. In the following we focus on CORAS, since it is the method most relevant for Artifacts 1 and 2.

The CORAS approach for model-based risk analysis consists of the CORAS language for risk modeling; the CORAS method which is a step-by-step description of the risk analysis process, and that comes with a detailed guideline for constructing CORAS diagrams; and the CORAS tool which is used for documenting, maintaining, and reporting risk analysis results in the form of CORAS diagrams.

The CORAS method is structured into eight steps: (1) preparation for the analysis; (2) customer presentation of target; (3) refining the target description using asset diagrams; (4) approval of target description; (5) risk identification using threat diagrams; (6) risk estimation using threat diagrams; (7) risk evaluation using risk diagrams; and (8) risk treatment using treatment diagrams.

The CORAS approach is based on the ISO 31000 standard [88] which is preceded by the AS/NZS 4360 standard [90]. This results in the CORAS method being very similar to other methods that are based on these or similar standards. What really distinguishes the CORAS method from many other methods is its strong focus on the modeling of the target and the risks, for the purpose of supporting communication, documentation, and analysis of the risk analysis results.

4.3.2 Analysis of risk in the context of dependencies

Markov analysis [91] is a stochastic mathematical analysis method that is well-suited for assessing the reliability of systems with component dependencies. In Markov analysis, the system is considered as a number of states, from perfect operation to no operation at all. A Markov model is used to describe the states and the state transitions. The states and the transitions are modeled graphically, and statistical calculations are
4.3 Risk analysis

used for determining the likelihoods of the different state transitions. The most important weakness of Markov analysis is the high workload and complexity resulting from analyzing large systems.

Dependent CORAS [11] is an approach for modular risk modeling, which can be used to document and reason about risk in the context of dependencies. It extends the CORAS risk modeling language with facilities for documenting and reasoning about risk analysis assumptions. It was motivated by the need to deal with mutual dependencies in risk analysis of systems of systems. By employing dependent CORAS we may document risk separately for the individual systems. In addition, we document the risk analysis assumptions for the different systems. A risk analysis assumption documents how a threat scenario or an unwanted incident of one system may lead to a threat scenario or an unwanted incident of another system. These assumptions are due to some form of dependencies, not necessarily service dependencies, between the different systems. The different risk models may be combined in the end, if the dependencies between them are well-founded, i.e., not circular.

There exist several approaches from the safety domain that apply component-based hazard analysis to describe fault propagation in systems containing dependent components. Giese et al. [92, 93] present a method for compositional hazard analysis of components described in the form of restricted UML [94] component and deployment diagrams. This method applies Fault Tree Analysis [95] (FTA) to describe hazards and the combination of components that causes them. For each component, incoming, outgoing, and internal failures are described, as well as the dependencies between the different failures. The dependency information is used to capture the propagation of failures by combining failure information of the different components. It should be noticed that the method of Giese et al. only considers failures caused by software and/or hardware. Thus, human failures, either accidental or deliberate, are not considered.

In [96], another approach to component-based hazard analysis is described. The approach extends, automates, and integrates well-established risk analysis techniques such as Functional Failure Analysis [97] (FFA), Failure Mode Effect Analysis [98] (FMEA), and FTA. A specialized version of FMEA is applied to describe component output failures of the individual components. The causes of the output failures are described as a logical combination of internal malfunctions of the component or deviations of the component’s inputs. Based on the results from the FMEA analyses, fault trees for the components are constructed. The approach synthesizes the individual fault trees in order to describe the fault propagation in the system.

Kaiser et al. [99] present a component concept for FTA. They divide a fault tree into so-called fault tree components. Each fault tree component has incoming and outgoing ports. These ports are used to connect the different fault tree components into a system fault tree. A major advantage of the approach is the ability to reuse fault tree components.

4.3.3 Validation of expert judgments in risk analysis

In [100], Otway and von Winterfeldt describe two different types of processes for making expert judgments in risk analyses; informal and formal processes. Informal processes are implicit, unstructured, and undocumented, while the formal processes are explicit, structured, and documented. In general, formal processes focus more on detecting and resolving biases than informal processes. On the other hand, formal processes are often
time- and cost-consuming, have lack of flexibility, and may result in possible loss of creativity due to the formalism.

Many approaches for the elicitation of expert judgments focus on the aggregation of expert judgments in order to achieve expert judgments of good quality. There are two classes of aggregation methods: behavioral and mathematical. Behavioral approaches (see e.g., [101–103]) focus on negotiation in order to achieve a consensus, while mathematical approaches (see e.g., [104]) are rule or formula based.

4.3.4 Indicator-based risk analysis

In [105], Refsdal and Stølen present an approach to risk monitoring where risk values are calculated from measurable indicators. The approach consists of three main steps. In the first step, a risk analysis of the system is performed, and risks to be monitored are identified. The risk analysis provides information about how threats may exploit vulnerabilities to initiate threat scenarios leading to the risks. In the second step, relevant measurable indicators are identified for risks or vulnerabilities, threat scenarios, etc., leading up to the risks. Functions for calculating likelihood, consequence, and risk values based on indicators are identified in the third step. In [105], the authors also provide guidelines for how to evaluate the internal consistency of the dynamic risk picture. The authors also present a view on how to measure confidence in the dynamic risk picture, based on the discovered internal inconsistencies.

In [106], Baker et al. propose an approach which uses measurable, real-world metrics to improve information security risk assessment and decision making. The main motivation behind the approach is the inability of business leaders to identify the most effective information security strategies for limiting organizational loss. In particular, the approach aims to do the following three things: allow accurate measurement and tracking of threats; enable determination of the impact of loss of successful threats; and aid in evaluating the effectiveness and return on investment of countermeasures.

Breier and Hudec [107] present an approach which uses metrics as an instrument for security risk assessment. In the paper, metrics are used for evaluating the fulfillment of security control objectives. The authors propose a mathematical model based on metrics for evaluating the security control objectives.

4.4 Monitoring

4.4.1 Monitoring of quality

Quality monitoring is often concerned with the monitoring of quality of service (QoS). A large number of approaches focus on the monitoring of service level agreement (SLA) fulfillment. SLAs are used to establish a contract between service providers and consumers concerning quality of service parameters. One such approach is presented in [108]. The paper addresses the problem of service providers that deviate from the SLAs when providing web services. QoS monitoring is necessary for measuring the fulfillment of the SLAs. The problem is that neither the service provider nor the consumer can be trusted when it comes to monitoring. The paper presents QoS monitoring mechanisms that are based on feedback from the consumers. The consumers are running the monitoring code, and they report periodically feedback to a trusted reputation mechanism (RM), which estimates the delivered QoS for the different providers based
on the reports. Providers that do not fulfill their SLAs are penalized by the RM. The RM also applies incentives for the consumers to report honestly.

In [109], Wang et al. present a QoS management framework and QoS management services, including monitoring and diagnostics, for service level management in networked enterprise systems. The monitoring service does not only monitor the fulfillment of SLAs, it also monitors the health of the systems responsible for providing the services. The diagnostics service performs analyses and QoS related diagnostics based on data provided by the monitoring service. If SLA violations or system degradations are detected by the diagnostics service, adaption mechanisms are activated in order to maximize the systems’ ability to meet QoS parameters specified in the SLAs.

Monitoring is also employed within business intelligence, often for the purpose of measuring achievement of business objectives. Data [110] and process [111] mining tools are two types of tools that do some sort of monitoring. Within business intelligence, data mining uses techniques from statistics and artificial intelligence to identify interesting patterns in often large sets of business data, while process mining is used to extract information about business processes by the use of event logs. Another type of tools that rely on monitoring is business performance management [112] tools. These tools are used to monitor, control, and manage the implementation of business strategies.

### 4.4.2 Dynamic risk monitoring

In [113], Trad et al. present a distributed system monitoring application called Factors Estimation System (FES), which is an application that can be used for proactive monitoring of information system risk and quality. By monitoring different sources to problems, the application can detect problems before they become critical. When problems are detected, reports and alarms are generated. Another tool for monitoring, called MASTER ESB, is presented in [114]. This tool is used to monitor compliance with access and usage policies in a system. The tool monitors low-level evidence data that is aggregated into meaningful evidence on how different parties comply with the policies. This evidence is then evaluated, and actions against compliance violations may be taken.

NIST [115] provides a guideline for information security continuous monitoring (ICSM). NIST defines ICSM as “maintaining ongoing awareness of information security, vulnerabilities, and threats to support organizational risk management decisions.” In this context, ongoing means that “security controls and organizational risks are assessed and analyzed at a frequency sufficient to support risk-based security decisions to adequately protect organizational information.” The purpose of the guideline is to assist organizations in the development of an ICSM strategy and the implementation of an ICSM program. The guideline describes the fundamentals of ongoing monitoring of information security in support of risk management, and it describes the process of ICSM, including implementation guidelines.

Risk monitoring is also central in approaches that focus on the protection of critical infrastructures (see e.g., [116,117]), as well as in approaches that focus on the protection of computerized networks (see e.g., [118,119]).
4.4.3 Patterns and the implementation of monitoring tools

The two main categories of patterns within software engineering are design patterns [23] and architectural patterns [24]. As already mentioned in Section 2.1.6, the difference between a design pattern and an architectural pattern is that design patterns describe design solutions at the object/class level, while architectural patterns describe design solutions at the architectural level, e.g., design of components and their relationships. There exist a number of different templates for describing patterns (see e.g., [23, 24]). It may of course be discussed what classifies as a good description of a pattern, but in general the description must capture the essence of solving the recurring problem in such a way that the pattern is easy to learn, compare, and use [23, 24].

In this thesis we focus on architectural patterns. The Model-View-Controller (MVC) pattern [24, 120] is one of the best-known examples of architectural patterns. MVC divides an interactive application into three main components: model, view, and controller. The model encapsulates core data and functionality, while views and controllers together comprise the user interface of the application. Each view displays data obtained from the model in a specific way, while the controllers are used to handle user input. The MVC pattern makes it easy to change the user interface of an interactive application, since the model is independent of the user interface.

Design patterns that specifically target the building of software health monitoring applications are described in [121]. The paper focuses on design patterns for sensors collecting information about the internal state and operation of software, and how this information can be combined into software health indicators describing different aspects of software health.

The tool framework called Mozart [122] uses a model driven approach to create monitoring applications that uses key performance indicators (KPIs). The framework mines KPIs from a data warehouse and builds an initial KPI net. In this net there will be KPIs that are central for reaching a goal. These KPIs are identified by the use of a goal model. There will also be other KPIs that can directly or indirectly influence the KPIs that are central for reaching the goal. In the next step, the framework makes use of the goal model and historical data on the different KPIs to discover how the different KPIs correlate with each other. This step results in a new KPI net. The new net contains dependencies that specify the correlations between the different KPIs. The next step is then to discover which of the dependency chains in the KPI net that are most influential for monitoring the achievement of the goal. After having identified these chains, a monitor model is constructed. This model can be transformed into a monitor application.

In [123], the design and implementation of a performance monitoring tool for clustered streaming media server systems is presented. The tool focuses on monitoring resources such as CPU utilization, memory usage, disk usage, and network bandwidth. In addition to the tool presented in [123], there are also commercial monitoring solutions that focus on similar monitoring tasks. Examples include SolarWinds ipMonitor [124] (for monitoring network devices, servers, and applications) and the IBM Tivoli Monitoring software [125] (for monitoring operating systems, databases, and servers in distributed and host environments).
4.5 Trust

4.5.1 Estimating the trustworthiness of external services

In [117,126], the challenge of security risk assessment in interdependent critical infrastructures is addressed. To assess risk, a critical infrastructure operator needs information about services provided by its own infrastructure, as well as information about services provided by infrastructures that its own infrastructure depends on. Thus, in the approach of [117,126], risk information is shared between the different interdependent critical infrastructures. The problem is that information provided by one critical infrastructure may be inaccurate. Such information will again affect the correctness of the risk assessment results. To tackle this problem, [117,126] use trust indicators to classify how accurate the exchanged information is. Information that is not trusted may be given a low weight during the risk assessment or it may be discarded.

Subjective logic [127,128] is another approach that can be used to deal with uncertainty. It is a probabilistic logic that captures uncertainty about probability values explicitly. The logic operates on subjective belief about the world. Different actors have different subjective beliefs, and these beliefs are associated with uncertainty. The approach makes it possible, for example, to calculate to what degree an actor believes that a service will be provided based on the actor’s beliefs about services that the service in question depends on, or to calculate the consensus opinion of a group of actors. Subjective logic deals strictly with the actors’ beliefs and reasoning, and does not address the question of how their beliefs affect their behavior. The belief calculus of subjective logic can be applied in risk analysis to capture the uncertainty associated with such analysis, as shown in [129]. This is achieved by using subjective beliefs about threats and vulnerabilities as input parameters to the analysis. Through application of the belief calculus, the computed risk assessments provide information about the uncertainty associated with the result of the analysis.

A Bayesian network [130] is a directed acyclic graph consisting of nodes with states and edges describing causal relationships between nodes. Each node is characterized by a probability distribution over its possible states, where these probabilities depend on the probabilities of the states of its parents. The probabilities of nodes are changed by the gathering of new evidence. For any changes of the probabilities of nodes, the effects both forwards (towards child nodes) and backwards (towards parent nodes) may be computed.

In [131], Wu et al. present a Bayesian network based QoS assessment model for web services. The purpose of the model is to predict whether different service providers can deliver a service that satisfies service consumers’ QoS requirements. The model is trained by computing the compliance between consumers’ QoS requirements and the QoS of the delivered service. The capability of the service to deliver the correct QoS is inferred based on the compliance values, and the Bayesian network is updated by using the inference outcome. In [132], Melaye and Demazeau propose a Bayesian dynamic trust model for determining an agent’s trust in another agent. The model can for instance be used to determine whether an agent delivers a service with the expected quality. The notion of trust is formalized by using a Bayesian network that is structured into three layers. The top layer is the trust level, while the second and third level represent basic beliefs and belief sources, respectively. A Bayesian Kalmar filter is used to capture dynamic aspects of trust, e.g., changes in beliefs. In [133], a Bayesian
network trust model for peer-to-peer networks is proposed. A peer can use trust to assess other peers’ ability to provide services with the expected quality. In [133], a peer calculates trust by the use of Bayesian networks. The peer maintains a Bayesian network for each peer that it has received services from. After each service interaction, the peer updates the Bayesian network based on the quality of the provided service. The trust values calculated by the Bayesian networks are used to rank the different service providers.

Fuzzy Logic [134] is a form of reasoning for computers that is very close to human reasoning. It is used to draw conclusions from uncertain, vague, ambiguous, or imprecise information. Fuzzy logic is therefore suitable for reasoning about trust, since trust assessments are often based on such information. In Fuzzy logic, we first perform a fuzzification by gathering crisp numerical values that are assigned to fuzzy sets by the use of fuzzy linguistic variables, fuzzy linguistic terms, and membership functions. A crisp numerical value can be the member of more than one fuzzy set. The degree of membership in a set ranges between 0 and 1 in fuzzy logic. Thus, a crisp value that represent some assessment of service quality can have a membership degree of 0.1 in the set “low” (quality) and a membership degree of 0.8 in the set “medium” (quality). After the fuzzification has been conducted, the fuzzy values are used to evaluate a set of IF ... THEN ... rules. A rule can for instance say: IF quality is low THEN trust is low. Afterwards, the results from the evaluation of the rules are aggregated. The last step is to perform a defuzzification by mapping the aggregation results to crisp numerical values. Such a crisp numerical value can for instance specify the trust in a service provider’s ability to provide a service with the required quality.

In [135], Griffiths et al. present a trust model based on fuzzy logic for peer-to-peer systems. The trust model uses fuzzy logic to represent and reason with imprecise and uncertain information regarding peers’ trustworthiness. A peer can use the model to select the most appropriate service providers among the other peers. More precisely, the model enables a peer to maximize the quality of the services that it requires according to its current preferences.

4.5.2 Trust-based decisions in interconnected systems

Reputation systems [136] are often used to decide whom to trust on the Internet. Such systems collect, distribute, and aggregates feedback about participants’ past behavior. In [137], Resnick and Zeckhauser present an empirical analysis of eBay’s reputation system. The analysis was based on a large data set from 1999 provided by eBay. The analysis resulted in discoveries such as: feedback was provided more than half the time; the feedback was almost always positive; and sellers’ feedback profiles were predictive of future performance. The authors found the low rate of negative feedback highly suspicious. Despite this, the authors come to the conclusion that the system appears to be working. One of the explanations that the authors consider is that the system may still work, even if it is unreliable and unsound, if its participants think it is working. Thus, if sellers believe that a strong reputation is needed in order to sell goods, then they will behave in ways that result in positive feedback.

In [138], Lim et al. investigate the effectiveness of two trust-building strategies to influence actual buying behavior in online shopping environments, particularly for first-time visitors to an Internet store without an established reputation. The two strategies investigated were portal association (e.g., the store is associated with Yahoo,
Amazon, etc.) and satisfied customer endorsements. Two studies were conducted at a large public university in Hong Kong with students as test subjects. Of the two strategies investigated, satisfied customer endorsements by similar (local, non-foreign) peers was found to increase the test subjects' trust in the online store investigated in the two studies. Portal association did not lead to an increase in trust.

In addition to the two studies presented above, a number of other surveys, reviews, and empirical studies on trust and reputation approaches have been conducted (see e.g., [139–142]).
Chapter 5

Summary of contribution

This thesis makes two kinds of contributions. Firstly, it contributes in terms of new artifacts. Secondly, it contributes in terms of industrial case studies. The industrial case studies were mainly carried out to support the development of the artifacts, but since the industrial case studies also provide insight into issues of a more general nature, they may be seen as contributions on their own. The three artifacts are

1. **Artifact 1**: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

2. **Artifact 2**: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

3. **Artifact 3**: Architectural pattern for constructing enterprise level monitoring tools based on indicators

The two industrial case studies are

1. Empirical study on trust-based decisions in interconnected systems

2. Empirical study on the design of indicators for monitoring risk

Although the three new artifacts may be viewed as a contribution on their own, and also be employed and used in practice independently of each other, they are also closely related and may be integrated. As indicated by Figure 5.1, we refer to the result of this integration (which is represented by the gray circle) as the **ValidKI Framework**, while we refer to the application of Artifacts 1, 2, and 3 within the framework as **Indicator Design**, **Dependency Analysis**, and **Monitoring Pattern**, respectively.

The gray circle does only cover parts of the three artifacts, since the artifacts have applications that go beyond the framework. In the framework, Artifact 1 is applied to business objectives focusing on service quality. The artifact is however not limited to this kind of business objectives only. The artifact can be used to design indicators to monitor the fulfillment of all sorts of business objectives focusing on quality, as long as ICT-supported monitoring of the indicators is possible.

In the framework, Artifact 2 is used to capture the impact of service dependencies on risk to the fulfillment of business objectives with respect to service quality. In this context, all unacceptable risks to the fulfillment of a business objective need to be selected for monitoring. When applied independently of the framework, the focus is
Figure 5.1: Relations between the ValidKI Framework and Artifacts 1–3

still on service quality, but not on business objectives focusing on the achievement of service quality. Then it is up to the client on whose behalf the artifact is applied to select the risks to be monitored.

In the framework, Artifact 3 is used to implement a risk cockpit based on indicators that facilitates the monitoring of business objectives focusing on the achievement of service quality. Artifact 3 is however not limited to the implementation of risk cockpits only. It may be used to implement all sorts of enterprise level monitoring tools based on indicators.

The remainder of this chapter consists of three sections. In Section 5.1 we present the ValidKI Framework. In Section 5.2 we provide an overview of the three contributed artifacts. Finally, Section 5.3 is devoted to the two industrial case studies.

### 5.1 The ValidKI Framework

In Figure 5.2, the ValidKI Framework is described. The figure shows in particular the integration of Indicator Design and Dependency Analysis corresponding to Artifacts 1 and 2, respectively. Arrows have been used to show in which order the different steps of the two methods are executed. Moreover, bold arrows are used to indicate inputs to and outputs from the artifacts. It should also be noticed that the grayed steps of Indicator Design method are not executed. These steps are replaced by steps of the Dependency Analysis method. In the following we explain the integration of the three artifacts. We use Step X (ID) to refer to Step X of Indicator Design, while we use Step Y (DA) to refer to Step Y of Dependency Analysis.

The input to the framework is a business objective focusing on the achievement of service quality. In Step 1.1 (ID) the business objective is expressed more precisely in order to understand exactly what it means to fulfill it. Step 1.2 (ID) is used to describe the part of the business that needs to reach the business objective and therefore is to be monitored. With this relevant part of business being interconnected systems that
5.1 The ValidKI Framework

A business objective focusing on the achievement of service quality

Artifact 1: Indicator Design

Step 1 Establish target
1.1 Express business objective more precisely
1.2 Describe relevant part of business

Step 2 Identify risks to fulfillment of business objective
2.1 Specify risk acceptance criteria
2.2 Risk identification and estimation
2.3 Risk evaluation

Step 3 Identify key indicators to monitor risks
3.1 Deploy sensors to monitor risks
3.2 Specify requirements to key indicators wrt deployed sensors

Step 4 Evaluate internal validity
4.1 Express business objective in terms of key indicators
4.2 Evaluate criteria for internal validity

Step 5 Specify key indicator designs

Step 6 Evaluate construct validity

Artifact 2: Dependency Analysis

Step 1 Document interconnected systems
1.1 Model interconnected systems
1.2 Capture service dependencies
1.3 Capture trust relations

Step 2 Analyze the impact of service dependencies on risk to quality of provided services
2.1 Identify quality assets
2.2 Construct high-level threat diagrams of the impact of service dependencies on identified quality assets
2.3 Construct detailed threat diagrams of the impact of service dependencies on identified quality assets

Step 3 Identify indicators for interconnected systems
3.1 Identify risks to be monitored
3.2 Identify relevant indicators for the risks to be monitored

Step 4 Specify design and deployment of identified indicators for interconnected systems
4.1 Specify design of indicators for risk monitoring
4.2 Specify deployment of indicators for risk monitoring

A set of indicators that is valid with respect to the relevant service quality

Artifact 3: Monitoring Pattern

Figure 5.2: The ValidKI Framework
depend on each other through service interactions, we replace Step 1.2 (ID) with Step 1 (DA) in order to document the interconnected systems, their services, service dependencies, quality requirements to services, and trust in services provided by systems of which we have insufficient documentation. Moreover, since we need to capture the impact of service dependencies on risk to the fulfillment of the business objective, Step 2.1 (ID) and Step 2.2 (ID) are replaced by Step 2 (DA).

The result of conducting Step 2 (DA) is a risk model capturing the impact of service dependencies on the fulfillment of the business objective. In Step 3.1 (DA), which replaces Step 2.3 (ID), we identify the risks that should be monitored by the use of indicators. All risks that are unacceptable with respect to the fulfillment of the business objective are identified for monitoring. In Step 3.2 (DA) we identify indicators for monitoring the risks, while in Step 3 (ID) we specify deployments of sensors in the interconnected systems, and we specify requirements to the indicators with respect to the sensor deployments. The sensors are needed for gathering the data necessary for calculating the indicators. After having evaluated the internal validity of the indicators in Step 4 (ID), the design and deployment of indicators are specified in Step 4 (DA). The designs specify in the form of algorithms how indicators should be calculated, while the deployments specify how data needed in the calculations should be extracted and transmitted within the relevant part of business. In Step 5 (ID), the specifications from Step 4 (DA) are refined. The result of Step 5 (ID) is specifications that describe how different sensors, actors, and components/systems need to interact in order to calculate the indicators. The final step is to evaluate the construct validity of the indicators. A risk analysis is conducted as part of Step 6 (ID). To analyze the risks we rely on Step 2 (DA) due to the dependencies between the interconnected systems.

The output from the two integrated artifacts is a set of indicators for monitoring the fulfillment of the business objective received as input. The output is used as input to Monitoring Pattern corresponding to Artifact 3. Based on the input, a risk cockpit that facilitates monitoring of the business objective can be constructed.

5.2 Overview of artifacts

5.2.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

The methodology is used for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. For a detailed description of the methodology, we refer to Chapter 9.

The focus of the methodology is on designing valid indicators. A set of indicators is valid with respect to a business objective if it measures the degree to which the business or relevant part thereof fulfills the business objective. Six main steps are conducted in order to design valid indicators for a business objective. The first main step is all about understanding what needs to be monitored. More precisely, we need to express the business objective in a precise manner, and we need to provide a description of the part of the business that needs to reach the business objective. The second main step is concerned with conducting a risk analysis to identify risks to the fulfillment of the
business objective. We distinguish between three sub-steps. Risk acceptance criteria are specified in the first sub-step, while risks are identified in the second sub-step. In the third sub-step, the identified risks are evaluated with respect to the specified risk acceptance criteria.

The third main step is concerned with identifying indicators to monitor the unacceptable risks identified in the previous main step. We distinguish between two sub-steps. Sensors to be deployed in the relevant part of business are identified in the first sub-step. In the second sub-step we identify indicators to be calculated based on data gathered by the sensors and we specify requirements to the indicators with respect to the deployed sensors. The internal validity of the set of indicators is evaluated in the fourth main step. We distinguish between two sub-steps. In the first sub-step we reformulate the precise business objective by expressing it in terms of the identified indicators. In the second sub-step we evaluate the internal validity of the set. The set is internally valid if the precise business objective expressed in terms of the indicators correctly measures the degree to which the business objective is fulfilled. For each indicator we evaluate whether it is internally valid based on a set of criteria.

In the fifth main step we specify designs for indicators, i.e., how they should be calculated based on data gathered by the sensors, while in the sixth main step, we evaluate whether the set of indicators has construct validity with respect to the business objective. The set has construct validity if the gathering of the sensor measurements of each indicator is suitable with respect to its requirements. Construct validity is evaluated based on a set of criteria. To evaluate the different criteria, we re-do the risk analysis from the second main step with the precise business objective replaced by the reformulated precise business objective. The latter objective is the precise business objective expressed in terms of indicators. For each indicator we identify risks towards the correctness of the reformulated precise business objective that are the result of threats to criteria for construct validity that the indicator needs to fulfill. If the risk analysis does not result in any new unacceptable risks, then we have established construct validity for each indicator. If the set of indicators is both internally valid and has construct validity with respect to the business objective, then we have established that the set is valid.

5.2.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

The methodology is used for capturing the impact of service dependencies on risk to the quality of provided services in interconnected systems, and for setting up monitoring of selected risks by the use of indicators for the purpose of providing a dynamic risk picture for the provided services. The methodology is described in detail in Chapter 10.

The client, on whose behalf the methodology is applied, controls some of the interconnected systems. These systems depend on other systems, which are controlled by other parties. In the first main step of the methodology, we document the interconnected systems in the form of a target model. The model documents systems, services, quality requirements to services, service dependencies, and trust relations. The trust relations are used when analyzing service dependencies involving systems of which we
have insufficient documentation. Each relation assigns a trust level to a quality requirement. The trust level states the degree to which the client trusts the service to be delivered according to the quality requirement in question. These trust levels are used in the second main step.

In the second main step, we conduct a risk analysis to capture the impact of service dependencies on risk to quality of provided services. For a provided service, its quality is represented by a number of quality attributes. For each provided service to be analyzed, we identify in the first sub-step one or more quality assets, where each asset represents a quality attribute of the service. By identifying these assets we restrict the identification of risks caused by service dependencies to only those risks that may harm the quality of provided services. The next sub-step is to construct high-level risk models of the impact of service dependencies on the identified quality assets. The models are constructed schematically from the target model by following a schematic procedure. These models establish a high-level understanding of how the failure of services to be delivered according to their quality requirements may lead to the failure of dependent services to be delivered according to their quality requirements. In the third sub-step, these high-level models are further detailed in order to establish a risk picture that can be monitored. As part of this detailing, we use the trust levels identified in the first main step to estimate likelihoods of quality requirements not being achieved for services involving systems of which we have insufficient documentation.

The third main step concerns the identification of risks to be monitored, as well as identification of indicators for monitoring their risk values. In the fourth main step we specify the design and deployment of the identified indicators. The designs specify how the indicators should be calculated, while the deployments specify how the indicators should be embedded in the interconnected systems. More precisely, the deployments specify how data needed in the calculations should be extracted and transmitted within the interconnected systems.

5.2.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

The architectural pattern serves as a basis for implementing monitoring tools, or more specifically enterprise level monitoring tools based on indicators. These are tools that: collect low-level indicators from the ICT infrastructure or similar; aggregate the low-level indicators into high-level indicators, useful at the enterprise level; and present the high-level indicators in a way that is understandable to the intended users. In the following we explain how the pattern is applied for constructing an enterprise level monitoring tool based on indicators. For a detailed description of the architectural pattern, we refer to Chapter 11.

The architectural pattern divides an enterprise level monitoring tool into three components; MonitorModel which collects low-level indicators from some data source and aggregate these indicators into high-level indicators; MonitorConsole which presents results from the monitoring in the form of high-level indicators and results from the evaluation of these indicators in a specific way to a specific group of users; and MonitorConfig which is used to set up the enterprise level monitoring tool and to configure it during run-time.

The MonitorConfig makes use of a data configuration file to configure the MonitorModel. This file specifies the low-level indicators to be collected and functions for
aggregating these indicators into high-level indicators. The MonitorConsole is configured by the use of a presentation model. This model specifies how results from the monitoring should be presented to a specific group of users. The presentation model is parameterized with respect to high-level indicators found in the MonitorModel’s data configuration file. The MonitorConsole can therefore easily be updated when high-level indicators, referred to in its presentation model, change. In the case of risk monitoring, the presentation model will typically be a risk model. In this risk model, some of the likelihood and consequence values, used to calculate risk values, have been replaced by high-level indicators. Likelihood and consequence values will be updated as a result of high-level indicators being updated, which again results in risk values being updated.

5.3 Overview of industrial case studies

5.3.1 Empirical study on trust-based decisions in interconnected systems

The first empirical study was conducted as part of an industrial project focusing on the use of a UML-based trust analysis method to model and analyze a public eProcurement system. This system makes use of a Validation Authority (VA) service for validating electronic IDs (eIDs) and digital signatures. The trust analysis was conducted on behalf of Det Norske Veritas (DNV) in the autumn of 2008. The goal of the trust analysis was to obtain a better understanding of the potential usefulness of a VA service for supporting trust-based decisions in systems which rely on electronically signed documents. The empirical study and the industrial project are described in detail in Chapter 12. In the following we provide an overview of the industrial project and the trust analysis method used, and we describe the main results from the empirical study.

Public eProcurement is used by public authorities, for instance within the EU, to award public contracts to economic operators. To apply for public contracts, economic operators submit electronically signed tenders, containing legal, financial, and technical information, to a public eProcurement system. A public eProcurement system must be aware of the risk implied by accepting digital signatures. The eProcurement system does not have information on whether the tender is authentic or not, i.e., whether the economic operator is willing or not willing to fulfill the tender. Thus, a possible scenario is that an economic operator can refute the validity of the submitted tender, if awarded the contract. The system can ensure that this risk is acceptable by making an assessment of the signature quality and accepting only those of a certain quality. The higher the quality is, the harder it would be for an economic operator to refute the validity of their tender. The quality of a signature can be decided from the quality of the eID, which is derived from the certificate policy of the certificate authority and the cryptography used. The eProcurement system can use a VA to assess the quality of digital signatures with respect to a quality policy set by the system. Based on the assessments, the VA informs the system whether the signatures are to be trusted or not.

The trust analysis method used for modeling and analyzing the eProcurement system consists of three main steps. In the first step we model the target. The models should only capture the aspects of the system (eProcurement system) and other actors
Summary of contribution

(the VA and economic operators) that enhance our understanding of the decisions that are taken on the basis of trust and the considerations that lie behind these decisions, as well as the resulting system behavior and outcomes that are relevant. In the second step we conduct the actual analysis. This involves investigating the current system behavior and the way in which trust-based decisions are being made, as well as potential alternative behaviors. The aim is to obtain a good understanding of the risks and opportunities involved. For an eProcurement system the risks are to accept trusted tenders that are non-authentic and to reject not trusted tenders that are authentic. Moreover, the opportunities are to accept trusted tenders that are authentic and to reject not trusted tenders that are non-authentic. In the third step we use the obtained knowledge to form policies to ensure and enforce the desirable behavior. The aim here is to select a quality policy that results in an optimal balance between risks and opportunities.

The empirical study motivates the need for using trust analysis methods when reasoning about the behavior of systems/actors in cases where the behavior of these systems/actors may result in risks and/or opportunities. For the particular method applied, the empirical study gave strong indications that the trust analysis method is feasible in practice. The empirical study also shows that: this kind of trust analysis can be carried within the frame of 100 man-hours (not including writing of a final report); there were no instances were the analysts (researchers) were not able to capture the relevant information in the models; and the models to a large extent were comprehensible for the industrial participant with some experience in UML but no background in the specific extensions used by the method.

5.3.2 Empirical study on the design of indicators for monitoring risk

The second empirical study was integrated in a commercial security risk analysis conducted in 2010. In this analysis, indicators were designed for the purpose of validating likelihood estimates obtained from expert judgments. In the following we provide a brief overview of the steps of the commercial security risk analysis that were of relevance to the empirical study, and we present the main results from the study. We refer to Chapter 13 for further details on the commercial security risk analysis and the empirical study.

The commercial security risk analysis included a six step process that was of relevance to the empirical study. The empirical study builds on data collected during the analysis and on semi-structured interviews with domain experts that participated on behalf of the client in the analysis. In Step 1 of the process, the domain experts provided the analysis team (the researchers) with likelihood estimates for security risks based on expert judgments. Indicators for validating the likelihood estimates were identified in Step 2. The analysis team designed a number of indicators, and these indicators were revised during a meeting with the domain experts in Step 3. During this meeting some indicators were rejected, some were subject to minor modifications, and some new indicators were identified. In Step 4 the analysis team formulated validation criteria for the likelihood estimates in terms of indicators. Each criterion specifies the expected values of the indicators related to the likelihood estimate in question. Here, each criterion makes a prediction about the value of a set of indicators under the assumption that the likelihood estimate in question is correct. Indicator values
were obtained by the domain experts in Step 5. In Step 6 the validation criteria were evaluated and some of the initial likelihood estimates were adjusted.

One result from the empirical study was that two out of 28 likelihood estimates were adjusted, while the main result was the identification of a number of challenges related to design of indicators for monitoring security risks. First, the empirical study shows that it is challenging to design indicators for which it is feasible to obtain values within the available time and resources of a security risk analysis. For a number of the indicators designed, their values were not obtainable within the client’s organization. By having some knowledge on the kinds of historical data that are available within the organization and whose responsible for the different kinds of data, it should be easier to both identify indicators and obtain their values. Unfortunately, it may be difficult to obtain this knowledge since data is often spread across the organization and since few, if any, have a complete overview of the data available. Second, it is challenging to relate likelihood estimates to indicators. It is especially difficult to predict how indicator values affect a likelihood estimate when the indicators are only indirectly related to the estimate in question. This will typically be a problem when formulating validation criteria for likelihood estimates of incidents that that are not easily observable. Third, the indicator values obtained from an organization may vary when it comes to correctness. In order to get the most out of indicator-based monitoring, the uncertainty of the values should be taken into account. Moreover, one should strive to reduce uncertainty by using several independent indicators to calculate/validate the same estimate.
Chapter 6

Overview of research papers

The main results of the work presented in this thesis are documented in the papers presented in Part II of the thesis. In this chapter we provide an overview of these papers. For each paper we describe its main topics and indicate how much of the work that is credited to the author of this thesis.

6.1 Paper A: ValidKI: A method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

Authors: Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen.

Publication status: Technical report SINTEF A23413, SINTEF ICT, 2012. The report presented in the thesis is an extended and revised version of the paper published in International Journal on Advances in Intelligent Systems (vol. 5, no. 1-2, 2012) [143]. This paper is again an extended and revised version of the paper published in proceedings of the First International Conference on Business Intelligence and Technology (BUSTECH’2011) [144]. The latter paper received a best paper award at the conference.

My contribution: Olav Skjelkvåle Ligaarden was the main author, responsible for about 90% of the work.

Main topics: The report presents the method ValidKI (Valid Key Indicators), which is a method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. The main focus of the method is on the design of valid indicators. A set of indicators is valid with respect to a business objective if it measures the degree to which the business or relevant part thereof fulfills the business objective. The method is divided into six main steps. In the report, the method is demonstrated on an example case focusing on the use of electronic patient records in a hospital environment. The main output from the method is specifications that describe the design and deployment of
the indicators. These specifications are to be used to implement ICT-based monitoring of the indicators.

In ValidKI, indicators are designed for monitoring unacceptable risks to the fulfillment of business objectives. A business objective is fulfilled if all of its unacceptable risks become acceptable as a result of the monitoring. Acceptable risks to the fulfillment of business objectives may be thought of to represent uncertainty we can live with. In other words, their potential occurrences are not seen to significantly influence the fulfillment of the business objectives. The validity of the indicators is evaluated based on validation criteria from the domain of software engineering metrics. The validation criteria used are general, thus not specific to software engineering.

6.2 Paper B: Using indicators to monitor risk in interconnected systems: How to capture and measure the impact of service dependencies on the quality of provided services

Authors: Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen.


My contribution: Olav Skjelkvåle Ligaarden was the main author, responsible for about 90% of the work.

Main topics: The report presents a method for capturing the impact of service dependencies on risk to the quality of provided services in interconnected systems, and for setting up monitoring of selected risks by the use of indicators for the purpose of providing a dynamic risk picture for the provided services. The method is divided into four main steps focusing on documenting the interconnected systems and their service dependencies, establishing the impact of service dependencies on risk to quality of provided services, identifying measurable indicators for dynamic monitoring, and specifying their design and deployment, respectively. These design and deployment specifications are to be used to implement risk monitoring based on the identified indicators. The report describes each of the four main steps as well as their sub-steps in terms of a detailed guideline. The method is illustrated in an example-driven fashion based on a case study from the domain of power supply.
6.3 Paper C: An architectural pattern for enterprise level monitoring tools

**Authors:** Olav Skjelkvåle Ligaarden, Mass Soldal Lund, Atle Refsdal, Fredrik Seehusen, and Ketil Stølen.

**Publication status:** Published in proceedings of the 2011 IEEE International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA’2011) [146].

**My contribution:** Olav Skjelkvåle Ligaarden was the main author, responsible for about 90% of the work.

**Main topics:** The paper presents an architectural pattern to serve as a basis for building enterprise level monitoring tools based on indicators. These are tools that: collect low-level indicators from the ICT infrastructure or similar; aggregate the low-level indicators into high-level indicators, useful at the enterprise level; and present the high-level indicators in a way that is understandable to the intended users. In the paper we identify the core components of such tools and describe their interactions. The pattern is the result of generalizing the architecture of a risk monitor that exhibits a number of features that are not specific to the monitoring of risks, but general to a broad class of enterprise level monitoring tools. In the paper, we demonstrate the pattern by showing the risk monitor as an instance, and we exemplify the use of the risk monitor in a health care scenario.

**Errata:** Replace “design pattern” with “architectural pattern” in Section V (Conclusion) in Paper C.

---

6.4 Paper D: Experiences from using a UML-based method for trust analysis in an industrial project on electronic procurement

**Authors:** Tormod Vaksvik Håvaldsrud, Olav Skjelkvåle Ligaarden, Per Myrseth, Atle Refsdal, Ketil Stølen, and Jon Ølnes.

**Publication status:** Published in Journal of Electronic Commerce Research (vol. 10, no. 3-4, 2010) [147].

**My contribution:** Olav Skjelkvåle Ligaarden was one of two main authors, responsible for about 45% of the work.

**Main topics:** The paper reports on experiences from using a UML-based method for trust analysis in an industrial project. The overall aim of the trust analysis method is to provide a sound basis for making trust policy decisions. The method makes use of UML sequence diagrams extended with constructs for probabilistic choice and subjective
belief, as well as the capture of policy rules. The trust analysis method is evaluated with respect to a set of criteria. The industrial project focused on the modeling and analysis of a public electronic procurement (eProcurement) system making use of a validation authority service for validating electronic certificates and signatures. The evaluation of the method gave strong indications that the trust analysis method is feasible in practice.

6.5 Paper E: Experiences from using indicators to validate expert judgments in security risk analysis

Authors: Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen.


My contribution: Olav Skjelkvåle Ligaarden was the main author, responsible for about 90% of the work.

Main topics: The report presents experiences from a commercial security risk analysis where indicators were used to validate likelihood estimates obtained from expert judgments. The experiences build on data collected during the analysis and on semi-structured interviews with the client experts who participated in the analysis. The empirical study identified several challenges related to the design of indicators for monitoring security risks.
Chapter 7

Discussion

In this chapter, we evaluate and discuss the contributions of this thesis. In Chapter 2 we defined success criteria for the framework and the three artifacts. In Section 7.1 we evaluate to what extent we have fulfilled the different success criteria, while in Section 7.2 we discuss how the three artifacts of this thesis relate to and extend the state of the art presented in Chapter 4.

7.1 Fulfillment of the success criteria

In the following three sub-sections we evaluate the success criteria of Artifacts 1–3. The success criterion of the framework as a whole is evaluated in the fourth sub-section, since this criterion depends on the evaluation of the other success criteria.

7.1.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

Success criterion 2 The application of the method results in indicators that measure correctly to what extent the business objectives received as input are fulfilled.

As previously explained, the industrial case study described in Paper E (presented in Chapter 13) focused on the use of indicators to validate likelihood estimates obtained from expert judgments. The need for indicators that measure correctly to what extent the business objectives received as input are fulfilled is motivated by experiences from this study. In particular, the study identified two cases where likelihood estimates could not be validated because the indicator values were too uncertain.

The method delivers a set of indicators that is valid with respect to a business objective received as input, where valid is defined as follows: a set of indicators is valid with respect to a business objective if it is valid in the following two ways:

1. **internal validity** – the precise business objective expressed in terms of the indicators correctly measures the degree to which the business objective is fulfilled; and
2. **construct validity** – the gathering of the sensor measurements of each indicator is suitable with respect to its requirements specification.

We claim that our notion of validity is a good approximation of correctness referred to in the success criterion. Firstly, the distinction between internal and construct validity is well-established in the literature. According to Meneely et al. [63], a metric has internal validity if "the metric measures the attribute it purports to measure." Similar definitions are given in [67, 149]. Moreover, [63] presents a systematic literature review of validation criteria for software metrics. The review ultimately ended up focusing on 20 papers. Most of the authors of these papers discuss some form of internal validation.

According to Meneely et al., a metric has construct validity if "the gathering of a metric’s measurements is suitable for the definition of the targeted attribute." This definition is based on [150] which refers to construct validity as when "the operational definition yields data related to an abstract concept." In both cases, construct refers to the implementation of a metric/indicator. As shown by Meneely et al., a number of the authors of the reviewed papers discuss validation criteria that can be classified as construct validity types of criteria, i.e., that they concern the validation of the implementation of metrics/indicators.

Our definitions of internal and construct validity follow the definitions given by Meneely et al. As can be seen in Section II-B in Paper A (presented in Chapter 9), we use a risk-based approach to evaluate the fulfillment of business objectives. For each business objective, we identify risks towards its fulfillment. Indicators are used to measure risks that represent uncertainty we cannot live with, i.e., that the potential occurrences of the risks significantly influence the fulfillment of the business objective. More precisely, the indicators are used to measure risk attributes (likelihood or consequence). Each indicator has internal validity if it measures such an attribute correctly.

In the case of construct validity, each indicator is associated with a requirements specification which specifies what it means to measure the risk attribute. The indicator has construct validity if it can be implemented correctly, i.e., if the gathering of the sensor measurements for which the calculation of the indicator relies is suitable with respect to the requirements specification.

By reviewing the 20 papers, Meneely et al. extracted and categorized 47 unique validation criteria for software metrics. A number of these criteria are general, thus not specific to software engineering. For some of the validation criteria, Meneely et al. noticed that the criteria are not atomically satisfiable criteria, but broad categories that can contain other criteria. The main categories are: internal, external, and construct validity. According to Meneely et al., a metric has external validity if "it is related in some way (e.g., by prediction, association, or causality) with an external quality factor." An external quality factor is an external attribute that is measured by a metric. In software engineering, the relationships between internal and external attributes of software products are often examined [151]. For instance, a metric for the internal attribute code size has external validity if it is related to the metric for the external attribute maintainability in some way.

The business objectives that we address focus on quality. The attributes that we measure by the use of indicators are of course related in some way to the quality attributes represented by the business objective. However, external validity is not rele-
7.1 Fulfillment of the success criteria

vant in our case, since we do not measure to what extent a business objective is fulfilled by relating indicators that measure different attributes to indicators that measure quality attributes. Instead, we measure to what extent a business objective is fulfilled by measuring risk attributes of risks whose potential occurrences may significantly influence the fulfillment of the business objective. This means that none of the validation criteria that Meneely et al. have classified as external validity types of criteria are relevant.

To evaluate internal and construct validity, we have taken a number of the criteria categorized in the internal and construct validity categories into consideration. In the following we justify the selection and rejection of different criteria. In the case of internal validity, the criterion “attribute validity” is taken into consideration. According to Meneely et al., a metric has attribute validity if “if the measurements correctly exhibit the attribute that the metric is intending to measure.” This criterion is relevant since it helps us to evaluate whether an indicator correctly exhibits the risk attribute (likelihood or consequence) of the risk that it is measuring.

In [63], the criterion “representation condition” is classified as a category. According to Meneely et al., a metric satisfies the representation condition if “the attribute is a numerical characterization that preserves properties of both the attribute and the number system it maps to.” The representation condition is a property from measurement theory. Under the representation condition, any property of the number system must appropriately map to a property of the attribute being measured and vice versa. From this category we have selected the following criteria:

- “appropriate continuity” (definition according to [63]: “a metric has appropriate continuity if the metric is defined (or undefined) for all values according to the attribute being measured”);
- “dimensional consistency” (definition according to [63]: “a metric has dimensional consistency if the formulation of multiple metrics into a composite metric is performed by a scientifically well-understood mathematical function”); and
- “unit validity” (definition according to [63]: “a metric has unit validity if the units used are an appropriate means of measuring the attribute”).

The criterion “appropriate continuity” is relevant since it helps us to check whether the indicator has any unexpected discontinuities. Such discontinuities may for instance arise from fraction calculation with a zero denominator. On the other hand, the criterion “dimensional consistency” is relevant for evaluating whether information is lost during the construction of a composite indicator from basic indicators. Loss of information may for instance be experienced if different scales are used for the basic and composite indicators. And finally, the criterion “unit validity” is relevant for evaluating whether the indicator’s unit is appropriate for measuring the risk attribute.

In the “representation condition” category we also find the criterion “scale validity.” According to Meneely et al., a metric has scale validity if “it is defined on an explicit, appropriate scale such that all meaningful transformations of the metric are admissible.” The scales discussed are typically nominal, ordinal, interval, ratio, and absolute. Each scale type comes with a set of admissible transformations. During the evaluation of “dimensional consistency” we can check whether appropriate scales are being used. Thus, it is not necessary to evaluate the criterion “scale validity” separately. The category also contains a number of criteria that are not relevant since they
are specific to software engineering metrics. This is the case for “appropriate granularity,” “interaction sensitivity,” “monotonicity,” “permutation validity,” and “renaming insensitivity.” We also do not find following criteria in the same category relevant:

- “increasing growth validity” (definition according to [63]: “a metric has increasing growth validity if the metric increases when concatenating two entities together”) and
- “non-uniformity” (definition according to [63]: “a metric has “non-uniformity if it can produce different values for at least two different entities”).

A metric does not have non-uniformity if it for instance rates all software programs as equal. Then the metric is not really a measure. With risks being our entities, neither of the criteria is relevant. In the method in Paper A, risks can be combined if the pull in the same direction. Since this is done before we identify indicators, the criterion “increasing growth validity” is not relevant. In our context, an indicator is used to measure a risk attribute of a risk at different points in time. We have only two requirements to the values produced by an indicator. The first requirement is that the indicator should measure the risk attribute correctly, while the second is that the indicator should not produce values that always result in the risk being unacceptable or acceptable. These requirements are checked during the evaluation of the “attribute validity” criterion. This means that the “non-uniformity” criterion is not relevant.

Besides the software engineering specific validation criteria already mentioned, we also find the category “content validity,” the criterion “product and process relevance” contained in this category, and the two criteria “actionability” and “transformation invariance” most relevant in a software engineering setting. In addition, we do not find the following two criteria relevant:

- “economic productivity” (definition according to [63]: “a metric has economic productivity if using the metric quantifies a relationship between cost and benefit”) and
- “non-exploitability” (definition according to [63]: “a metric exhibits non-exploitability if developers cannot manipulate a metric to obtain desired results”).

A metric/indicator does not have “economic productivity” if it does not result in saving money in the long run. We do not consider this criterion since it more relevant to consider the cost/benefit of an indicator when taking a decision on whether to implement it or not. In the case of the “non-exploitability” criterion, Meneely et al. introduces the term “exploitability” to describe the phenomenon where people can manipulate a metric’s measurements without changing the attribute being measured. Contrary to Meneely et al., we consider this criterion during the evaluation of construct validity rather than during the evaluation of internal validity. We do not evaluate the criterion directly. Instead, we consider manipulation of an indicator as one of many threats to construct validity types of criteria.

In addition to the validation criteria already selected, we have also selected the following two criteria from the internal validity category:

- “internal consistency” (definition according to [63]: “a metric has internal consistency if “all of the elementary measurements of a metric are assessing the same construct and are inter-related””) and
7.1 Fulfillment of the success criteria

- “factor independence” (definition according to [63]: “a metric has factor independence if the individual measurements used in the metric formulation are independent of each other”).

The first criterion applies especially to indicators that are composed of basic indicators. If the basic indicators are not conceptually related, then the composite indicators will not have internal consistency and for that reason be hard to interpret. The second criterion also applies especially to indicators that are composed of basic indicators. If the basic indicators use measures that are not independent of each other, then the composite indicator’s ability to measure a risk attribute may be affected. In the category the “factor independence” criterion belongs to, we also find the “causal relationship validity” criterion. According to Meneely et al., a metric has causal relationship validity if “it has a causal relationship to an external quality factor.” In our opinion, it would have been more suitable to categorize the criterion as an external validity type of criteria than an internal validity type of criteria, since it concerns the relationship between a metric/indicator and an external quality factor. We have therefore not selected this criterion. It should also be noticed that we have not selected the criterion “metric reliability.” According to Meneely et al., a metric has metric reliability if “the measurements are “accurate and repeatable”. Even though this criterion is relevant, we have not selected it since it is covered by the “internal consistency” and “attribute validity” criteria, and by construct validity types of criteria.

In the case of construct validity, we consider the following criteria:

- “instrument validity” (definition according to [63]: “a metric has instrument validity if the underlying measurement instrument is valid and properly calibrated”);
- “stability” (definition according to [63]: “a metric has stability if it produces the same values “on repeated collections of data under similar circumstances””); and
- “definition validity” (definition according to [63]: “a metric has definition validity if the metric definition is clear and unambiguous such that its collection can be implemented in a unique, deterministic way”).

The “instrument validity” criterion is relevant for evaluating whether the sensors provide correct data to the calculations of the indicators, while the “stability” criterion is relevant for evaluating whether calculations of indicators that involves human decisions result in correct indicators. Moreover, the “definition validity” criterion is relevant since it concerns the implementation of indicators. To implement indicators correctly, their designs must be given in a clear and unambiguous way.

On the other hand, the following criteria from the construct validity category are not considered:

- “protocol validity” (definition according to [63]: “a metric has protocol validity if it is measured by a widely accepted measurement protocol”);
- “usability” (definition according to [63]: “a metric has usability if it can be cost-effectively implemented in a quality assurance program”); and
- “notation validity” (definition according to [63]: “a metric has notation validity if the metric is reasoned about “mathematically with precise, consistent notation””).
We do not consider the “protocol validity” criterion since we will in many cases calculate indicators based on measures for which there do not exist widely accepted measurement protocols. The “usability” criterion is also not considered since it is similar to the criterion “economic productivity” discussed earlier. It is more relevant to consider the “usability” criterion when taking a decision on whether to implement the indicator or not. In the case of the “notation validity” criterion, it should be noticed that this criterion is a member of the category “definition validity.” We do find “notation validity” to be a too strict criterion. In many cases we will design indicators that cannot be reasoned about “mathematically with precise, consistent notation.” The “definition validity” criterion is more general, since it can be used to evaluate the design of all kinds of indicators. This completes the evaluation of the success criterion.

**Success criterion 3** The application of the method results in specifications of indicators that are well-suited for ICT-based monitoring.

It seems fair to argue that the specifications are well-suited for ICT-based monitoring if they contain information that makes it easy for developers to correctly implement the indicators. In our method, as described in Paper A (presented in Chapter 9), each indicator is specified in terms of:

- a deployment specification for the sensors used by the indicator;
- a requirements specification; and
- a design specification.

The deployment specification pinpoints the locations of the sensors. The requirements specification is basically a pre-post specification formally defining the indicator function, while the design specification, in the form of a UML [94] sequence diagram, describes its implementation within the target. The decision of using UML sequence diagrams was not only based on their ability to document the implementation of indicators. It was also based on results from the evaluation of their comprehensibility in the industrial case study described in Paper D (presented in Chapter 12). The results from this evaluation indicate that UML sequence diagrams to a large extent are comprehensible for industrial participants with some knowledge of UML. By being comprehensible, the sequence diagrams serve as an aid in communication between the analysts and the industrial participants.

The design specifications may be used to configure the sensors to extract the data needed in the calculations of indicators. Moreover, since the design specifications describe the main entities (sensors, components, humans) that need to take part in the calculation of indicators and their interactions in the form of data exchanges, they provide a good starting point for implementing monitoring tools, ICT-based work processes, and the infrastructure needed for monitoring the indicators. In addition, with the design specifications given in the form of UML sequence diagrams, developers can detail these diagrams further and achieve specifications that are close to actual implementations. All of this is very much in line with best-practice software engineering.
7.1 Fulfillment of the success criteria

**Success criterion 4** The method is applicable in an industrial context within acceptable effort.

Paper A (presented in Chapter 9) demonstrates the method on a large, realistic example case focusing on the use of electronic patient records in a hospital environment. The example case covers in detail all steps of the method. Based on this we may at least claim that the method is applicable in an industrial context given the following two assumptions:

- **a)** the client is able to provide the required data; and
- **b)** the analyst is equal to the method designer (and main author of Paper A).

In the following we argue that assumption a) is justified. In Section II-B in Paper A, we have described the different models/descriptions that are developed when the method is applied. An overview of the different models/descriptions is given in Fig. 2 in Paper A. To develop the different models/descriptions, the analyst relies on relevant data to be provided by the client and its co-operating parties. The business objectives for which indicators should be designed are central to all the models/descriptions to be developed. Since these business objectives are central to the success of the client’s organization, they are easily obtainable by the client.

The specification “Specification of relevant part of business” documents the actors/systems that are to comply with the business objectives. It also documents how these actors/systems interact by exchanging data. To develop this specification we rely on the client to provide data on relevant actors/systems. If the fulfillment of the business objectives also relies on actors/systems of parties that the client co-operates with, then data on these actors/systems must be provided as well. The specification is to be used to specify the deployment of sensors for monitoring data exchanges between actors/systems. The data needed for developing the specification is available and easily obtainable both within the client’s and the co-operating partners’ organizations. To develop the specification we only rely on high-level descriptions of the architectures of the client’s and the co-operating parties’ ICT infrastructures and data on the actors that rely on these infrastructures.

To develop the other eight models/descriptions described in Section II-B in Paper A, we rely on information provided by domain experts that act on behalf of the client. We require domain experts that have knowledge about the business objectives, the relevant part of business and its threat picture, and the ICT infrastructures of the client and the co-operating parties. Knowledge about the latter is needed in order to design indicators for ICT-based monitoring. The client and its co-operating partners are able to provide us with domain experts with the above mentioned knowledge, since the knowledge that we require is essential for the operation of both the client’s and the co-operating partners’ organizations.

In the following we argue that assumption b) may be weaken into the analyst being an experienced risk analyst. In fact, we claim that most analysts with some experience in risk analysis and a basic understanding of indicators/metrics would be able to do equally well as the method designer. An important part of every risk analysis is to understand the target to be analyzed. An experienced risk analyst is therefore able to conduct Step 1 of the method, since this step is all about understanding the target to be analyzed. Moreover, an experienced risk analyst is able to conduct Steps 2 and 6.
of the method, since these steps concern the application of risk analysis. He/she will also have some knowledge about, and quite possibly experience with, risk monitoring. Thus, he/she will have a basic understanding of metrics/indicators. The experienced risk analyst is therefore capable of conducting Step 3 when supported by the domain experts. By interacting with the domain experts, the experienced risk analyst can identify and document sensors to be deployed in the ICT infrastructures of the client and its co-operating partners, as well as specifying requirements to indicators with respect to the identified sensors. The experienced risk analyst is also able to conduct Step 4 when supported by the domain experts. By interacting with the domain experts, the experienced risk analyst can evaluate to what extent the different criteria to internal validity are fulfilled for the different indicators. He/she is also capable of specifying the designs of indicators (Step 5) when supported by the domain experts, since an experienced risk analyst has experience with UML [94] sequence diagrams or similar documentation approaches.

The arguments above show that an analyst only needs a basic understanding of indicators/metrics, since he/she is supported by domain experts during the design of the indicators.

Regarding the effort, consider Table 7.1 estimating the expected effort for the example case in Paper A. We assume that the analyst team consists of two persons; one analysis leader and one analysis secretary. The first leads the meetings with the client, while the second is responsible for documenting the results of the analysis. Moreover, we assume that three domain experts participate on behalf of Client H (the client); one from Client H and one from each of the two co-operating partners Blood test analysis and X-ray. The three domain experts have knowledge about the business objectives, the relevant part of business and its threat picture, and the ICT infrastructures of Client H, Blood test analysis, and X-ray.

In Table 7.1 we have estimated the expected effort in hours for each step and sub-step of the method for the Analyst (total for the two analysts) and the Client (total for the three domain experts). We estimate both time spent in meetings (M) and time spent between and preparing for meetings (P). Both of the analysts participate in all the steps and sub-steps of the method. This is also true for all of the domain experts.

Step 1.1 and Step 1.2 are conducted as 2 and 3 hours meetings, respectively. In Step 1.1, the domain expert from Client H presents the business objective and constraints that should be satisfied in order to fulfill the objective. Based on this presentation, the business objective is expressed more precisely. Since only one business objective is considered in the example case, a 2 hours meeting should be sufficient. For the same reason, a preparation time of 3 hours should be sufficient for giving the presentation. No preparation time is needed for the analysts and the other two domain experts.

In the case of Step 1.2, the domain experts give a presentation of the different actors/systems and their interactions that may be of relevance for describing the relevant part of business. A preparation time of 5 hours in total should be sufficient for the domain experts. The domain experts will also provide the two analysts with documentation before the meeting. A preparation time of 5 hours is reasonable for each of the analysts. Based on the presentation, the participants of the meeting discuss the relevance of the different actors/systems and interactions. The analysts use the results from this discussion and the documentation provided by the domain experts to develop a specification of the relevant part of business. As can be seen in Fig. 4 in Paper A, this development results in a high-level description of how different actors/systems interact.
Table 7.1: Expected effort (in hours) for the Analyst (total for the two analysts) and the Client (total for the three domain experts) with respect to the example case in Paper A. M = “time spent in meetings,” while P = “time spent between and preparing for meetings”

<table>
<thead>
<tr>
<th>Step</th>
<th>Analyst</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>M</td>
</tr>
<tr>
<td>Step 1: Establish target</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>Step 1.1: Express business objectives more precisely</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Step 1.2: Describe relevant part of business</td>
<td>22</td>
<td>6</td>
</tr>
<tr>
<td>Step 2: Identify risks to fulfillment of business objective</td>
<td>20</td>
<td>18</td>
</tr>
<tr>
<td>Step 2.1: Specify risk acceptance criteria</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Step 2.2: Risk identification and estimation</td>
<td>20</td>
<td>12</td>
</tr>
<tr>
<td>Step 2.3: Risk evaluation</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Step 3: Identify key indicators to monitor risks</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>Step 3.1: Deploy sensors to monitor risks</td>
<td>11</td>
<td>6</td>
</tr>
<tr>
<td>Step 3.2: Specify requirements to key indicators wrt deployed sensors</td>
<td>12</td>
<td>6</td>
</tr>
<tr>
<td>Step 4: Evaluate internal validity</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>Step 4.1: Express business objective in terms of key indicators</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Step 4.2: Evaluate criteria for internal validity</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>Step 5: Specify key indicator designs</td>
<td>24</td>
<td>6</td>
</tr>
<tr>
<td>Step 6: Evaluate construct validity</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>Total (preparations and meetings)</td>
<td>112</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>136</td>
</tr>
<tr>
<td>Write the analysis report</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>Total (preparations, meetings, and analysis report)</td>
<td>220</td>
<td>136</td>
</tr>
</tbody>
</table>
Based on this, we would say that no more than 10 hours are needed for developing the specification. After the specification has been developed it needs to be checked by the domain experts. Each domain expert should not need more than 1 hour to check the correctness of the specification, while the analysts should not need more than 2 hours to implement changes proposed by the domain experts.

In Step 2.1 we specify the risk acceptance criteria. A 1 hour meeting should be sufficient for specifying the criteria. In Step 2.2 we identify risks towards the fulfillment of the business objective and we estimate their risk values. The results from this step are documented in Figs. 5–9 in Paper A. Even though it does not seem as much work to develop the risk models depicted in Figs. 5–9, we expect that it may take as much as two meetings of 3 hours each and 10 hours spent by the analysts outside the meetings to develop the risk models. Moreover, we believe that the analysts would require 5 hours each to prepare for the meetings. No preparation is necessary for the domain experts in this case. In the case of Step 2.3, we both evaluate risks and accumulate risks that pull in the same direction. A 2 hours meeting should be sufficient for conducting this step. The risk evaluation mainly consists of plotting the risks according to their likelihoods and consequences in a risk matrix given by the risk acceptance criteria, while risks are accumulated by accumulating their likelihood and consequence values.

Steps 3.1 and 3.2 will be conducted in parallel. Fig. 10 in Paper A specifies the deployment of sensors in the relevant part of business. Based on this figure, we would say that it is possible to identify these sensors within a time period of 3 hours. Moreover, 3 hours should also be sufficient for conducting Step 3.2. This means that a 6 hours meeting is sufficient for conducting Steps 3.1 and 3.2. In order to conduct this meeting within 6 hours, the analysts should come up with proposals to sensor deployments and requirements to indicators with respect to the deployed sensors before the meetings. We estimate that each of the analysts would need to spend 10 hours to prepare the proposals. The domain experts should use these proposals in their preparation for the meetings. We expect that the three domain expert use a total of 10 hours to prepare for the meeting. In addition to the time spent in the meeting and for preparations, we estimate that the analysts spend 1 hour in total to develop Fig. 10 in Paper A and 2 hours in total to develop the requirements specifications in Tables V–X after the meeting has been concluded.

Not much effort is required to conduct Step 4.1. The business objective can easily be expressed in terms of indicators by using the results of Steps 2.3 and 3.2. An hour spent by one of the analysts should be sufficient for conducting this step. In the case of Step 4.2, we estimate that a 3 hour meeting should be sufficient. Moreover, an additional 2 hours would be needed by the analysts to document the results of the evaluation of internal validity after the meeting has been concluded.

The results of Step 5 are the indicator design specifications given in Figs. 11–24 in Paper A. To develop these specifications, the analysts should first come up with some proposals to the design of indicators that can be discussed in a meeting. We estimate that 10 hours should be sufficient for coming up with these proposals, while the domain experts would need about 10 hours in total to study these proposals before the meeting. A meeting of 3 hours should be sufficient for discussing the proposals. After the meeting has been conducted, the analysts may need as much as 10 hours to update the specifications. The updated specifications need to be checked by the domain experts. We estimate that a total of 3 hours are used to check their correctness. An additional 4 hours may be needed by the analysts to update the specifications that are
7.1 Fulfillment of the success criteria

not correct.

The results of Step 6 are documented in Figs. 25–30 and Table XII in Paper A. The expected effort for the analysts and the domain experts in Step 6 should be similar to the expected effort for Steps 2.2 and 2.3, since Step 6 and the two sub-steps concerns similar tasks, and since the outputs from Step 6 and the two sub-steps are similar with respect to size.

After all the meetings have been conducted, the analysts document the analysis and its results in a report. The report will contain the models/descriptions developed during the analysis, and text explaining these models/descriptions. We estimate that 40 hours should be sufficient for writing the report. The domain experts will not spend any time in relation to the report writing.

As can be seen in Table 7.1, the estimated effort for the analysts and the three domain experts for the example case in Paper A is 356 hours, when including the time spent on writing the analysis report. To assess whether this estimated effort is acceptable, we compare the estimated effort of the example case in Paper A to the estimated effort of the industrial case study in Paper E (presented in Chapter 13). The industrial case study is described in Appendix A in Paper E. Even though the two methods reported on in the two papers differ and the industrial case study in Paper E requires more effort than the example case in Paper A, the industrial case study has a number of things in common with the example case:

- Both the industrial case study and the example case focus on the fulfillment of business objectives. In the industrial case study, the client presented us with a business objective that they need to comply with. To comply with this business objective, a number of requirements must be fulfilled. The main purpose of the industrial case study was to identify and assess different risks to the fulfillment of these requirements.

- Both the industrial case study and the example case rely heavily on risk analysis by the use of CORAS and the design of indicators.

- Both the industrial case study and the example case involved two analysts and three domain experts.

Because of the similarities between the industrial case study and the example case, the industrial case study can be used to some extent to assess whether the estimated effort of the example case is acceptable. As can be seen in Table II in Paper E, the estimated effort for the industrial case study was 485 hours. Experiences from other commercial projects of this kind indicate that this is a reasonable effort. For example, CORAS [11] has been developed for analyses handling 150–300 man-hours on behalf of the analyst team alone. The estimate from the industrial case study does not include the time spent by the domain experts between meetings, because we do not have these numbers. It does also not include the time spent on writing the analysis report. Since the effort in the industrial case study was acceptable from the client’s viewpoint and since the example case is smaller than the industrial case study, we find it reasonable that the client in the example case also would find the effort acceptable.

Based on all of the above discussion, we conclude that the method is applicable in an industrial setting within acceptable effort.
7.1.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

**Success criterion 5** *The application of the method results in specifications of indicators that correctly capture and measure the impact of service dependencies on the quality of provided services.*

It seems fair to argue that the ability of the method in Paper B (presented in Chapter 10) to result in specifications of indicators that correctly capture and measure the impact of service dependencies on the quality of provided services may be decomposed into its ability to deliver:

a) a target model that correctly captures the service dependencies of relevance for the quality of the provided services;

b) a risk model that correctly captures the impact of service dependencies on risk to the quality of provided services with respect to the target model; and

c) specifications of indicators that correctly measure to what extent the risks identified for monitoring are acceptable.

Regarding a), which is the output of Step 1, it is of course possible to get the target model wrong. However, we claim that our modeling language has the essential features needed for capturing the service dependencies of relevance for the quality of the provided services. Firstly, our approach to capture relations at the system level is completely general since we use graphs to represent systems (vertices) and services (edges). Moreover, quality requirements to services are captured by annotating edges with required service levels. This is also much in line with other approaches as for example [84–87]. In [84], Holmgren models electric power delivery networks by the use of graphs. The models are used to calculate values of topological characteristics of networks and to compare their error and attack tolerance. In [85], the CIMS framework for infrastructure interdependency modeling and analysis is presented. In this framework, infrastructure entities are modeled by the use of vertices, while edges are used to represent the flow of a physical quantity, information, or influence (e.g., geographical, policy/procedural, etc.). The latter shows that the framework does not only consider dependencies that are the result of physical linkages between entities. In [86], Svendsen presents a multigraph model for critical infrastructures. The model uses vertices to represent infrastructure components that act as consumers and/or providers of different commodities, while edges represent exchanges of commodities. In addition, requirements to the exchanged commodities are specified by the use of response functions and predicates that represent the maximum capacity of an edge and the lower threshold of flow through the edge. In [87], an approach to modeling interdependent infrastructures in the context of vulnerability analysis is presented. Each infrastructure is represented by both a network model, in the form of a graph, and a functional model. The network model shows how the infrastructure’s physical components interact, while the functional model captures physical and operational characteristics of the infrastructure.

Secondly, we also facilitate the capturing of dependencies between services by using logical gates. It could of course be argued that it should be possible to put weights on
the dependencies in order to capture the strengths of the dependencies, and this is a potential generalization of our approach. We have not gone into this issue in this thesis however, since we know from experience (e.g., [147, 148]) how difficult it is to get this kind of estimates.

Thirdly, we support the modeling of trust relations in the case of third-party dependencies. These relations are captured in terms of probabilities, which is very much in line with standard trust definitions [20, 21]. The third-party dependencies involve third-party systems for which we have insufficient information. As previously explained in Section 3.4, the need for reasoning about trust is motivated by the industrial case study described in Paper D (presented in Chapter 12). In this paper, trust is used to decide whether tenders submitted by economic operators for which we have insufficient information should be trusted to be authentic or not.

One issue we have not gone into is the issue of uncertainty. We acknowledge that getting exact estimates regarding probabilities in particular and quantitative measures in general is problematic. However, we claim that our methodologies may be generalized to work with intervals following the recommendations in [152].

Regarding b), which is the output of Step 2, high-level risk models are schematically constructed from the target model by following the schematic procedure specified in Section 3.2.2 in Paper B. By following this procedure, the high-level impact of service dependencies on risk to quality of provided services is documented. The high-level risk models are constructed by the use of the CORAS risk modeling language [11]. This language is part of the CORAS approach, which is based on the ISO 31000 standard [88] which is preceded by the AS/NZS 4360 standard [90]. To achieve a detailed understanding of the impact of service dependencies on risk to quality of provided services and to establish a risk picture that can be monitored, the high-level risk models are detailed by using the CORAS risk modeling language. The language has characteristics that support the construction of correct risk models. It has been developed to facilitate communication and interaction during structured brain-storming sessions involving people of heterogeneous backgrounds [153, 154]. Moreover, the language makes use of graphical symbols that are closely related to the underlying risk concepts, and that are intended to be easily comprehensible.

Regarding c), which is the output of Steps 3 and 4, then our notion of correctness corresponds to internal and construct validity. We have already discussed this in detail in relation to **Success criterion 2**.

**Success criterion 6** The application of the method results in specifications for the deployment and design of indicators that are sufficient for setting up risk monitoring based on indicators.

The method in Paper B (presented in Chapter 10) identifies indicators for measuring likelihood and consequence values that are used in the monitoring of risk values. It seems fair to argue that the application of the method results in specifications for the deployment and design of indicators that are sufficient for setting up risk monitoring based on indicators if the specifications describe:

a) how indicators for measuring likelihood/consequence values should be calculated;

b) how entities (e.g., sensors, humans, etc.) for extracting the data needed in the calculation of indicators should be deployed within the interconnected systems, and
how and where the extracted data should be transmitted; and

c) how risk values should be monitored based on the indicators and other relevant factors.

Regarding a), the indicators’ design specifications describe how the different indicators for measuring likelihood/consequence values should be calculated. Each design specification is given in the form of an algorithm, where the algorithm specifies the data needed for calculating an indicator and how the indicator should be calculated based on the data. For the example case in Paper B, examples of such algorithms are given in Tables 3, 4, 7–9, 12–14, 17, and 18 in Paper B. By implementing the algorithms, the indicators needed for measuring likelihood/consequence values can be calculated.

Regarding b), the indicators’ deployment specifications describe how the data needed in the calculation of indicators should be extracted and transmitted within the interconnected systems. More precisely, the specifications describe the data to be extracted, the deployment of entities for extracting the data, when the data should be extracted, and where and how the extracted data should be transmitted for further processing. For the example case in Paper B, deployment specifications are given in Tables 5, 10, 11, 15, 16, 19, and 20 in Paper B. The deployment specifications serve as a good starting point for deployment of sensors, humans, etc., and for setting up the monitoring infrastructure and the ICT-based work processes needed for extracting and transmitting the data needed in the calculation of indicators.

Regarding c), we can describe how the likelihood/consequence values measured by the indicators should be used in the monitoring of risk values. The risk value of a risk is derived from its likelihood and consequence. Based on likelihoods measured by indicators, we can calculate other likelihoods, including the likelihood of the risk. On the other hand, consequences measured by indicators can be used directly in the calculation of risk values. The rules specified in Appendix D.1 in Paper B are used to specify how risk values should be calculated based on indicators and other relevant factors. Such specifications are used by a risk monitor to calculate risk values. Appendices D.2–D.5 document the application of these rules on the example case in Paper B.

Based on the above arguments for a), b), and c), we conclude that the deployment and design specifications are sufficient for setting up risk monitoring based on indicators.

Success criterion 7 The method is applicable in an industrial context within acceptable effort.

Paper B (presented in Chapter 10) demonstrates the method on a large, realistic example case within the domain of power supply. Domain knowledge on systems with dependencies was used to develop the example case. This knowledge was obtained in the industrial case study that is described in Paper E (presented in Chapter 13). The example case covers in detail all steps of the method. Based on this we may at least claim that the method is applicable in an industrial context given the following two assumptions:

a) the client is able to provide the required data; and

b) the analyst is equal to the method designer (and main author of Paper B).
In the following we argue that assumption a) is justified. In Section 3 in Paper B, we have described the documentation provided by the client during the development of the models/specifications. To develop the target model in Step 1.1, the analyst relies on documentation from the client on the interconnected systems, their service interactions, and the requirements to the different services in the form of required service levels. As can be seen in Figure 10 in Paper B, the different systems and their services are documented at a high level of abstraction in the target model. Requirements to the services may be found in service level agreements or in other relevant documentation. In the target model, it is only necessary to document systems of other parties that the client’s systems interact directly with, but other systems in the environment of the client’s systems may be documented as well if such information is available. The documentation needed for developing the target model is available and easily obtainable within the client’s organization, since only high-level documentation is needed.

To annotate the target model with dependency constructs in Step 1.2, the analyst relies on documentation from the client on how services provided by the client’s systems depend on other services. Detailed documentation is not needed, since dependencies are only documented at a high level of abstraction, as can for instance be seen in Figure 11 in Paper B. It is not necessary to document dependencies for systems of parties that operate in the environment of the client’s systems, but these dependencies may be documented if the relevant information is available. The documentation needed for annotating the target model with dependency constructs is available and easily obtainable within the client’s organization, since only high-level documentation on service dependencies is needed.

To annotate the target model with trust relations in Step 1.3, the analyst relies on trust estimates from the client. These estimates will typically be the result of expert judgments given by domain experts that act on behalf of the client. Each trust estimate is assigned to a required service level of a service provided by an external system to a system of the client. The trust estimate states the degree to which the client trusts the required service level of the service to be delivered. In the client’s organization there will be employees that have knowledge about the services provided by external parties. Thus, the knowledge required for specifying the trust estimates is available.

To perform Steps 2–4, we rely on information provided by domain experts that act on behalf of the client. We require domain experts that have knowledge about the relevant systems and services, the dependencies between the different services, the threat picture of the systems and services of the client, and the ICT infrastructure underlying the different systems. Knowledge about the latter is needed in order to design indicators for ICT-based monitoring. The client is able to provide us with domain experts with the above mentioned knowledge, since the knowledge that we require is essential for the operation of the client’s systems and services.

In the following we argue that assumption b) may be weaken into the analyst being an experienced risk analyst. In fact, we claim that most analysts with some experience in risk analysis and a basic understanding of service dependencies and indicators/metrics would be able to do equally well as the method designer. An important part of every risk analysis is to understand the target to be analyzed. An experienced risk analyst is therefore able to conduct Step 1 of the method, since this step is all about understanding the target to be analyzed. In addition, an experienced risk analyst will also have a basic understanding of service dependencies, since service dependencies are found in numerous risk analysis targets. The experienced risk analyst is therefore able
to document the service dependencies of the target when supported by the domain experts. Moreover, an experienced risk analyst is able to conduct Step 2 of the method, since this step concerns the application of risk analysis. He/she will also have some knowledge about, and quite possibly experience with, risk monitoring. Thus, he/she will have a basic understanding of metrics/indicators. The experienced risk analyst is therefore capable of conducting Step 3 when supported by the domain experts. By interacting with the domain experts, the experienced risk analyst can identify the risks to be monitored and indicators for monitoring these risks. He/she is also capable of specifying the designs and deployments of indicators (Step 4) when supported by the domain experts, since an experienced risk analyst have experience with documentation approaches suitable for conducting such tasks.

The arguments above show that an analyst only needs a basic understanding of service dependencies and indicators/metrics, since he/she is supported by domain experts during the identification and documentation of service dependencies and during the specification of designs and deployments for indicators.

Regarding the effort, consider Table 7.2 estimating the expected effort for the example case in Paper B. We assume that the analyst team consists of two persons; one analysis leader and one analysis secretary. The first leads the meetings with the client, while the second is responsible for documenting the results of the analysis. Moreover, we assume that three domain experts participate on behalf of Client EPP (the client). The three domain experts have knowledge about the relevant systems and services, the dependencies between the different services, the threat picture of the systems and services of the client, and the ICT infrastructure underlying the different systems.

In Table 7.2 we have estimated the expected effort in hours for each step and sub-step of the method for the Analyst (total for the two analysts) and the Client (total for the three domain experts). We estimate both time spent in meetings (M) and time spent between and preparing for meetings (P). Both of the analysts participate in all the steps and sub-steps of the method. This is also true for all of the domain experts.

Step 1.1 is conducted as two meetings of 3 and 2 hours. In the first meeting, one of the domain experts presents the different systems and services of the electrical power production infrastructure (EPP), the public telecom infrastructure (PTI), and the electrical power grid (EPG) that may be of relevance to the analysis. A preparation time of 5 hours should be sufficient for giving this presentation. The domain expert will also provide the two analysts with relevant documentation before the meeting. A preparation time of 5 hours is also reasonable for each of the analysts. The two other domain experts would not need to prepare for the meeting. Based on the presentation, we identify the different systems and services that should be considered in the analysis. We also decide to capture the impact of service dependencies on risk to the quality of each service that Client EPP provides to systems of the PTI and the EPG. Based on this meeting, the analysts develop an initial target model. The two analysts will use 4 hours in total to develop the initial target model. This initial target model is distributed to the three domain experts. In the second meeting, the domain experts identify errors and shortcomings in the initial target model. We also specify the required service levels for the different services documented in the initial target model based on information provided by the domain experts. The result of this meeting is the target model in Figure 10 in Paper B. Based on the size of the complete target model in Figure 10, we expect that each of the domain experts will need 2 hours each to prepare for the meeting. A preparation time of 2 hours is also reasonable for each of the analysts.
Table 7.2: Expected effort (in hours) for the Analyst (total for the two analysts) and the Client (total for the three domain experts) with respect to the example case in Paper B. \(M\) = “time spent in meetings,” while \(P\) = “time spent between and preparing for meetings”

<table>
<thead>
<tr>
<th>Step</th>
<th>Analyst</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(P)</td>
<td>(M)</td>
</tr>
<tr>
<td>Step 1: Document interconnected systems</td>
<td>26</td>
<td>18</td>
</tr>
<tr>
<td>Step 1.1: Model interconnected systems</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Step 1.2: Capture service dependencies</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Step 1.3: Capture trust relations</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Step 2: Analyze the impact of service dependencies on risk to quality of provided services</td>
<td>58</td>
<td>32</td>
</tr>
<tr>
<td>Step 2.1: Identify quality assets</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>Step 2.2: Construct high-level threat diagrams of the impact of service dependencies on identified quality assets</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Step 2.3: Construct detailed threat diagrams of the impact of service dependencies on identified quality assets</td>
<td>46</td>
<td>32</td>
</tr>
<tr>
<td>Step 3: Identify indicators for interconnected systems</td>
<td>30</td>
<td>16</td>
</tr>
<tr>
<td>Step 3.1: Identify risks to be monitored</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Step 3.2: Identify relevant indicators for the risks to be monitored</td>
<td>30</td>
<td>12</td>
</tr>
<tr>
<td>Step 4: Specify design and deployment of identified indicators for interconnected systems</td>
<td>25</td>
<td>12</td>
</tr>
<tr>
<td>Step 4.1: Specify design of indicators for risk monitoring</td>
<td>12.5</td>
<td>6</td>
</tr>
<tr>
<td>Step 4.2: Specify deployment of indicators for risk monitoring</td>
<td>12.5</td>
<td>6</td>
</tr>
<tr>
<td><strong>Total (preparations and meetings)</strong></td>
<td>139</td>
<td>78</td>
</tr>
<tr>
<td><strong>Write the analysis report</strong></td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total (preparations, meetings, and analysis report)</strong></td>
<td>257</td>
<td>160</td>
</tr>
</tbody>
</table>

7.1 Fulfillment of the success criteria
After the meeting has been conducted, we estimate that the analysts will use 2 hours to update the target model.

We conduct both Step 1.2 and Step 1.3 during a 4 hours meeting. For each step, we spend 2 hours. The target model in Figure 10 is distributed to the domain experts before the meeting. The domain experts are asked to identify service dependencies and to come up with trust estimates. Based on the size of the target model in Figure 10, a preparation time of 2 hours is reasonable for each of the domain experts. A preparation time of 2 hours is also reasonable for each of the analysts. For both the analysts and the domain experts, we divide the preparation time between the two steps. During the meeting, we discuss the findings of the domain experts and we annotate the target model with dependency constructs and trust relations. After the meeting has been conducted, we estimate that the analysts use about 2 hours in total to update the target model based on the results from the meeting. The 2 hours are divided between the two steps.

The two analysts conduct Steps 2.1 and 2.2 without the involvement of the domain experts. In Step 2.1, a quality asset is identified for each of the required service levels of the five provided services for which service dependencies’ impact on risk to quality should be captured. We estimate that 2 hours in total are sufficient for identifying and documenting these assets. The results of conducting Step 2.2 are the high-level threat diagrams in Figures 13 and 23–26 in Paper B. Each diagram provides a high-level overview of the impact of service dependencies on risk to quality of a provided service. Moreover, each diagram has been schematically constructed from the target model in Figure 12 by following the procedure described in Section 3.2.2 in Paper B. With the exception of the diagram in Figure 26, it should be straight-forward to schematically construct all of these diagrams from the target model. We estimate that 10 hours, in average 2 hours for each diagram, should be sufficient.

A number of meetings are required for conducting Step 2.3. In the first meeting, the analysts present the results of Steps 2.1 and 2.2 to the domain experts. In this meeting, we also create the likelihood scale in Table 1 in Paper B and the consequence scales in Tables 2 and 6, as well as the risk evaluation criteria in Equations 1 and 2 and in Equations 5–7. We estimate that the first meeting can be conducted within 3 hours. A total preparation time of 2 hours is estimated for the two analysts. The domain experts do not need to prepare for this meeting.

In the other meetings, we detail the high-level threat diagrams in Figures 13 and 23–26 in Paper B. The high-level threat diagrams in Figures 13, 24, and 25 are detailed during the same meeting, since all of these diagrams focus on services provided by the same system. Thus, some of the diagrams resulting from the detailing of these high-level threat diagrams have a lot in common. It should also be noticed that most of the diagrams resulting from the detailing of the high-level threat diagram in Figure 24 also represent a detailing of the high-level threat diagram in Figure 25. To detail the high-level threat diagrams in Figures 13, 24, and 25, we estimate that a six hours meeting should be sufficient. Before the meeting, the analysts create the first version of all the diagrams that should result from the detailing of the high-level threat diagrams. Most of these diagrams will not contain much information. By creating them before the meeting, the time that would otherwise be used for modeling during the meeting can be used for more important tasks. We estimate that the analysts will use 4 hours in total to create the first version of the diagrams in Figures 14–20, 36–42, 45, and 46 in Paper B. After the meeting has been conducted, we estimate that the analysts will
7.1 Fulfillment of the success criteria

use about 16 hours in total to complete the diagrams that were developed during the meeting.

A meeting of 3 hours should be sufficient for detailing the high-level threat diagram in Figure 23. We estimate that the analysts will use about 2 hours in total to create the first version of the diagrams in Figures 27–33 before the meeting. After the meeting has been conducted, we estimate that the analysts will use about 8 hours in total to complete the diagrams that were developed during the meeting.

For the high-level threat diagram in Figure 26, we estimate that a meeting of 4 hours should be sufficient for detailing it. We estimate that the analysts will use about 2 hours in total to create the first version of the diagrams in Figures 47–52, 57, and 58. The diagrams in Figures 53–56 will be created from scratch during the meeting, since these diagrams are the result of detailing the diagram in Figure 52. After the meeting has been conducted, we estimate that the analysts will use about 12 hours in total to complete the diagrams that were developed during the meeting.

We estimate that Step 3.1 can be conducted during a 2-hour meeting. The domain experts should use some time to prepare before this meeting, in order to make correct decisions regarding the risks that should be monitored. A preparation time of 6 hours in total should be sufficient. Two meetings of 3 hours each are necessary for conducting Step 3.2. The first meeting is used for brainstorming ideas on how to monitor the identified risks by the use of indicators. Based on the ideas from the meeting, the analysts create proposals for indicators. These proposals are distributed to the domain experts. In the second meeting, the proposals are discussed and we identify the indicators that should be used for monitoring the risks. To prepare for the two meetings, we estimate that the domain experts will need 10 hours for preparations in total, while the two analysts will need about 15 hours in total to prepare for the meetings and to create the proposals. The diagrams in Figures 21, 34, 35, 43, 44, and 59 in Paper B will be created during and between these meetings. These diagrams are created based on diagrams developed during Step 2.3. As part of this step, the analysts also need to specify how the risk values of the risks should be monitored based on the identified indicators. For the example case in Paper B, this is documented in Appendices D.2–D.5. We estimate that the analysts need to use 15 hours in total to specify how risk values should monitored.

Steps 4.1 and 4.2 will be conducted in parallel. Two meetings of 3 hours each are necessary for conducting the two steps. The first meeting is used for brainstorming ideas on how the indicators should be calculated and how they should be deployed within the ICT-infrastructure of Client EPP. Based on the ideas from the meeting, the analysts develop design and deployment specifications for the different indicators. These specifications are distributed to the domain experts. In the second meeting, the specifications are discussed. Based on the discussion during this meeting, the analysts will correct errors and shortcomings in the specifications after the meeting has been concluded. To prepare for the two meetings, we estimate that the domain experts will need 10 hours for preparations in total, while the two analysts will need about 25 hours in total to prepare for the meetings and to create and update the specifications. The specifications in Tables 3–5 and 7–20 in Paper B will be created during and between these meetings. In total, the estimated time spent by the analysts on Steps 4.1 and 4.2 is 37 hours, while the estimated time spent by the domain experts is 28 hours. In Table 7.2 we have divided the estimated time spent by the analysts and the domain experts equally between Steps 4.1 and 4.2.
After all the meetings have been conducted, the analysts document the analysis and its results in a report. The report will contain the different models/specifications developed during the analysis, and text explaining these models/specifications. We estimate that 40 hours should be sufficient for writing the report. The domain experts will not spend any time in relation to the report writing.

As can be seen in Table 7.2, the estimated effort for the analysts and the three domain experts for the example case in Paper B is 417 hours. To assess whether this estimated effort is acceptable, we compare it with the estimated effort of the industrial case study in Paper E in the same way as we did in relation to Success criterion 4. The industrial case study and the example case have the following things in common:

- Both the industrial case study and the example case address systems that depend on other systems.
- Both the industrial case study and the example case rely heavily on risk analysis by the use of CORAS and the design of indicators.
- Both the industrial case study and the example case involved two analysts and three domain experts.

By using the same arguments as the ones given in relation to Success criterion 4, we come to the conclusion that the estimated effort of the example case in Paper B is acceptable. Based on all of the above discussion, we conclude that the method is applicable in an industrial setting within acceptable effort.

### 7.1.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

**Success criterion 8** The architectural pattern serves as a basis for building monitoring tools based on indicators within a wide range of domains and enterprises.

It seems fair to argue that the architectural pattern in Paper C (presented in Chapter 11) serves as a basis for building monitoring tools based on indicators within a wide range of domains and enterprises if:

a) it is documented in such a way that it is easy to learn, compare, and use;

b) the capabilities that we consider to be core features of enterprise level monitoring tools are general; and

c) the architectural pattern captures all of the core features.

Regarding a) it may of course be discussed what classifies as a good description of a pattern, but in general the description must capture the essence of solving the recurring problem in such a way that the pattern is easy to learn, compare, and use. In [23, 24], pattern templates have been used to achieve such descriptions. The architectural pattern is presented in Paper C by the use of the template described in [24]. This template is very similar to the pattern template by Gamma et al. [23], which is one of the best-known pattern templates.

According to the template in [24], a pattern is described by the use of 14 different categories. Table 7.3 provide a summary of these categories. We have not used all of
7.1 Fulfillment of the success criteria

Table 7.3: Summary of the pattern template in [24]

<table>
<thead>
<tr>
<th>Category</th>
<th>Short description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name and a short summary of the pattern.</td>
</tr>
<tr>
<td>Also Known As</td>
<td>Other names for the pattern, if applicable.</td>
</tr>
<tr>
<td>Example</td>
<td>A real-world example that demonstrates the existence of the problem and the need for the pattern. The example is used throughout the description to illustrate solution and implementation aspects where this is necessary or useful.</td>
</tr>
<tr>
<td>Context</td>
<td>The situations in which the pattern may apply.</td>
</tr>
<tr>
<td>Problem</td>
<td>The problem that the pattern addresses.</td>
</tr>
<tr>
<td>Solution</td>
<td>The fundamental solution principle underlying the pattern.</td>
</tr>
<tr>
<td>Structure</td>
<td>A detailed specification of the structural aspects of the pattern.</td>
</tr>
<tr>
<td>Dynamics</td>
<td>Typical scenarios describing the run-time behavior of the pattern.</td>
</tr>
<tr>
<td>Implementation</td>
<td>Guidelines for implementing the pattern in form of example code. Notice that the guidelines only serve as a suggestion.</td>
</tr>
<tr>
<td>Example Resolved</td>
<td>Discussion of any important aspects for resolving the real-world example that have not been covered in the Solution, Structure, Dynamics, and Implementation categories.</td>
</tr>
<tr>
<td>Variants</td>
<td>A brief description of variants or specializations of the pattern.</td>
</tr>
<tr>
<td>Known Uses</td>
<td>Examples of the pattern in use, taken from existing systems.</td>
</tr>
<tr>
<td>Consequences</td>
<td>The benefits that the pattern provide and any potential liabilities.</td>
</tr>
<tr>
<td>See Also</td>
<td>References to patterns that solve similar problems and to patterns that help us to refine the pattern we are describing.</td>
</tr>
</tbody>
</table>
these categories, since not all of them are relevant for describing our pattern. In the following we describe how we applied the template in Paper C, and we argue for the exclusion of certain categories.

We cover most of the template categories in Section II (Architectural Pattern) of Paper C. For some categories, however, we found it more suitable to cover them in other sections. For instance, the two categories “Context” and “Problem” are covered as part of Section I (Introduction), while the category “Example” is covered as part of Section III (Demonstration of Architectural Pattern). In the case of the “Example” category, it was not necessary to illustrate different aspects of the pattern during its presentation in Section II. This also means that we found the category “Example Resolved” redundant. Moreover, we also found the categories “Also Known As” and “Variants” redundant, since the pattern is not known by any other names and since there are no other variants or specializations of the pattern. It should also be noticed that we do not explicitly document the “Known Uses” category in the paper. On the other hand, we refer several times to the risk monitor that the architectural pattern is a generalization of. For instance, in Section III we show that the risk monitor is an instance of the pattern.

All of the other categories are covered in Section II. In the case of the categories “Name” and “See Also,” we used the names “Name and short summary” and “Related patterns” instead. We found these names to be more descriptive than the category names given in [24]. In the case of the “Structure” category, Buschmann et al. [24] recommends the use of Class-Responsibility-Collaborator (CRC) cards [155] and OMT class diagrams [156] to describe the structural aspects of the pattern. On the other hand, in case of the “Dynamics” category, the authors of [24] recommend the use of Object Message Sequence Charts\(^1\) to describe the run-time behavior of the pattern. To document relations between the components of the pattern we have used a UML [94] class diagram instead of an OMT class diagram, while CRC cards have been used to document the responsibilities of each component of the pattern. Moreover, we have used UML sequence diagrams instead of Object Message Sequence Charts to document typical scenarios that describes the run-time behavior of the pattern, while in the case of the “Implementation” category, we have used Java for the example code. These modeling/programming languages, and especially UML and Java, are well-known to software developers, and therefore suitable for describing different aspects of the pattern.

Regarding b) as previously explained, the architectural pattern is a generalization of the architecture of the CORAS risk monitor [36] that was developed in the MASTER [37] research project. The risk monitor has a number of features that makes it well suited for dynamic risk monitoring in the context of MASTER [36]. In particular, likelihoods and consequences in the CORAS threat diagrams (the risk picture) can be expressed by indicators that are dynamically updated during run-time. In the MASTER framework, the CORAS risk monitor has been successfully integrated in the part of the framework that provides functionality for monitoring the execution of business and control processes, as well as indicator values and risk levels. The risk monitor exhibits a number of features that are not specific to the monitoring of risks, but general to a broad class of enterprise level monitoring tools. The features are as follows:

\(^1\)The authors of [24] own adaption of Message Sequence Charts [157].
7.1 Fulfillment of the success criteria

1. Collect low-level data from the ICT infrastructure or similar.

2. Aggregate the collected low-level data.

3. Evaluate the aggregated data.

4. Present the aggregated data and the evaluation results to different types of enterprise users.

5. Present the most recent aggregated data and evaluation results.

6. Configure the tool with respect to:
   a. The low-level data that should be collected.
   b. How the low-level data should be aggregated into information that is relevant and useful.
   c. How aggregated data should be evaluated.
   d. The kind of aggregated data and evaluation results that should be presented and how this should be made comprehensible to different types of enterprise users.

In the following we argue that these features are general. We consider monitoring tools that operate based on an ICT infrastructure or other types of ICT-based systems. When designing such a tool we need to take into consideration the specific characteristics of the infrastructure/system on which the tool is based. The data generated at the ICT infrastructure level is typically of a low-level nature, e.g., events such as service calls or responses. At the enterprise level, an individual event has often little significance when considered in isolation. In order to make sense of the events, they need to be collected and aggregated into high-level information that is both useful and relevant at the enterprise level.

In an enterprise setting, monitoring is typically used for evaluation. An enterprise may want to evaluate whether it complies with requirements from laws and regulations, whether risks are at an acceptable level, or whether it fulfills business objectives. Thus, an enterprise level monitoring tool needs to evaluate the aggregated data. Moreover, the various enterprise users have different information needs and different preferences for how information should be presented. For instance, the chief security officer may want a high-level assessment of the information security risk to which the company is exposed, while a security engineer may want to know how often a port in a firewall is open.

It is almost needless to say, but when users of an enterprise level monitoring tool are presented with updates of aggregated data and evaluation results, they should always be presented with the most recent aggregated data and evaluation results. Moreover, an enterprise level monitoring tool will often experience changes with respect to the collection of low-level data, the aggregation of low-level data, the evaluation of aggregated data, and the presentation of aggregated data and evaluation results during its life-time. Thus, it must be possible to configure the tool.

Regarding c), the architectural pattern divides an enterprise level monitoring tool into three components: MonitorModel, which contains the core monitoring functionality and data; MonitorConsole, which presents aggregated data and evaluation results.
in a specific way to a group of users; and MonitorConfig, which is used to set up the enterprise level monitoring tool and to configure it during run-time. If the enterprise level monitoring tool needs to present aggregated data and evaluation results to more than one group of users, then the tool will have more than one MonitorConsole.

The MonitorModel collects relevant low-level data in the form of basic indicators from a component referred to as DataSource in Paper C. This component is part of the pattern’s environment. The actual collection of low-level data from the ICT infrastructure or similar, for instance by the use of sensors, is outside the scope of the pattern. The MonitorModel aggregates the basic indicators into composite indicators. Thus, MonitorModel captures features 1 and 2. Moreover, the MonitorConsoles retrieve the most recent updated composite indicators from MonitorModel, evaluate them if needed, and update the displays used by their users based on the composite indicators and evaluation results. Thus, the MonitorConsoles capture features 3, 4, and 5. The MonitorModel and the MonitorConsoles are configured by the MonitorConfig before run-time and during run-time. The MonitorModel is configured with respect to 6a and 6b, while the MonitorConsoles are configured with respect to 6c and 6d. Thus, MonitorConfig captures features 6 a–d.

Based on the above arguments for a), b), and c), we conclude that the architectural pattern can be used as a starting point for building specialized monitoring tools within various kinds of domains and enterprises.

**Success criterion 9** The architectural pattern facilitates modularity and reuse.

ISO/IEC/IEEE 24765 [17] defines modularity as “the degree to which a system or computer program is composed of discrete components such that a change to one component has minimal impact on other components,” while it defines reuse as “building a software system at least partly from existing pieces to perform a new application.” Moreover, it defines reusable as “pertaining to a software module or other work product that can be used in more than one computer program or software system.”

It seems fair to argue that the architectural pattern in Paper C (presented in Chapter 11) facilitates modularity and reuse if:

- changes to a MonitorModel, MonitorConsole, or MonitorConfig has minimal impact on the other components in the enterprise level monitoring tool; and

- the MonitorModel, MonitorConsoles, and MonitorConfig developed for one enterprise level monitoring tool can at least be partly reused in another enterprise level monitoring tool.

Regarding a), our architectural pattern is closely related to the Model-View-Controller (MVC) pattern [24,120], since this pattern was used as inspiration when we designed our pattern. As previously explained in Section 4.4.3, MVC divides an interactive application into three main components: model, view, and controller. The model encapsulates core data and functionality, while views and controllers together comprise the user interface of the application. The model is independent of the user interface. The same is also true for our pattern. As described in Section II in Paper C, the pattern separates user interfaces (MonitorConsoles) from the component handling core monitoring data and functionality (MonitorModel). Thus, changes to the implementations
of MonitorConsoles will not require changes to the implementation of the MonitorModel. Also, the introduction of new MonitorConsoles during run-time has no effect on the implementation of the MonitorModel.

The MonitorConfig is used to configure the MonitorModel and the MonitorConsoles with a data configuration file and presentation models, respectively. There is a close coupling of MonitorModel to MonitorConfig. The MonitorModel depends on the specific language in which its data configuration file is expressed. Change of language may most likely require changes to the implementation of MonitorModel. Similar, a MonitorConsole may depend on the specific language in which its presentation model is expressed. Changes to this language may require changes to the implementation of the MonitorConsole. However, changes to the language are considered to be quite rare in both cases.

Both the MonitorConsoles and the MonitorConfig make direct calls to the MonitorModel. In addition, the MonitorConfig makes direct calls to the MonitorConsoles. This means that changes to the interface of MonitorModel will break the code of both the MonitorConsoles and the MonitorConfig, while changes to the interfaces of MonitorConsoles will break the code of the MonitorConfig. Such changes to the interfaces will however be quite rare. It should be noticed that MonitorModel does not make direct calls to MonitorConsoles or MonitorConfig. The interaction between MonitorConfig and MonitorModel is one-way only, while it interacts with MonitorConsoles through Observer objects. The latter is due to the use of a change-propagation mechanism, which is implemented by the use of the Publisher-Subscriber design pattern [24].

A MonitorConsole depends on a number of the MonitorModel’s composite indicators. A re-configuration of the MonitorModel may result in the removal of composite indicators that one or more MonitorConsoles depend on. If this happens, then the MonitorConfig needs to re-configure the affected MonitorConsoles. If a MonitorConsole is re-configured with a presentation model that is parameterized with new composite indicators, then the MonitorConsole needs to notify the MonitorModel that it wants to receive updates for the new ones.

Regarding b), if the same language is used to create all the data configuration files, then it is possible to implement a generic MonitorModel, which becomes specialized when configured. Two MonitorModels that are based on the same generic MonitorModel will only be different with respect to: the data sources that they retrieve basic indicators from; the basic indicators that they retrieve; the composite indicators that result from the aggregation of their basic indicators; the MonitorConsoles that depend on their composite indicators; and the MonitorConfigs that configure them. In the case of different languages, the majority of the code base of one MonitorModel can be used in the implementation of another.

A MonitorConfig has functionality for creating and updating data configuration files and presentation models, and for configuring MonitorModels and MonitorConsoles. The data configuration files are created and updated by the use of some text-based editor, while each presentation model is created and updated either by the use of a text-based editor or a graphical modeling tool. Both a text-based editor and a graphical modeling tool can often be extended to handle new languages. Thus, it may be possible to use a MonitorConfig in another monitoring tool where support for other languages is required. If it is not possible to extend the MonitorConfig, then at least some of its code base may be reused.

The reuse of a MonitorConsole depends on how tightly integrated its presentation
model is with the rest of the component. If the component only updates the composite indicators that the presentation model is parameterized with, then it may be the case that the presentation model is easily replaceable by another. On the other hand, if the component performs other types of updates of the model or is heavily involved in the evaluation of the composite indicators, then it may be more difficult to reuse the component. If it is not possible to reuse the MonitorConsole, then at least some of its code base may be reused.

Based on the above arguments for a) and b), we conclude that the architectural pattern facilitates modularity and reuse.

**7.1.4 Framework for analyzing and monitoring the impact of dependencies on quality**

**Success criterion 1** The framework fulfills its intended purpose.

As explained in Section 2.2.1, the purpose of the framework is to: (1) analyze the impact of service dependencies on quality of services; and (2) support the set-up of monitoring of the impact of service dependencies on quality of services.

In Section 5.1 we have described the integration of the three artifacts. The framework takes as input a business objective that focus on the achievement of service quality. With the relevant part of business being interconnected systems that depend on each other through service interactions, some of the steps of the method of Artifact 1 are replaced by steps of the method of Artifact 2 in order to document the interconnected systems and to capture the impact of service dependencies on the relevant service quality.

Artifact 2 can be used to analyze and capture the impact of service dependencies on risk to the fulfillment of the business objective that focus on the achievement of service quality. The artifact is therefore an answer to (1). By using Artifact 2, we can also identify indicators for monitoring all risks that are unacceptable with respect to the business objective.

Artifact 1 is used to specify deployments of sensors within the interconnected systems, where the sensors are needed for gathering the data necessary for calculating the indicators. It is also used to specify requirements to the indicators with respect to the sensor deployments. By using Artifacts 1 and 2, we can specify how the indicators should be calculated and how the calculations should be embedded within the interconnected systems. Artifact 1 is also used to evaluate the validity of the designed indicators with respect to the relevant service quality.

The output from the integration of Artifacts 1 and 2 is a set of indicators that is valid with respect to the relevant service quality. These indicators can be used to monitor risks to the fulfillment of the business objective. By using Artifact 3, we can implement a risk monitor based on the designed indicators. With respect to (2), Artifact 2 is partly an answer, while Artifacts 1 and 3 are the main answer.

The integration as described in Section 5.1 is feasible. Moreover, the evaluation of the fulfillment of the success criteria for Artifacts 1–3 in Sections 7.1.1–7.1.3 shows that the three artifacts fulfill their intended purposes. Based on the above discussion and the results from the evaluation of Artifacts 1–3, we conclude that the framework also fulfills its intended purpose.
7.2 How the artifacts relate to and extend the state of the art

In this section we discuss how the three artifacts relate to and extend the state of the art presented in Chapter 4. We structure the discussion into three sub-sections; one for each of the artifacts. The reason why such an discussion was not conducted already in Chapter 4 is that the different artifacts were first presented in detail in Chapter 5. For further comparison of our artifacts to the state of the art, the reader is referred to the related work sections of the papers presented in Part II of this thesis.

7.2.1 Artifact 1: Method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators

Section 4.1 presents different approaches related to the use, design and deployment, and validation of indicators. GQM [54,55] and GQ(I)M [56,57] are two approaches that are that are closely related to our method. What makes GQM and GQ(I)M different from our method is that they do not put the same emphasis on the validation of metrics/indicators. For both GQM and GQ(I)M, no specific method, beyond reviews, is specified for validating the metrics/indicators. Our method and the two approaches have in common that all three specify the design and deployment of metrics/indicators. In [55], Solingen and Berghout provide instructions for how to document metrics when using the GQM approach, while in [57], Goethert and Siviy provide a template that can be use to document the construction and use of indicators when using GQ(I)M.

The frameworks presented in [58,59] by Popova and Sharpsanskykh are also related to our method. In [58,59], the relationships between goals and performance indicators are made explicit. Moreover, the frameworks use a formal approach to model organizational goals based on performance indicators. The frameworks also use mechanisms for establishing goal satisfaction and for checking consistency of and correspondence between the goal and the performance indicator structures. Popova and Sharpsanskykh do not evaluate the validity of the performance indicators. What can be said, however, is that the modeling results in a hierarchical goal structure that can be used during the evaluation of internal validity. Based on [58,59], we have not identified any specifications/models that can be used during the evaluation of construct validity.

Our research on how to validate indicators has been inspired by research on the validation of software engineering metrics. This research has resulted in many criteria for evaluating metrics, where many of them are not specific to software engineering. Based on the systematic literature review of Meneely et al. [63] of papers focusing on validation of software engineering metrics, we selected a number of criteria for evaluating the validity of the indicators that our method designs.
7.2.2 Artifact 2: Method for capturing and monitoring the impact of service dependencies on the quality of provided services

Artifact 2 is a specialization of the approach presented in [105] by Refsdal and Stølen. The approach in [105] is general in the sense that it only restricts the risk identification to the identified assets and nothing else. In our approach, the risk identification focuses entirely on risk to quality of provided services that have been caused by service dependencies. The approach in [105] can of course be used to identify indicators for the purpose of measuring the impact of service dependencies on risk to quality of provided services, because of its generality. Compared to our approach, however, it is inferior. The approach in [105] does not offer any support for dealing with service dependencies. In addition, it focuses to a much lesser extent on the calculations of indicators, and it cannot be used to specify how the indicator calculations should be deployed in the systems to be monitored.

Section 4.2.2 presents approaches for the modeling and analysis of service dependencies. Our graph-based approach to the modeling of service dependencies is similar to other approaches within critical infrastructure protection (see e.g., [84–87]). Our modeling approach also supports the modeling of trust relations in the case of third-party service dependencies. These relations are captured in terms of probabilities. In our method, trust is used to estimate likelihoods of third-party services failing to be delivered according to their requirements. The trust approaches presented in [131–133,135] can be used to address a similar problem. All these approaches can be used to assess service providers’ ability to provide services with the expected quality. The main difference between our approach to trust and the trust approaches presented in [131–133,135] is that the latter approaches focus on dynamic aspects of trust. These dynamic trust approaches also differ from our approach in that they are primarily used for selecting appropriate interaction partners, for instance in a peer-to-peer network.

Section 4.3.2 presents approaches for analysis of risk in the context of dependencies. In [92,93], Giese et al. present a method for compositional hazard analysis of components by the use fault trees [95]. For each component, incoming, outgoing, and internal failures are described, as well as the dependencies between the different failures. The dependency information is used to capture the propagation of failures by combining failure information of the different components. This method differs from our method in many respects. First, fault trees cannot be used to address mutual dependencies. Second, the method of Giese et al. is limited to failures caused by software and/or hardware. Thus, human failures, either accidental or deliberate, cannot be addressed. Both mutual dependencies and human failures can be addressed by the use of our method. In [96] and [99], other approaches to component-based hazard analysis are presented. These approaches do also apply fault trees. Thus, they lack the ability to address mutual dependencies.

7.2.3 Artifact 3: Architectural pattern for constructing enterprise level monitoring tools based on indicators

Section 4.4.3 presents state of the art on patterns and the implementation of monitoring tools. The Model-View-Controller (MVC) pattern [24,120] was used as inspiration when we designed our pattern. Both patterns separate the core data and functional-
ity from the user interfaces. In addition, both patterns use the same mechanism for reflecting changes in the core data in the user interfaces. More precisely, the Publisher-Subscriber design pattern [24] is used to implement this change-propagation mechanism. One of the main differences between our pattern and MVC is that all the user interfaces in MVC display the same core data, while in our pattern the MonitorConsoles can display different core monitoring data, which means that they need to be updated differently. Another difference is that we have replaced the controller component in MVC with the MonitorConfig component. In MVC, each controller is used for handling user input for a view. The MonitorConfig component on the other hand is used for configuring the MonitorModel and the MonitorConsoles before and during run-time by the use of configuration files.

Approaches like [121] and [123] differ from our architectural pattern in that they address specific monitoring problems within specific domains. Our architectural pattern, on the other hand, may be used to build monitoring tools within a wide range of domains and enterprises. The pattern presented in [121] do also differ from our pattern in that it focus on the sensors collecting the information needed in the monitoring. The construction of the sensor infrastructure is outside the scope of our pattern.

The tool framework called Mozart [122] uses a model driven approach to create monitoring applications that uses key performance indicators (KPIs). Mozart differs from our pattern in that it focuses both on the aggregation of already existing indicators and on the transformation of these indicators into a monitoring application. This means that Mozart both design and deploy indicators. Our pattern, on the other hand, is only concerned with the deployment of indicators through the implementation of a monitoring tool.
Chapter 8

Conclusion

This chapter concludes Part I of the thesis by summarizing the achievements and by outlining directions for future work.

8.1 What has been achieved

The main contributions of this thesis are three new artifacts

1. a method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators;

2. a method for capturing and monitoring the impact of service dependencies on the quality of provided services; and

3. an architectural pattern for constructing enterprise level monitoring tools based on indicators.

These artifacts may be integrated into a framework for analyzing and monitoring the impact of service dependencies on quality of services. We have argued that the framework is

1. well-suited to analyze the impact of service dependencies on quality of services;

2. well-suited to support the set-up of monitoring of the impact of service dependencies on quality of services; and

3. applicable in an industrial context within acceptable effort.

Artifact 2 can be used to analyze the impact of service dependencies on quality of services. The artifact is therefore an answer to part 1 of the overall objective (presented above). We are not aware of any other approaches that have the same capabilities as Artifact 2.

In the framework, Artifacts 1 and 2 are used to design indicators for monitoring the impact of service dependencies on quality of services. Artifact 1 is also used evaluate the validity of the designed indicators with respect to the relevant service quality. Moreover, Artifact 3 can be used to implement a monitoring tool that deploys the indicators designed by Artifacts 1 and 2. Artifact 2 is partly an answer to part 2 of the overall objective, while Artifacts 1 and 3 are the main answer. We are not aware...
of any other approaches that have the same capabilities as Artifact 1. The same may also be said about Artifact 3.

The evaluation of the success criteria in Section 7.1 shows that Artifacts 1 and 2 are applicable in an industrial context within acceptable effort. Based on this, Artifacts 1 and 2 are the answer to part 3 of the overall objective.

This thesis also contributes in terms of

1. an empirical study on trust-based decisions in interconnected systems; and
2. an empirical study on the design of indicators for monitoring risk.

These industrial case studies were mainly carried out to support the development of the artifacts, but since they also provide insight into issues of a more general nature, they may be seen as contributions on their own.

8.2 Directions for future work

There are a number of directions for future work. An obvious direction for future work is to apply the framework and the artifacts in an industrial setting to gather more empirical evidence on their applicability and to assess their performance in a practical setting.

As already mentioned during the evaluation of Success criterion 5 in Section 7.1.2, the method in Paper B (presented in Chapter 10) does not take into account the strengths of service dependencies. A potential generalization of our method is therefore to put weights on the service dependencies in order to capture their strengths. By taking this into account we will get a more correct risk picture of the impact of service dependencies on the quality of provided services. Moreover, one way to improve the correctness of estimates, e.g., likelihood estimates, is to measure the same thing in different ways. This is especially relevant when it comes to monitoring by the use indicators. We can for instance use several independent indicators to monitor the same likelihood value. The possibilities and challenges with respect to using several independent indicators are something that we want to investigate further.

Another direction for future work is to provide tool support for Step 1 and Step 2.2 of the method in Paper B. Step 1 concerns the creation of the target model, i.e., the modeling of interconnected systems, services, service dependencies, etc., while Step 2.2 concerns the schematic construction of high-level risk models of the impact of service dependencies on risk to the quality of provided services from the target model. A tool should be able to construct these high-level risk models automatically from the target model by following the schematic procedure specified in Section 3.2.2 in Paper B.

An interesting direction for future work is to combine the framework with approaches from the critical infrastructure domain that have simulation capabilities. By using an approach such as [86] we can for instance run simulations on a graph-based model of interconnected systems and investigate how the functionality of systems and services changes when nodes are removed. In this way, the most critical systems of the graph can be identified. The framework can then be applied for these systems to analyze and monitor the impact of service dependencies on the quality of the critical services they provide.

Another interesting direction for future work is the use of a dynamic trust approach for reasoning about third-party service dependencies. In the method in Paper B, trust
is used to estimate likelihoods of third-party services failing to be delivered according to their requirements. Dynamic trust approaches (e.g., [131–133, 135]) can be used to capture the change in trust over time. By using a dynamic trust approach we can update the initial trust estimates based on past behavior of third-party services and other factors. The likelihoods estimated based on trust estimates can be expressed by the use of indicators. The likelihoods can then be monitored, and changes in trust can be reflected in the risk picture.
Bibliography


[23] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1st ed., 1994.


Bibliography


Bibliography


Part II

Research Papers
Chapter 9

Paper A: ValidKI: A method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators
Report

ValidKI: A Method for Designing Indicators to Monitor the Fulfillment of Business Objectives with Particular Focus on Quality and ICT-supported Monitoring of Indicators

Author(s)
Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen
ValidKI: A Method for Designing Indicators to Monitor the Fulfillment of Business Objectives with Particular Focus on Quality and ICT-supported Monitoring of Indicators

ABSTRACT
In this report we present our method ValidKI for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. A set of indicators is valid with respect to a business objective if it measures the degree to which the business or relevant part thereof fulfills the business objective. ValidKI consists of six main steps. We demonstrate the method on an example case focusing on the use of electronic patient records in a hospital environment.
CONTENTS

I Introduction 4

II Basic terminology and definitions 5
   II-A The artifacts addressed by ValidKI 5
   II-B The models/descriptions developed by ValidKI 6
   II-C Validity 6

III Overview of ValidKI 7
   III-A Establish target 7
   III-B Identify risks to fulfillment of business objective 7
   III-C Identify key indicators to monitor risks 8
   III-D Evaluate internal validity 8
   III-E Specify key indicator designs 9
   III-F Evaluate construct validity 9

IV Establish target 9
   IV-A Express business objectives more precisely (Step 1.1 of ValidKI) 10
   IV-B Describe relevant part of business (Step 1.2 of ValidKI) 10

V Identify risks to fulfillment of business objective 13
   V-A Specify risk acceptance criteria (Step 2.1 of ValidKI) 13
   V-B Risk identification and estimation (Step 2.2 of ValidKI) 13
   V-C Risk evaluation (Step 2.3 of ValidKI) 16

VI Identify key indicators to monitor risks 19
   VI-A Deploy sensors to monitor risks (Step 3.1 of ValidKI) 19
   VI-B Specify requirements to key indicators wrt deployed sensors (Step 3.2 of ValidKI) 21

VII Evaluate internal validity 23
   VII-A Express business objective in terms of key indicators (Step 4.1 of ValidKI) 23
   VII-B Evaluate criteria for internal validity (Step 4.2 of ValidKI) 24

VIII Specify key indicator designs 25
   VIII-A Key indicator designs for $K_{PR,SP,EPR,INFO}$ and its basic key indicators 25
   VIII-B Key indicator designs for $K_{PR,HSP,EPR,INFO}$ and its basic key indicators 26
   VIII-C Key indicator designs for $K_{NOT,APP,UNAUTH,ACC}$ and its basic key indicators 28
   VIII-D Key indicator designs for $K_{SP,EPR,INFO}$ and its basic key indicators 28
   VIII-E Key indicator designs for $K_{HSP,EPR,INFO}$ and its basic key indicators 31
   VIII-F Key indicator designs for $K_{ILL,ACC,SC}$ and its basic key indicators 31

IX Evaluate construct validity 38

X Related work 44

XI Conclusion 45

References 45
ValidKI: A Method for Designing Indicators to Monitor the Fulfillment of Business Objectives with Particular Focus on Quality and ICT-supported Monitoring of Indicators

Olav Skjelkvåle Ligaarden*, Atle Refsdal*, and Ketil Stølen†

* Department for Networked Systems and Services, SINTEF ICT
PO Box 124 Blindern, N-0314 Oslo, Norway
E-mail: {olav.ligaarden, atle.refsdal, ketil.stolen}@sintef.no
† Department of Informatics, University of Oslo
PO Box 1080 Blindern, N-0316 Oslo, Norway

Abstract

In this report we present our method ValidKI for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. A set of indicators is valid with respect to a business objective if it measures the degree to which the business or relevant part thereof fulfills the business objective. ValidKI consists of six main steps. We demonstrate the method on an example case focusing on the use of electronic patient records in a hospital environment.

Keywords

Indicator, key indicator, business objective, quality, ICT-supported monitoring, electronic patient record

I. INTRODUCTION

Today's companies benefit greatly from ICT-supported business processes, as well as business intelligence and business process intelligence applications monitoring and analyzing different aspects of a business and its processes. The output from these applications may be indicators which summarize large amounts of data into single numbers. Indicators can be used to evaluate how successful a company is with respect to specific business objectives. For this to be possible it is important that the indicators are valid. A set of indicators is valid with respect to a business objective if it measures the degree to which the business or relevant part thereof fulfills the business objective. Valid indicators facilitate decision making, while invalid indicators may lead to bad business decisions, which again may greatly harm the company.

In today's business environment, companies cooperate across company borders. Such co-operations often result in sharing or outsourcing of ICT-supported business processes. One example is the interconnected electronic patient record (EPR) infrastructure. The common goal for this infrastructure is the exchange of EPRs facilitating the treatment of the same patient at more than one hospital. In such an infrastructure, it is important to monitor the use of EPRs in order to detect and avoid misuse. This may be achieved through the use of indicators. It may be challenging to identify and compute good indicators that are valid with respect to business objectives that focus on quality in general and security in particular. Furthermore, in an infrastructure or system stretching across many companies we often have different degrees of visibility into how the cooperating parties perform their part of the business relationship, making the calculation of indicators particularly hard.

In [1] we presented the method ValidKI (Valid Key Indicators) for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. ValidKI facilitates the design of a set of indicators that is valid with respect to a business objective. In this report we present an improved version of the method.

We demonstrate ValidKI by applying it on an example case targeting the use of EPRs. We have developed ValidKI with the aim of fulfilling the following characteristics:
• **Business focus:** The method should facilitate the design and assessment of indicators for the purpose of measuring the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators.

• **Efficiency:** The method should be time and resource efficient.

• **Generality:** The method should be able to support the design and assessment of indicators based on data from systems that are controlled and operated by different companies or organizations.

• **Heterogeneity:** The method should not place restrictions on how indicators are designed.

The rest of the report is structured as follows: in Section II we introduce our basic terminology and definitions. In Section III we give an overview of ValidKI and its six main steps. In Sections IV – IX we demonstrate our six-step method on an example case addressing the use of EPRs in a hospital environment. In Section X we present related work, while in Section XI we conclude by characterizing our contribution and discussing the suitability of our method.

## II. Basic terminology and definitions

Hammond et al. defines indicator as “something that provides a clue to a matter of larger significance or makes perceptible a trend or phenomenon that is not immediately detectable” [2]. For example, a drop in barometric pressure may signal a coming storm, while an unexpected rise in the traffic load of a web server may signal a denial of service attack in progress. Thus, the significance of an indicator extends beyond what is actually measured to a larger phenomenon of interest.

Indicators are closely related to metrics. ISO/IEC/IEEE 24765 [3] defines metric as “a quantitative measure of the degree to which a system, component, or process possesses a given attribute,” while it defines attribute as “the specific characteristic of the entity being measured.” For the web server mentioned above, an example of an attribute may be availability. An availability metric may again act as an indicator for denial of service attacks, if we compare the metric with a baseline or expected result [4]. As we can see, metrics are not that different from indicators. For that reason, indicators and metrics are often used interchangeably in the literature.

Many companies profit considerably from the use of indicators [5] resulting from business process intelligence applications that monitor and analyze different aspects of a business and its processes. Indicators can be used to measure to what degree a company fulfills its business objectives and we then speak of key indicators. Some business objectives may focus on business performance, while others may focus on risk or compliance with laws and regulations. We will in the remainder of the report refer to indicators as key indicators, since we focus on indicators in the context of business objectives.

### A. The artifacts addressed by ValidKI

The UML [6] class diagram in Fig. 1 relates the main artifacts addressed by ValidKI. The associations between the different concepts have cardinalities that specify how many instances of one concept that may be associated to an instance of the other concept.

As characterized by the diagram, one or more key indicators are used to measure to what extent a business objective is fulfilled with respect to a relevant part of the business. Each key indicator is calculated based on data provided by one or more sensors. The sensors gather data from the relevant part of the business. A sensor may gather data for more than one key indicator.
A specification, at a suitable level of abstraction, documents the relevant part of the business in question. Business objectives are typically expressed at an enterprise level and in such a way that they can easily be understood by for example shareholders, board members, partners, etc. It is therefore often not completely clear what it means to fulfill them. This motivates the need to capture each business objective more precisely.

The fulfillment of a precise business objective may be affected by a number of risks. We therefore conduct a risk analysis to capture risk to the fulfillment of the precise business objective. To evaluate which risks that are acceptable and not acceptable with respect to the fulfillment of the precise business objective, we use risk acceptance criteria. It is the risks that are not acceptable that we need to monitor. The acceptable risks may be thought of to represent uncertainty we can live with. In other words, their potential occurrences are not seen to significantly influence the fulfillment of the business objective.

The degree of fulfillment of a precise business objective is measured by a set of key indicators. To measure its degree of fulfillment there is a need to express each precise business objective in terms of key indicators. We refer to this reformulation as the reformulated precise business objective. Moreover, the correctness of key indicators will be affected if they are not implemented correctly. This may again lead to new unacceptable risks that affect the fulfillment of the precise business objective. Since the reformulated precise business objective is the precise business objective expressed in terms of key indicators, we need to analyze risks to the correctness of the reformulated precise business objective.

The computation of key indicators relies on different kinds of data. To collect the data, sensors need to be deployed in the relevant part of business. Thus, there is a need to specify the deployment of different sensors.

For each key indicator we distinguish between two specifications: the key indicator requirements specification and the key indicator design specification. The first captures requirements to a key indicator with respect to the sensor deployment specifications, while the second defines how the key indicator should be calculated.

C. Validity

ISO/IEC 9126 defines validation as “confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled” [7]. Since an indicator is basically a metric that can be compared to a baseline/expected result, the field of metric validation is highly relevant. There is however no agreement upon what constitutes a valid metric [8]. In [8], Meneley et al. present a systematic literature review of papers focusing on validation of software engineering metrics. The literature review began with 2288 papers, which were later reduced to 20 papers. From these 20 papers, the authors extracted and categorized 47 unique validation
Input: A business objective

Step 1: Establish target
Step 1.1: Express business objectives more precisely
Step 1.2: Describe relevant part of business

Step 2: Identify risks to fulfillment of business objective
Step 2.1: Specify risk acceptance criteria
Step 2.2: Risk identification and estimation
Step 2.3: Risk evaluation

Step 3: Identify key indicators to monitor risks
Step 3.1: Deploy sensors to monitor risks
Step 3.2: Specify requirements to key indicators wrt deployed sensors

Step 4: Evaluate internal validity
Step 4.1: Express business objective in terms of key indicators
Step 4.2: Evaluate criteria for internal validity

Step 5: Specify key indicator designs

Step 6: Evaluate construct validity

Output: A set of key indicators and a report arguing its validity with respect to the business objective received as input

Fig. 3. Overview of ValidKI

Criteria. The authors argue that metric researchers and developers should select criteria based on the intended usage of the metric. Even though the focus in [8] is on validation of software engineering metrics, a number of the validation criteria presented are general, thus not specific to software engineering. In particular, following [8] we define a set of key indicators to be valid with respect to a business objective if it is valid in the following two ways:

1) **Internal validity** – the precise business objective expressed in terms of the key indicators correctly measures the degree to which the business objective is fulfilled; and
2) **Construct validity** – the gathering of the sensor measurements of each key indicator is suitable with respect to its requirements specification.

III. Overview of ValidKI

Fig. 3 provides an overview of the ValidKI method. It takes as input a business objective and delivers a set of key indicators and a report arguing its validity with respect to the business objective received as input. When using ValidKI in practice we will typically develop key indicators for a set of business objectives, and not just one which we restrict our attention to here. It should be noticed that when developing key indicators for a set of business objectives, we need to take into account that key indicators (i.e., software or infrastructure) developed for one business objective may affect the validity of key indicators developed for another.

In the following we offer additional explanations for each of the six main steps of the ValidKI method.

A. Establish target

The first main step of ValidKI is all about understanding the target, i.e., understanding exactly what the business objective means and acquiring the necessary understanding of the relevant part of business for which the business objective has been formulated. We distinguish between two sub-steps. In the first sub-step we characterize the business objective more precisely by formulating constraints that need to be fulfilled. In the second sub-step we specify the relevant part of the business.

B. Identify risks to fulfillment of business objective

The second main step of ValidKI is concerned with conducting a risk analysis to identify risks to the fulfillment of the business objective. We distinguish between three sub-steps. In the first sub-step the risk acceptance criteria are
specified. The criteria classify a risk as either acceptable or unacceptable based on its likelihood and consequence. In the second sub-step we identify how threats may initiate risks. We also identify vulnerabilities and threat scenarios leading up to the risks, and we estimate likelihood and consequence. During the risk analysis we may identify risks that pull in the same direction. Such risks should be combined into one risk. The individual risks may be acceptable when considered in isolation, while the combined risk may be unacceptable. In the third sub-step we evaluate the identified risks with respect to the specified risk acceptance criteria.

C. Identify key indicators to monitor risks

The third main step of ValidKI is concerned with identifying key indicators to monitor the unacceptable risks identified in the previous step. We distinguish between two sub-steps. In the first sub-step we specify how sensors should be deployed in the relevant part of business. The key indicators that we identify are to be calculated based on data gathered by the sensors. In the second sub-step we specify our requirements to the key indicators with respect to the deployed sensors. The two sub-steps are typically conducted in parallel.

D. Evaluate internal validity

The fourth main step of ValidKI is concerned with evaluating whether the set of key indicators is internally valid with respect to the business objective. We distinguish between two sub-steps. In the first sub-step we reformulate the precise business objective by expressing it in terms of the identified key indicators. This step serves as an introductory step in the evaluation of internal validity. In the second sub-step we evaluate whether the set of key indicators is internally valid by showing that the reformulated precise business objective from Step 4.1 correctly measures the fulfillment of the precise business objective from Step 1.1.

Internal validity may be decomposed into a broad category of criteria [8]. In the following we list the criteria that we take into consideration. For each criterion, we first provide the definition as given in [8], before we list the papers on which the definition is based.

- **Attribute validity:** “A metric has attribute validity if the measurements correctly exhibit the attribute that the metric is intending to measure” [9][10]. In our case, the key indicator needs to correctly exhibit the risk attribute (likelihood or consequence) of the risk that it is measuring. In addition, the key indicator is of little value if it can only produce values that always result in the risk being acceptable or unacceptable.

- **Factor independence:** “A metric has factor independence if the individual measurements used in the metric formulation are independent of each other” [11]. This criterion applies especially to composite key indicators that are composed of basic key indicators. A composite key indicator has factor independence if the basic key indicators are independent of each other, i.e., if they do not rely on the same measurements.

- **Internal consistency:** “A metric has internal consistency if “all of the elementary measurements of a metric are assessing the same construct and are inter-related”” [12]. This criterion also applies especially to composite key indicators that are composed of basic key indicators. If the basic key indicators measure things that are not conceptually related, then the composite key indicator will not have internal consistency. For instance, let us say that we have a composite key indicator that is composed of two basic key indicators. The first basic key indicator measures the code complexity of a software product, while the second measures the cost of shipping the software product to the customers. In this case, the composite key indicator does not have internal consistency, since the two basic key indicators are not conceptually related.

- **Appropriate continuity:** “A metric has appropriate continuity if the metric is defined (or undefined) for all values according to the attribute being measured” [10]. An example of a discontinuity is fraction calculations when the denominator is zero. To avoid discontinuity, the key indicator should be defined for that case.

- **Dimensional consistency:** “A metric has dimensional consistency if the formulation of multiple metrics into a composite metric is performed by a scientifically well-understood mathematical function” [10][13]. Under dimensional consistency, no information should be lost during the construction of composite key indicators. Loss of information may be experienced if different scales are used for the basic and composite key indicators.

- **Unit validity:** “A metric has unit validity if the units used are an appropriate means of measuring the attribute” [10][14]. For instance, the unit fault rate may be used to measure the attribute program correctness [10].

If the set is not internally valid, then we iterate by re-doing Step 3.
E. Specify key indicator designs

In the fifth main step of ValidKI we specify the designs of the identified key indicators. Each design specifies how the key indicator should be calculated. The design also shows how sensors, actors, and different components interact.

F. Evaluate construct validity

In the sixth main step of ValidKI we evaluate whether the set of key indicators has construct validity with respect to the business objective. As with internal validity, construct validity may be decomposed into a broad category of criteria [8]. In the following we list the criteria that we take into consideration. For each criterion, we first provide the definition as given in [8], before we list the papers on which the definition is based.

- **Stability:** “A metric has stability if it produces the same values “on repeated collections of data under similar circumstances”” [12][15][16]. A key indicator whose calculation involves decisions made by humans, may for example result in different values and thus lack of stability.

- **Instrument validity:** “A metric has instrument validity if the underlying measurement instrument is valid and properly calibrated” [10]. In our case, this criterion concerns the sensors that perform the measurements that the key indicator calculations rely on.

- **Definition validity:** “A metric has definition validity if the metric definition is clear and unambiguous such that its collection can be implemented in a unique, deterministic way” [11][15][16][17][18]. This criterion concerns the implementation of the key indicators. To implement a key indicator correctly, the key indicator’s design specification needs to be clear and unambiguous.

To evaluate the different criteria, we re-do the risk analysis from Step 2.2 with the precise business objective replaced by the reformulated precise business objective, which is the precise business objective expressed in terms of key indicators. For each key indicator we identify risks towards the correctness of the reformulated precise business objective that are the result of threats to criteria for construct validity that the key indicator needs to fulfill. If the risk analysis does not result in any new unacceptable risks, then we have established construct validity for each key indicator. If the set does not have construct validity, then we iterate. We will most likely be re-doing Step 5, but it may also be the case that we need to come up with new key indicators and new sensors. In that case, we re-do Step 3. If the set of key indicators is both internally valid and has construct validity with respect to the business objective, then we have established that the set is valid.

IV. ESTABLISH TARGET

In the following we assume that we have been hired to help the public hospital Client H design key indicators to monitor their compliance with Article 8 in the European Convention on Human Rights [19]. The article states the following:

**Article 8 – Right to respect for private and family life**

1) Everyone has the right to respect for his private and family life, his home and his correspondence.

2) There shall be no interference by a public authority with the exercise of this right except such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others.

Client H needs to comply with Article 8 since it is a public authority. The consequence for Client H of not complying with Article 8 may be economic loss and damaged reputation. One example [20] of violation of Article 8 is from Finland. A Finnish woman was first treated for HIV at a hospital, before she later started working there as a nurse. While working there she suspected that her co-workers had unlawfully gained access to her medical data. She brought the case to the European Court of Human Rights in Strasbourg which unanimously held that the district health authority responsible for the hospital had violated Article 8 by not protecting the medical data of the woman properly. The district health authority was held liable to pay damages to the woman. Client H has therefore established the following business objective:

Client H wants to make use of key indicators to monitor the degree of fulfillment of BO-A8, and now they have hired us to use ValidKI to design them. In the rest of this section we conduct Step 1 of ValidKI on behalf of Client H with respect to BO-A8.

A. Express business objectives more precisely (Step 1.1 of ValidKI)

Article 8 states under which circumstances a public authority can interfere with someone’s right to privacy. One of these circumstances is “for the protection of health,” which is what Client H wants us to focus on. In the context of Client H this means to provide medical assistance to patients. The ones who provide this assistance are the health-care professionals of Client H.

The medical history of a patient is regarded as both sensitive and private. At Client H, the medical history of a patient is stored in an electronic patient record (EPR). An EPR is “an electronically managed and stored collection or collocation of recorded/registered information on a patient in connection with medical assistance” [21]. The main purpose of an EPR is to communicate information between health-care professionals that provide medical care to a patient. To protect the privacy of its patients, Client H restricts the use of EPRs. In order to comply with Article 8, Client H allows a health-care professional to interfere with the privacy of a patient only when providing medical assistance to this patient. Hence, the dealing with EPRs within the realms of Client H is essential.

For Client H it is important that every access to information in an EPR is in accordance with Article 8. A health-care professional should only access a patient’s EPR if he/she provides medical assistance to that patient, and he/she should only access information that is necessary for providing the medical assistance. The information accessed can not be used for any other purpose than providing medical assistance to patients. Accesses to information in EPRs not needed for providing medical assistance would not be in accordance with Article 8. Also, employees that are not health-care professionals and work within the jurisdiction of Client H are not allowed to access EPRs. Based on the constraints provided by Client H, we decide to express BO-A8 more precisely as follows:

Precise business objective PBO-A8: $C_1 \land C_2 \land C_3$

- **Constraint $C_1$:** Health-care professionals acting on behalf of Client H access:
  - a patient’s EPR only when providing medical assistance to that patient
  - only the information in a patient’s EPR that is necessary for providing medical assistance to that patient
- **Constraint $C_2$:** Health-care professionals acting on behalf of Client H do not use the information obtained from a patient’s EPR for any other purpose than providing medical assistance to that patient.
- **Constraint $C_3$:** Employees that are not health-care professionals and that work within the jurisdiction of Client H do not access EPRs.

As indicated by PBO-A8’s definition, all three constraints must be fulfilled in order for PBO-A8 to be fulfilled.

B. Describe relevant part of business (Step 1.2 of ValidKI)

To design key indicators to monitor BO-A8 we need to understand the part of business that is to comply with BO-A8 and therefore is to be monitored. “Public hospital Client H” has outsourced some of its medical services to two private hospitals. These two are referred to as “Private hospital X-ray” and “Private hospital Blood test analysis” in Fig. 4. The first hospital does all the X-ray work for Client H, while the second hospital does all the blood test analyses. Client H is not only responsible for its own handling of EPRs, but also the outsourcing partners’ handling of EPRs, when they act on behalf of Client H.

In Fig. 4, the rectangles inside and outside the gray containers represent systems/actors, while the arrows in the figure represent the exchange of data between different systems/actors. In the figure, we only show some of the rectangles and arrows that should be part of the gray containers of “Public hospital Client H” and “Private hospital Blood test analysis.” All the rectangles and arrows with names in italic that are part of the gray container of “Private hospital X-ray” should also be part of the gray containers of “Public hospital Client H” and “Private hospital Blood test analysis.”

As can be seen in Fig. 4, Client H outsources medical tasks to the two private hospitals, and gets in return the results from performing these tasks. All three health-care institutions employs some kind of EPR system for
handling the EPRs. An EPR system is “an electronic system with the necessary functionality to record, retrieve, present, communicate, edit, correct, and delete information in electronic patient records” [21]. These systems use EPRs provided by different health-care institutions. As shown in Fig. 4, these systems are only of interest when they handle EPRs where Client H is responsible for their handling.

At the three health-care institutions, most of the medical tasks that a health-care professional conducts during a working day are known in advance. It is known which patients the professional will treat and what kind of information the professional will need access to in order to treat the different patients. Client H and the two outsourcing partners maintain for each health-care professional an authorization list documenting which patients the professional is treating and what kind of information the professional needs for this purpose. These lists are used by the EPR systems and they are updated on a daily basis by the medical task management systems. Many of these updates are automatic. For instance, when Client H is assigned a new patient, then this patient is added to the lists of the health-care professionals who will be treating this patient.
Each EPR is owned by a patient, which is natural since the information stored in the EPR is about the patient in question. As already mentioned, the content of a patient's EPR is both considered sensitive and private. Moreover, some of the EPRs may contain information that is considered highly sensitive and private. Such information may for instance describe medical treatment received by a patient in relation to:

- the patient being the victim of a crime (e.g., rape, violence, etc.);
- sexual transferable diseases or abortion; and
- mortal or infectious mortal diseases.

Information classified as highly sensitive and private is handled with even more care than information that is just classified as sensitive and private. To raise awareness of the criticality of such information and to enable monitoring of its use, the EPR systems at the three health-care institutions tag highly sensitive and private information in EPRs based on predefined rules.

Accesses to information in EPRs can be classified as authorized or unauthorized based on the authorization lists of health-care professionals. An access is classified as authorized if the professional needs the information to do a planned task. Otherwise, the access is classified as unauthorized. If an access is classified as unauthorized then it is possible to check in retrospect whether the access was necessary. In an emergency situation, for instance when a patient is having a heart attack, a health-care professional often needs access to information in an EPR that he/she was not supposed to access. By checking in retrospect whether unauthorized accesses were necessary it is possible to classify the unauthorized accesses into two groups; one for accesses that were necessary, and one for those that were not. The first group is called approved unauthorized accesses, while the second group is called not approved unauthorized accesses. All accesses that are classified as not approved unauthorized accesses are considered as illegal accesses.

At Client H and the two outsourcing partners, health-care professionals use smart cards for accessing information in EPRs. If a card is lost or stolen, the owner must report it as missing, since missing cards may be used by other health-care professionals or others to access EPRs illegally. When the card has been registered as missing it can no longer be used. When reporting it as missing, the last time the card owner used it before noticing that it was missing is recorded. All accesses to EPRs that have occurred between this time and the time it was registered as missing are considered as illegal accesses.

At the three hospitals, the doors into the different areas are fitted with smart card locks. In order to open a door, an employee needs to insert his/hers smart card into the lock. A security system is used by each hospital to allow or deny an employee access to a specific area based on the employee’s access credentials. Moreover, health-care professionals often need to print information in EPRs. Each hospital relies on a printing system to achieve this. This system issues the different print jobs to printers located in rooms with doors fitted with smart card locks. Since each printer is used by a number of employees, the three hospitals run the risk of printed information being disclosed to other employees if the employee responsible for the print job forgets to collect his/hers printout. To minimize this risk, each hospital has security employees that collect uncollected printouts of information from EPRs at the different printers on a regular basis. Each printer at the three hospitals annotates each printout with the date and time it was printed, as well as an ID for the employee that issued the print job. A security employee removes a printout of sensitive and private information from an EPR if it has been laying on the printer for 30 minutes or more, while he/she removes a printout of highly sensitive and private information if it has been laying on the printer for 15 minutes or more. For each removed printout, the health-care professional that issued the print job is notified about the removal and asked to collect the printout at the security office at the hospital in question.

A health-care professional relies from time to time on information obtained from patients’ EPRs for other purposes than providing medical assistance to the patients in question. The information may be needed for the purpose of providing medical assistance to another patient, or it may be needed in research projects. To support these tasks, the three hospitals have made it possible for health-care professionals to obtain anonymized information from EPRs, i.e., information that cannot be linked to specific patients. It should be noticed that health-care professionals need to obtain specific permissions to obtain and use anonymized information from EPRs.

At each of the three hospitals, a media retriever service is used to collect relevant information from the traditional media (newspapers, TV, radio, etc.) and the Internet media (Internet newspapers, etc.). The three hospitals also encourage the general public to provide feedback on how satisfied they are with the hospitals’ services. The general public also serves another purpose for the three hospitals. The media retriever services can only to a limited extent retrieve information from social media (Facebook, Twitter, etc.) and Internet forums. The three hospitals therefore
TABLE I  
CONSEQUENCE SCALE FOR THE ASSET “FULFILLMENT OF PBO-A8” (TOP) AND LIKELIHOOD SCALE (BOTTOM)

<table>
<thead>
<tr>
<th>Consequence</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catastrophic</td>
<td>Law enforcement agencies penalize Client H after having been notified about the incident</td>
</tr>
<tr>
<td>Major</td>
<td>Health authorities penalize Client H after having been notified about the incident</td>
</tr>
<tr>
<td>Moderate</td>
<td>Health authorities are notified about the incident</td>
</tr>
<tr>
<td>Minor</td>
<td>Head of hospital is notified about the incident</td>
</tr>
<tr>
<td>Insignificant</td>
<td>Head of department is notified about the incident</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certain</td>
<td>Five times or more per year $[50, \infty)$ : 10 years</td>
</tr>
<tr>
<td>Likely</td>
<td>Two to five times per year $[20, 49]$ : 10 years</td>
</tr>
<tr>
<td>Possible</td>
<td>Once a year $[6, 19]$ : 10 years</td>
</tr>
<tr>
<td>Unlikely</td>
<td>Less than once per year $[2, 5]$ : 10 years</td>
</tr>
<tr>
<td>Rare</td>
<td>Less than once per ten years $[0, 1]$ : 10 years</td>
</tr>
</tbody>
</table>

TABLE II  
RISK EVALUATION MATRIX FOR THE ASSET “FULFILLMENT OF PBO-A8”

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Insignificant</th>
<th>Minor</th>
<th>Moderate</th>
<th>Major</th>
<th>Catastrophic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unlikely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Likely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

eourage the general public to notify them about information found in social media or on Internet forums that may be of relevance. A person of the general public is awarded if the information is very relevant. The information provided by the media retriever services and the general public is first and foremost used by the hospitals to assess how they are perceived by the public. Sometimes, however, the collected information may indicate or reveal that information from EPRs have been leaked to the public.

V. IDENTIFY RISKS TO FULFILLMENT OF BUSINESS OBJECTIVE

A. Specify risk acceptance criteria (Step 2.1 of ValidKI)

Before we specify the risk acceptance criteria, we need to establish scales for measuring likelihood and consequence. Table I presents these scales. We view “Fulfillment of PBO-A8” as the asset to be protected. In Table II the risk acceptance criteria for the asset “Fulfillment of PBO-A8” are expressed in terms of a risk evaluation matrix. Risks whose values belong to the white area of the matrix are acceptable, while risks whose values belong to the gray area are unacceptable.

B. Risk identification and estimation (Step 2.2 of ValidKI)

Based on the information provided by the representatives of Client H, we identify and estimate risk. For this purpose we use the CORAS methodology [22]. However, other approaches to risk analysis may be used instead. Using CORAS we identify how threats may initiate risks that harm the asset “Fulfillment of PBO-A8” if they occur.

The CORAS threat diagram in Fig. 5 provides a high-level overview of how the fulfillment of the precise business objective PBO-A8 may be harmed. The threat diagram contains four referring threat scenarios that refer to the referenced threat scenarios in Figs. 6 – 9. We refer to $i_x$ and $o_y$ of the referring threat scenarios as in-gate and out-gate, respectively. Relations to an element inside a referenced threat scenario must go through an in-gate, while relations to an element outside the referenced threat scenario must go through an out-gate. The likelihood value of an in-gate $i_x$ documents the contribution of an element outside the referenced threat scenario via gate $i_x$ to the
CORAS threat diagram providing a high-level overview of the results from the risk identification and estimation likelihood of an element inside the referenced threat scenario, while the likelihood of the out-gate $o_y$ documents the contribution of the likelihood of an element inside the referenced threat scenario via gate $o_y$ to the likelihood of an element outside the referenced threat scenario.

The CORAS threat diagram in Fig. 5 contains three human threats; one accidental (the white one) and two deliberate (the black ones). The accidental human threat “Health-care professional” may initiate the threat scenario “Unauthorized access to information in a patient’s EPR” in the referenced threat scenario “A health-care professional performs a not approved unauthorized access to information in an EPR” in Fig. 7 via the in-gate $i_3$ with likelihood “Likely” by exploiting the vulnerability “No restrictions on what EPRs a health-care professional can access.” We can also see that the deliberate human threat “Health-care professional” may initiate this threat scenario via the in-gate $i_4$ with likelihood “Possible” by exploiting the same vulnerability, and that the threat scenario occurs with likelihood “Certain.” If the threat scenario in Fig. 7 occurs then it leads to the threat scenario “Unauthorized access to sensitive and private information” in the same figure with conditional likelihood “0.7.” This threat scenario leads to the risk “R5: Not approved unauthorized access to sensitive and private information in an EPR, where the owner
EPR information printed by a health-care professional is found by another health-care professional or an employee that is not a health-care professional on the printer.

Fig. 6. The referenced threat scenario “EPR information printed by a health-care professional is found by another health-care professional or an employee that is not a health-care professional on the printer,” referred to in Fig. 5 of the EPR is a patient of the accessor” with conditional likelihood “0.6” if it occurs. The risk occurs with likelihood “Likely.” As can be seen in Figs. 5 and 7, the risk impacts the asset “Fulfillment of PBO-A8” via the out-gate $o_7$ with consequence “Insignificant” if it occurs.

The referenced threat scenarios in Figs. 6 – 9 document risks that affect the fulfillment of the constraints referred to in the precise business objective PBO-A8. The risks $R_2$, $R_4$, $R_5 – R_8$, $R_{15}$, and $R_{16}$ affect the fulfillment of constraint $C_1$, while the risks $R_9 – R_{14}$ affect the fulfillment of constraint $C_2$. Moreover, the risks $R_1$, $R_3$, $R_{17}$, and $R_{18}$ affect the fulfillment of constraint $C_3$. Notice that in the referenced threat scenario in Fig. 7, we distinguish between not approved unauthorized accesses to information in EPRs where the owner of the EPR is a patient and not a patient of the accessor. Client H finds it most serious if the owner of the EPR is not a patient of the accessor. We also distinguish between not approved unauthorized accesses to sensitive and private information and not approved unauthorized accesses to highly sensitive and private information. Naturally, Client H finds not approved unauthorized accesses to the latter type of information the most serious.
A health-care professional performs a not approved unauthorized access to information in an EPR,

R5: Not approved unauthorized access to sensitive and private information in an EPR, where the owner of the EPR is a patient of the accessor

R6: Not approved unauthorized access to sensitive and private information in an EPR, where the owner of the EPR is not a patient of the accessor

R7: Not approved unauthorized access to highly sensitive and private information in an EPR, where the owner of the EPR is a patient of the accessor

R8: Not approved unauthorized access to highly sensitive and private information in an EPR, where the owner of the EPR is not a patient of the accessor

A referenced threat scenario “A health-care professional performs a not approved unauthorized access to information in an EPR,” referred to in Fig. 5

Fig. 7. The referenced threat scenario “A health-care professional performs a not approved unauthorized access to information in an EPR,” referred to in Fig. 5

TABLE III
THE RISK EVALUATION MATRIX FROM TABLE II WITH THE ACCEPTABLE AND UNACCEPTABLE RISKS INSERTED

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rare</td>
<td>Insignificant</td>
</tr>
<tr>
<td>Unlikely</td>
<td>R15, R17</td>
</tr>
<tr>
<td>Possible</td>
<td>R3</td>
</tr>
<tr>
<td>Likely</td>
<td>R5, R13</td>
</tr>
<tr>
<td>Certain</td>
<td></td>
</tr>
</tbody>
</table>

C. Risk evaluation (Step 2.3 of ValidKI)

The risk evaluation consists in plotting the risks into the risk evaluation matrix according to their likelihoods and consequences. As indicated in Table III, four out of the 18 risks namely R8, R10, R12, and R16 are unacceptable with respect to the fulfillment of the precise business objective PBO-A8.

During the risk evaluation, we also decide that some of the risks need to be accumulated since they pull in the same direction. We decide to accumulate the following risks: R1 and R2; R3 and R4; R15 and R17; and R16 and R18. All of these risks, with the exception of R18, are acceptable when considered in isolation. Risks are accumulated by accumulating their likelihood and consequence values. We accumulate the risks as follows:
Information obtained from a patient’s EPR is used for other purposes than providing medical assistance to the patient in question.

Fig. 8. The referenced threat scenario “Information obtained from a patient’s EPR is used for other purposes than providing medical assistance to the patient in question,” referred to in Fig. 5
The referenced threat scenario “A health-care professional or an employee that is not a health-care professional uses a lost/stolen smart card to access information in an EPR,” referred to in Fig. 5:

- The accumulated risk “R1 & R2: Another health-care professional or employee that is not a health-care professional finds a printout of sensitive and private information on the printer.” It occurs with likelihood “Likely” and it impacts the asset with consequence “Moderate.” The accumulated risk is based on:
  - The risk R1 which occurs with likelihood “Possible,” while it impacts the asset with consequence “Moderate.”
  - The risk R2 which occurs with likelihood “Likely,” while it impacts the asset with consequence “Minor.”

- The accumulated risk “R3 & R4: Another health-care professional or employee that is not a health-care professional finds a printout of highly sensitive and private information on the printer.” It occurs with likelihood “Possible” and it impacts the asset with consequence “Major.” The accumulated risk is based on:
  - The risk R3 which occurs with likelihood “Unlikely,” while it impacts the asset with consequence “Major.”
  - The risk R4 which occurs with likelihood “Possible,” while it impacts the asset with consequence “Moderate.”

- The accumulated risk “R15 & R17: Access by a health-care professional or an employee that is not a health-care professional to sensitive and private information in an EPR from a lost/stolen smart card.” It occurs with likelihood “Rare” and it impacts the asset with consequence “Major.” The accumulated risk is based on:
  - The risk R15 which occurs with likelihood “Rare,” while it impacts the asset with consequence “Major.”
  - The risk R17 which occurs with likelihood “Rare,” while it impacts the asset with consequence “Major.”

- The accumulated risk “R16 & R18: Access by a health-care professional or an employee that is not a health-care professional to highly sensitive and private information in an EPR from a lost/stolen smart card.” It occurs
TABLE IV
THE RISK EVALUATION MATRIX FROM TABLE III AFTER RISKS HAVE BEEN ACCUMULATED

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
<th>Insignificant</th>
<th>Minor</th>
<th>Moderate</th>
<th>Major</th>
<th>Catastrophic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rare</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unlikely</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible</td>
<td></td>
<td>R6, R7</td>
<td></td>
<td></td>
<td>R9, R11</td>
<td>R3k&amp;R4, R8, R10</td>
</tr>
<tr>
<td>Likely</td>
<td></td>
<td>R13</td>
<td></td>
<td></td>
<td>R5, R14</td>
<td>R1k&amp;R2</td>
</tr>
<tr>
<td>Certain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

with likelihood “Unlikely” and it impacts the asset with consequence “Catastrophic.” The accumulated risk is based on:

- The risk R16 which occurs with likelihood “Unlikely,” while it impacts the asset with consequence “Catastrophic.”
- The risk R18 which occurs with likelihood “Rare,” while it impacts the asset with consequence “Catastrophic.”

Since we are operating with a coarse-grained likelihood scale with intervals, we find it sufficient to do a rough aggregation of the likelihoods in order to determine to which likelihood interval the different accumulated risks belong. For the accumulated risk R15&R17 we end up with the likelihood “Rare,” while for each of the other accumulated risks, we end up with an aggregated likelihood that gravitates towards the highest of the two likelihoods. We therefore decide to use the highest likelihood to represent the accumulated likelihood in each of these cases. Moreover, we accumulate consequences by taking the average. In all of the cases where the two consequence values differ, we end up with an average that gravitates towards the highest consequence value. We therefore find it suitable to use the highest consequence value to represent the accumulated consequence in each of these cases.

In Table IV we have plotted the accumulated risks according to their likelihoods and consequences. As we can see from the table, all the accumulated risks with the exception of R15&R17 are unacceptable. Table IV shows that the risks R1k&R2, R3k&R4, R8, R10, R12, and R16&R18 are unacceptable with respect to the fulfillment of the precise business objective PBO-A8.

VI. IDENTIFY KEY INDICATORS TO MONITOR RISKS

A. Deploy sensors to monitor risks (Step 3.1 of ValidKI)

Fig. 10, which is a detailing of the target description in Fig. 4, specifies the deployment of sensors in the relevant part of business. This specification corresponds to the sensor deployment specification referred to in Fig. 2. An antenna-like symbol is used to represent each sensor in Fig. 10. The different sensors monitor data messages exchanged within the relevant part of business. The results from the monitoring are to be used in the calculation of key indicators.

In Fig. 10, sensor deployments are only shown for “Private hospital X-ray.” It should be noticed that “Public hospital Client H” and “Private hospital Blood test analysis” will have the same sensors as “Private hospital X-ray.” The following sensors are deployed in the relevant part of business:

- S_{CH-REG-MIS-SC}, S_{BTA-REG-MIS-SC}, and S_{XR-REG-MIS-SC} monitor data messages related to the registration of missing smart cards at Client H, Blood test analysis, and X-ray, respectively.
- S_{CH-AUTH-LIST}, S_{BTA-AUTH-LIST}, and S_{XR-AUTH-LIST} monitor data messages related to the authorization lists employed by the EPR systems at Client H, Blood test analysis, and X-ray, respectively.
- S_{CH-ACC-INFO-EPR}, S_{BTA-ACC-INFO-EPR}, and S_{XR-ACC-INFO-EPR} monitor data messages where each message is a request issued by health-care professional to access information in an EPR at Client H, Blood test analysis, and X-ray, respectively. It is not necessary to monitor the actual information received, since health-care professionals will always get the information they request.
- S_{CH-INFO-GP}, S_{BTA-INFO-GP}, and S_{XR-INFO-GP} monitor data messages where each message contains info/feedback from the general public for Client H, Blood test analysis, and X-ray, respectively.
• $S_{CH-INF-MRS}$, $S_{BTA-INF-MRS}$, and $S_{XR-INF-MRS}$ monitor data messages where each message contains relevant information collected by media retriever services from the traditional media or the Internet for Client H, Blood test analysis, and X-ray, respectively.

• $S_{CH-PR-REQS}$, $S_{BTA-PR-REQS}$, and $S_{XR-PR-REQS}$ monitor data messages related to printing of information in EPRs by health-care professionals at Client H, Blood test analysis, and X-ray, respectively.

• $S_{CH-ACC-REQS-HCP}$, $S_{BTA-ACC-REQS-HCP}$, and $S_{XR-ACC-REQS-HCP}$ monitor data messages related to area access requests issued by health-care professionals at Client H, Blood test analysis, and X-ray, respectively.

• $S_{CH-ACC-REQS-NHCP}$, $S_{BTA-ACC-REQS-NHCP}$, and $S_{XR-ACC-REQS-NHCP}$ monitor data messages related to area access requests issued by employees that are not health-care professionals at Client H, Blood test analysis, and X-ray, respectively.

• $S_{CH-INF-UNC-PO}$, $S_{BTA-INF-UNC-PO}$, and $S_{XR-INF-UNC-PO}$ monitor data messages related to registrations of uncol-
TABLE V
KEY INDICATOR REQUIREMENTS SPECIFICATIONS FOR THE COMPOSITE KEY INDICATOR $K_{PR-SP-EPR-INFO}$ AND THE BASIC KEY INDICATORS $K_{CH-PR-SP-EPR-INFO}$, $K_{BTA-PR-SP-EPR-INFO}$, AND $K_{XR-PR-SP-EPR-INFO}$

<table>
<thead>
<tr>
<th>Requirements for $K_{X-PR-SP-EPR-INFO}$ where $X \in {CH, BTA, XR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>In: $S_{X-ACC-REQS-NHCP}, S_{X-ACC-REQS-HCP}, S_{X-PR-REQS}, S_{X-INFO-UNC-PO} : M^*$</td>
</tr>
<tr>
<td>Out: $K_{X-PR-SP-EPR-INFO} : R$</td>
</tr>
<tr>
<td>Description: $K_{X-PR-SP-EPR-INFO} = \text{&quot;The number of times since the monitoring started that health-care professionals or employees that are not health-care professionals have found printouts of sensitive and private information from EPRs on printers at } X\text{&quot;}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements for $K_{PR-SP-EPR-INFO}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>In: $S_{CH-ACC-REQS-NHCP}, S_{BTA-ACC-REQS-NHCP}, S_{XR-ACC-REQS-NHCP} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-ACC-REQS-HCP}, S_{BTA-ACC-REQS-HCP}, S_{XR-ACC-REQS-HCP} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-PR-REQS}, S_{BTA-PR-REQS}, S_{XR-PR-REQS} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-INFO-UNC-PO}, S_{BTA-INFO-UNC-PO}, S_{XR-INFO-UNC-PO} : M^*$</td>
</tr>
<tr>
<td>Out: $K_{PR-SP-EPR-INFO} : R$</td>
</tr>
<tr>
<td>Description: $K_{PR-SP-EPR-INFO} = \frac{10 \cdot (K_{CH-PR-SP-EPR-INFO} + K_{BTA-PR-SP-EPR-INFO} + K_{XR-PR-SP-EPR-INFO})}{\text{Number of years since the monitoring started}}$</td>
</tr>
</tbody>
</table>

TABLE VI
KEY INDICATOR REQUIREMENTS SPECIFICATIONS FOR THE COMPOSITE KEY INDICATOR $K_{PR-HSP-EPR-INFO}$ AND THE BASIC KEY INDICATORS $K_{CH-PR-HSP-EPR-INFO}$, $K_{BTA-PR-HSP-EPR-INFO}$, AND $K_{XR-PR-HSP-EPR-INFO}$

<table>
<thead>
<tr>
<th>Requirements for $K_{X-PR-HSP-EPR-INFO}$ where $X \in {CH, BTA, XR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>In: $S_{X-ACC-REQS-NHCP}, S_{X-ACC-REQS-HCP}, S_{X-PR-REQS}, S_{X-INFO-UNC-PO} : M^*$</td>
</tr>
<tr>
<td>Out: $K_{X-PR-HSP-EPR-INFO} : R$</td>
</tr>
<tr>
<td>Description: $K_{X-PR-HSP-EPR-INFO} = \text{&quot;The number of times since the monitoring started that health-care professionals or employees that are not health-care professionals have found printouts of highly sensitive and private information from EPRs on printers at } X\text{&quot;}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements for $K_{PR-HSP-EPR-INFO}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>In: $S_{CH-ACC-REQS-NHCP}, S_{BTA-ACC-REQS-NHCP}, S_{XR-ACC-REQS-NHCP} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-ACC-REQS-HCP}, S_{BTA-ACC-REQS-HCP}, S_{XR-ACC-REQS-HCP} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-PR-REQS}, S_{BTA-PR-REQS}, S_{XR-PR-REQS} : M^*$</td>
</tr>
<tr>
<td>$S_{CH-INFO-UNC-PO}, S_{BTA-INFO-UNC-PO}, S_{XR-INFO-UNC-PO} : M^*$</td>
</tr>
<tr>
<td>Out: $K_{PR-HSP-EPR-INFO} : R$</td>
</tr>
<tr>
<td>Description: $K_{PR-HSP-EPR-INFO} = \frac{10 \cdot (K_{CH-PR-HSP-EPR-INFO} + K_{BTA-PR-HSP-EPR-INFO} + K_{XR-PR-HSP-EPR-INFO})}{\text{Number of years since the monitoring started}}$</td>
</tr>
</tbody>
</table>

lected printouts of information from EPRs by security employees at Client H, Blood test analysis, and X-ray, respectively.

B. Specify requirements to key indicators wrt deployed sensors (Step 3.2 of ValidKI)

Two key indicators $K_{PR-SP-EPR-INFO}$ and $K_{PR-HSP-EPR-INFO}$ are identified to monitor the likelihood values of the two unacceptable risks $R1$ & $R2$ and $R3$ & $R4$, respectively. In Tables V and VI their requirements are given. The two key indicators calculate likelihoods with respect to a ten year period, because the likelihoods in the likelihood scale in Table I are defined with respect to a ten year period. Both key indicators are composed of basic key indicators. Table V presents the requirements to the basic key indicators that $K_{PR-SP-EPR-INFO}$ is composed of, while Table VI presents the requirements to the basic key indicators that $K_{PR-HSP-EPR-INFO}$ is composed of.

For each key indicator we specify required sensor data. All of the key indicators rely on sequences of data
TABLE VII
KEY INDICATOR REQUIREMENTS SPECIFICATIONS FOR THE COMPOSITE KEY INDICATOR \( K_{\text{NOT-APP-UNAUTH-ACC}} \) AND THE BASIC KEY INDICATORS \( K_{\text{CH-NOT-APP-UNAUTH-ACC}}, K_{\text{BTA-NOT-APP-UNAUTH-ACC}}, \) AND \( K_{\text{XR-NOT-APP-UNAUTH-ACC}} \)

<table>
<thead>
<tr>
<th>Requirements for ( K_{X,\text{NOT-APP-UNAUTH-ACC}} ), where ( X \in { \text{CH}, \text{BTA}, \text{XR} } )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In:</strong></td>
</tr>
<tr>
<td><strong>Out:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements for ( K_{\text{NOT-APP-UNAUTH-ACC}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In:</strong></td>
</tr>
<tr>
<td><strong>Out:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
</tbody>
</table>

Number of years since the monitoring started

TABLE VIII
KEY INDICATOR REQUIREMENTS SPECIFICATIONS FOR THE COMPOSITE KEY INDICATOR \( K_{\text{SP-EPR-INFO}} \) AND THE BASIC KEY INDICATORS \( K_{\text{CH-SP-EPR-INFO}}, K_{\text{BTA-SP-EPR-INFO}}, \) AND \( K_{\text{XR-SP-EPR-INFO}} \)

<table>
<thead>
<tr>
<th>Requirements for ( K_{X,\text{SP-EPR-INFO}} ), where ( X \in { \text{CH}, \text{BTA}, \text{XR} } )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In:</strong></td>
</tr>
<tr>
<td><strong>Out:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Requirements for ( K_{\text{SP-EPR-INFO}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>In:</strong></td>
</tr>
<tr>
<td><strong>Out:</strong></td>
</tr>
<tr>
<td><strong>Description:</strong></td>
</tr>
</tbody>
</table>

Number of years since the monitoring started

messages \((M^*)\) gathered by the different sensors. We also specify the output type and requirements to output. For a key indicator \( K \) we refer to its requirement description as \( \text{Req}(K) \).

Key indicators have also been identified for monitoring the unacceptable risks \( R8, R10, R12, \) and \( R16 & R18 \). Tables VII, VIII, IX, and X specify requirements to key indicators for monitoring the likelihood values of the risks \( R8, R10, R12, \) and \( R16 & R18 \), respectively.
TABLE IX
KEY INDICATOR REQUIREMENTS SPECIFICATIONS FOR THE COMPOSITE KEY INDICATOR \( \text{K}_{\text{HSP-EPR-INFO}} \) AND THE BASIC KEY INDICATORS \( \text{K}_{\text{CH-HSP-EPR-INFO}} \), \( \text{K}_{\text{BTA-HSP-EPR-INFO}} \), AND \( \text{K}_{\text{XR-HSP-EPR-INFO}} \)

<table>
<thead>
<tr>
<th>Requirements for ( \text{K}_{\text{x-HSP-EPR-INFO}} ) where ( X \in { \text{CH}, \text{BTA}, \text{XR} } )</th>
<th>In: ( S_{\text{X-ACC-INFO-EPR}}, S_{\text{X-INFO-GP}}, S_{\text{X-INFO-MRS}} )</th>
<th>Out: ( \text{K}_{\text{x-HSP-EPR-INFO}} )</th>
<th>Description: ( \text{K}_{\text{x-HSP-EPR-INFO}} = \text{&quot;The number of times since the monitoring started that highly sensitive and private information from patients' EPRs have been shared by health-care professionals with others and where this information have ended up in the traditional media or on the Internet&quot;} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements for ( \text{K}_{\text{HSP-EPR-INFO}} )</td>
<td>( S_{\text{CH-ACC-INFO-EPR}}, S_{\text{BTA-ACC-INFO-EPR}}, S_{\text{XR-ACC-INFO-EPR}}, S_{\text{CH-INFO-GP}}, S_{\text{BTA-INFO-GP}}, S_{\text{XR-INFO-GP}}, S_{\text{CH-INFO-MRS}}, S_{\text{BTA-INFO-MRS}}, S_{\text{XR-INFO-MRS}} )</td>
<td>( \text{K}_{\text{HSP-EPR-INFO}} )</td>
<td>( \text{K}<em>{\text{HSP-EPR-INFO}} = 10 \cdot \left( \text{K}</em>{\text{CH-HSP-EPR-INFO}} + \text{K}<em>{\text{BTA-HSP-EPR-INFO}} + \text{K}</em>{\text{XR-HSP-EPR-INFO}} \right) )</td>
</tr>
<tr>
<td>Requirements for ( \text{K}_{\text{ILL-ACC-SC}} ) where ( X \in { \text{CH}, \text{BTA}, \text{XR} } )</td>
<td>( S_{\text{X-REG-MIS-SC}}, S_{\text{X-ACC-INFO-EPR}} )</td>
<td>( \text{K}_{\text{X-ILL-ACC-SC}} )</td>
<td>Description: ( \text{K}_{\text{X-ILL-ACC-SC}} = \text{&quot;The number of illegal accesses at } X \text{ since the monitoring started to highly sensitive and private information in EPRs from lost/stolen smart cards&quot;} )</td>
</tr>
<tr>
<td>Requirements for ( \text{K}_{\text{ILL-ACC-SC}} )</td>
<td>( S_{\text{CH-REG-MIS-SC}}, S_{\text{BTA-REG-MIS-SC}}, S_{\text{XR-REG-MIS-SC}}, S_{\text{CH-ACC-INFO-EPR}}, S_{\text{BTA-ACC-INFO-EPR}}, S_{\text{XR-ACC-INFO-EPR}} )</td>
<td>( \text{K}_{\text{ILL-ACC-SC}} )</td>
<td>( \text{K}<em>{\text{ILL-ACC-SC}} = 10 \cdot \left( \text{K}</em>{\text{CH-ILL-ACC-SC}} + \text{K}<em>{\text{BTA-ILL-ACC-SC}} + \text{K}</em>{\text{XR-ILL-ACC-SC}} \right) )</td>
</tr>
</tbody>
</table>

VII. EVALUATE INTERNAL VALIDITY

A. Express business objective in terms of key indicators (Step 4.1 of ValidKI)

The precise business objective \( \text{PBO-A8’} \) is a reformulation of the precise business objective \( \text{PBO-A8} \) expressed in terms of key indicators.

\[
\text{PBO-A8’} = \text{K}_{\text{PR-SP-EPR-INFO}} \in [0, 19] \land \text{Req}(\text{K}_{\text{SP-PR-EPR-INFO}}) \land \text{K}_{\text{PR-HSP-EPR-INFO}} \in [0, 5] \land \text{Req}(\text{K}_{\text{HSP-PR-EPR-INFO}}) \land \text{K}_{\text{NOT-APP-UNAUTH-ACC}} \in [0, 5] \land \text{Req}(\text{K}_{\text{NOT-APP-UNAUTH-ACC}}) \land \text{K}_{\text{SP-EPR-INFO}} \in [0, 5] \land \text{Req}(\text{K}_{\text{SP-EPR-INFO}}) \land \text{K}_{\text{HSP-EPR-INFO}} \in [0, 1] \land \text{Req}(\text{K}_{\text{HSP-EPR-INFO}}) \land \text{K}_{\text{ILL-ACC-SC}} \in [0, 1] \land \text{Req}(\text{K}_{\text{ILL-ACC-SC}})
\]

The precise business objective \( \text{PBO-A8} \) is fulfilled if the likelihood values of the six unacceptable risks \( R_{1} \& R_{2}, R_{3} \& R_{4}, R_{8}, R_{10}, R_{12}, \) and \( R_{16} \& R_{18} \) change in such a way that the six risks become acceptable. The risks become acceptable if their likelihood values change in the following way:
The definitions of the risks far from being fulfilled. For instance, if "Possible" to "Unlikely" or "Rare." The likelihood will change in such a way if the composite key indicator $K_{\text{PR-SP-EPR-INFO}}$, monitoring the likelihood, is contained in the interval $[0, 5]$ (interval capturing both "Rare: $[0, 1]: 10$ years," "Unlikely: $[2, 5]: 10$ years").

- The risk $R8$ becomes acceptable if the likelihood changes from "Possible" to "Unlikely" or "Rare." The likelihood will change in such a way if the composite key indicator $K_{\text{SP-EPR-INFO}}$, monitoring the likelihood, is contained in the interval $[0, 5]$ (interval capturing both "Rare: $[0, 1]: 10$ years" and "Unlikely: $[2, 5]: 10$ years").

- The risk $R10$ becomes acceptable if the likelihood changes from "Possible" to "Unlikely" or "Rare." The likelihood will change in such a way if the composite key indicator $K_{\text{HSP-EPR-INFO}}$, monitoring the likelihood, is contained in the interval $[0, 5]$ (interval capturing both "Rare: $[0, 1]: 10$ years" and "Unlikely: $[2, 5]: 10$ years").

- The risk $R12$ becomes acceptable if the likelihood changes from "Unlikely" to "Rare." The likelihood will change in such a way if the composite key indicator $K_{\text{HSP-EPR-INFO}}$, monitoring the likelihood, is contained in the interval $[0, 1]$ (interval capturing "Rare: $[0, 1]: 10$ years").

Moreover, the different composite key indicators need to measure the likelihoods correctly in order to measure the fulfillment of PBO-A8. This can be determined based on the requirements to the different composite key indicators. These requirements are captured by $Req(K_{\text{PR-SP-EPR-INFO}}), Req(K_{\text{PR-HSP-EPR-INFO}}), etc.$

The reformulated precise business objective can also be used to determine to what degree the precise business objective is fulfilled. For instance, if $K_{\text{PR-SP-EPR-INFO}}$ equals 20 while the other composite key indicators equal 0, then PBO-A8 is close to being fulfilled. On the other hand, if $K_{\text{PR-SP-EPR-INFO}}$ equals 25 instead, then PBO-A8 is far from being fulfilled.

**B. Evaluate criteria for internal validity (Step 4.2 of ValidKI)**

To evaluate the internal validity of the set of key indicators, we need to show that the reformulated precise business objective PBO-A8 measures the fulfillment of the precise business objective PBO-A8. We evaluate the internal validity of each composite key indicator based on the criteria given in Section III-D.

To evaluate attribute validity we need to compare the definitions of the six risks with the requirements to the composite key indicators. The definitions of the risks $R8, R10,$ and $R12$ are given in Figs. 7 and 8, while the definitions of the accumulated risks $R1\&R2, R3\&R4,$ and $R16\&R18$ are given in Section V-C. Moreover, the
Moreover, the risks will have the following likelihood values when acceptable:

matrix in Table XI. In this situation, all of the risks indicators is internally valid. When the precise business objective PBO-A8 is fulfilled, we get the risk evaluation measuring the six likelihood attributes of the risks.

key indicators do also have unit validity. All six use the unit "likelihood per ten years," which is appropriate for lost when constructing the composite key indicators from their respective basic key indicators. The six composite transforms the resulting likelihood into a likelihood which is for a period of ten years. Thus, no information is adds three likelihoods, where each is for the period of "Number of years since the monitoring started" continuity. All are discontinuous if “Number of years since the monitoring started” equals zero. Client H does not consider this to be a problem, since the denominator will in all six cases be a real number that is never zero. We also show that the six composite key indicators have dimensional consistency. Each composite key indicator adds three likelihoods, where each is for the period of “Number of years since the monitoring started” years, and transforms the resulting likelihood into a likelihood which is for a period of ten years. Thus, no information is lost when constructing the composite key indicators from their respective basic key indicators. The six composite key indicators do also have unit validity. All six use the unit “likelihood per ten years,” which is appropriate for measuring the six likelihood attributes of the risks.

Based on the evaluation of the different internal validity types of criteria above, we conclude that the set of key indicators is internally valid. When the precise business objective PBO-A8 is fulfilled, we get the risk evaluation matrix in Table XI. In this situation, all of the risks R1&2, R3&4, R8, R10, R12, and R16&R18 are acceptable. Moreover, the risks will have the following likelihood values when acceptable:

- The risk R1&2 will either have the likelihood “Rare” (R1&2′), “Unlikely” (R1&2′′), or “Possible” (R1&2′′′).
- The risk R3&4 will either have the likelihood “Rare” (R3&4′) or “Unlikely” (R3&4′′).
- The risk R8 will either have the likelihood “Rare” (R8′) or “Unlikely” (R8′′).
- The risk R10 will either have the likelihood “Rare” (R10′) or “Unlikely” (R10′′).
- The risk R12 will have the likelihood “Rare”.
- The risk R16&R18 will have the likelihood “Rare”.

VIII. SPECIFY KEY INDICATOR DESIGNS

We use the UML [6] sequence diagram notation for the key indicator design specifications, but one may of course also use other languages depending on the problem in question. In the following sub-sections, we specify the designs of the six composite key indicators and their respective basic key indicators.

A. Key indicator designs for K_{PR-SP-EPR-INFO} and its basic key indicators

The sequence diagram in Fig. 11 specifies how the key indicator K_{PR-SP-EPR-INFO} is calculated. Each entity in the sequence diagram is either a component, a sensor, or an employee at Client H, and it is represented by a dashed, vertical line called a lifeline, where the box at its top specifies which entity the lifeline represents. The entities interact with each other through the transmission and reception of messages, which are shown as horizontal arrows from the transmitting lifeline to the receiving lifeline. We can also see that a lifeline can be both the sender and receiver of a message.

The sequence diagram contains one reference (ref) to another sequence diagram. This reference can be replaced by the content of the sequence diagram that it refers to. The reference refers to the sequence diagram given in Fig. 12, which describes the calculation of the basic key indicator K_{CH-PR-SP-EPR-INFO} at Client H. We do not present sequence diagrams describing the calculations of the two other basic key indicators, since these calculations are performed in the same way as the calculation of K_{CH-PR-SP-EPR-INFO}, and since these calculations involve the same
types of lifelines as the ones described in Fig. 12. For the two other basic key indicators we only show that they are sent to “Component for calculating K_{PR-SP-EPR-INFO},” and that they are used in the calculation of K_{PR-SP-EPR-INFO}.

The sequence diagram in Fig. 12 shows that the basic key indicator K_{CH-PR-SP-EPR-INFO} is updated each week. The first thing that happens is that “Component for calculating K_{CH-PR-SP-EPR-INFO}” retrieves the value that was computed for the basic key indicator in the previous week. Afterwards, the component counts for each printout the number of health-care professionals and employees that are not health-care professionals that accessed the printer room between TIME_1 (the time the print job was completed) and TIME_2 (the time when the health-care professional collected his/hers printout or the time when the printout was collected by a security employee). The number NUM is the number of other health-care professionals and employees that are not health-care professionals that may have seen the printout of sensitive and private information.

Client H is of the opinion that between 10% and 30% of the other health-care professionals and employees that are not health-care professionals that accessed the printer rooms between TIME_1 and TIME_2 have seen the printouts of sensitive and private information from patients’ EPRs. Thus, the number TOTAL_NUM is multiplied by [0.1, 0.3]. In the end, the component stores the basic key indicator before sending it to “Component for calculating K_{PR-SP-EPR-INFO},” as illustrated in the sequence diagram in Fig. 11.

B. Key indicator designs for K_{PR-HSP-EPR-INFO} and its basic key indicators

The sequence diagram in Fig. 13 specifies how the key indicator K_{PR-HSP-EPR-INFO} is calculated, while the sequence diagram in Fig. 14 describes the calculation of the basic key indicator K_{CH-PR-HSP-EPR-INFO} at Client H. We use the same argument as the one given in Section VIII-A for not presenting sequence diagrams for the two other basic key indicators.

The sequence diagram in Fig. 14 shows that the basic key indicator K_{CH-PR-HSP-EPR-INFO} is updated each week. This sequence diagram is almost identical to the one in Fig. 12. Thus, we do not give any further explanations for the sequence diagram.
Calculation of $K_{CH-PR-SP-EPR-INFO}$

**Component for calculating $K_{CH-PR-SP-EPR-INFO}$**

- **$K_{CH-PR-SP-EPR-INFO}$**: "The number of times since the start of the monitoring up to the end of the previous week that health-care professionals or employees that are not health-care professionals have found printouts of sensitive and private information from EPRs on printers at Client H".

**SP_PRINT_LIST**: All requests to print sensitive and private info in EPRs issued by health-care professionals at Client H in the period of one week backwards.

**ACC_PRINT_AREAS_HCP**: All requests to access printer rooms at Client H issued by health-care professionals in the period of one week backwards.

**ACC_PRINT_AREAS_NHCP**: All requests to access printer rooms at Client H issued by employees that are not health-care professionals in the period of one week backwards.

**REG_UNC_PO**: All registrations of uncollected printouts of sensitive and private information that have been removed by security employees at Client H in the period of one week backwards.

---

**Loop**

- Initialize $TOTAL_NUM = 0$.

- For each $PRINT_REQ$ in $SP_PRINT_LIST$:
  - Get the room number $ROOM_NO$ for the location of the printer handling $PRINT_REQ$.
  - Get the time $TIME_1$ for when $PRINT_REQ$ was printed and the time $TIME_2$ for when the health-care professional accessed $ROOM_NO$ to collect $PRINT_REQ$.
  - [If $TIME_2$ is undefined, then $TIME_2 = TIME_REM$.]

  - Get the number of health-care professionals or employees that are not health-care professionals $NUM$ that accessed $ROOM_NO$ between $TIME_1$ and $TIME_2$.

  - $TOTAL_NUM = TOTAL_NUM + NUM$.

- $K_{CH-PR-SP-EPR-INFO} = K_{CH-PR-SP-EPR-INFO} + TOTAL_NUM \cdot [0.1, 0.3]$

---

**Opt**

- Get the time $TIME_REM$ from $REG_UNC_PO$ for when the uncollected printout resulting from $PRINT_REQ$ was removed by a security employee.

- Store $K_{CH-PR-SP-EPR-INFO}$.

---

Fig. 12. The sequence diagram “Calculation of $K_{CH-PR-SP-EPR-INFO}$”
Calculation of $K_{PR-HSP-EPR-INFO}$

\[ K_{PR-HSP-EPR-INFO} = \frac{(10 \times (K_{CH-PR-HSP-EPR-INFO} + K_{KBTA-PR-HSP-EPR-INFO} + K_{KXR-PR-HSP-EPR-INFO}))}{\text{Number of years since the monitoring started}} \]

**Fig. 13.** The sequence diagram “Calculation of $K_{PR-HSP-EPR-INFO}$”

### C. Key indicator designs for $K_{NOT-APP-UNAUTH-ACC}$ and its basic key indicators

The sequence diagram in Fig. 15 specifies how the key indicator $K_{NOT-APP-UNAUTH-ACC}$ is calculated, while the sequence diagram in Fig. 16 describes the calculation of the basic key indicator $K_{CH-NOT-APP-UNAUTH-ACC}$ at Client H. We use the same argument as the one given in Section VIII-A for not presenting sequence diagrams for the two other basic key indicators.

The sequence diagram in Fig. 16 shows that the basic key indicator $K_{CH-NOT-APP-UNAUTH-ACC}$ is updated each week. The first thing that happens is that “Component for calculating $K_{CH-NOT-APP-UNAUTH-ACC}$” sends the value that was computed for the basic key indicator in the previous week to “Employee at Client H.” Afterwards, the component identifies “All unauthorized accesses at Client H in the period of one week backwards to highly sensitive and private information in EPRs, where the owners of the EPRs are not patients of the accessors” based on input from the entities representing the sensors. The “Employee at Client H” performs a manual inspection of each of these unauthorized accesses, and classifies each as approved or not approved. If the unauthorized access is classified as not approved, then the basic key indicator is incremented by one. After all the unauthorized accesses have been inspected and classified, “Employee at Client H” sends the basic key indicator to the component which stores it. Afterwards, the component sends the basic key indicator to “Component for calculating $K_{NOT-APP-UNAUTH-ACC}$,” as illustrated in the sequence diagram in Fig. 15.

### D. Key indicator designs for $K_{SP-EPR-INFO}$ and its basic key indicators

The sequence diagram in Fig. 17 specifies how the key indicator $K_{SP-EPR-INFO}$ is calculated, while the sequence diagram in Fig. 18 describes the calculation of the basic key indicator $K_{CH-SP-EPR-INFO}$ at Client H. We use the same argument as the one given in Section VIII-A for not presenting sequence diagrams for the two other basic key indicators.

The sequence diagram in Fig. 18 shows that the basic key indicator $K_{CH-SP-EPR-INFO}$ is updated each week. The first thing that happens is that “Component for calculating $K_{CH-SP-EPR-INFO}$” sends the value that was computed for the basic key indicator in the previous week to “Employee at Client H”. Afterwards, the component receives
**Calculation of $K_{\text{CH-PR-HSP-EPR-INFO}}$**

$K_{\text{CH-PR-HSP-EPR-INFO}}$ = "The number of times since the start of the monitoring up to the end of the previous week that health-care professionals or employees that are not health-care professionals have found printouts of highly sensitive and private information from EPRs on printers at Client H"

$HSP\_PRINT\_LIST$: All requests to print highly sensitive and private info in EPRs issued by health-care professionals at Client H in the period of one week backwards

$ACC\_PRINT\_AREAS\_NHCP$: All requests to access printer rooms at Client H issued by employees that are not health-care professionals in the period of one week backwards

$REG\_UNC\_PO$: All registrations of uncollected printouts of highly sensitive and private information that have been removed by security employees at Client H in the period of one week backwards

TOTAL_NUM = 0

**loop ( PRINT_REQ = 0, number of elements in HSP\_PRINT\_LIST )**

Get the room number ROOM_NO for where the printer handling PRINT_REQ is located

Get the time TIME_1 for when PRINT_REQ was printed and get the time TIME_2 for when the health-care professional accessed ROOM_NO to collect PRINT_REQ

opt

[TIME_2 is undefined]

Get the time TIME_REM from REG\_UNC\_PO for when the uncollected printout resulting from PRINT_REQ was removed by a security employee

TIME_2 = TIME_REM

Get the number of health-care professionals or employees that are not health-care professionals NUM that accessed ROOM_NO between TIME_1 and TIME_2

TOTAL_NUM = TOTAL_NUM + NUM

$K_{\text{CH-PR-HSP-EPR-INFO}} = K_{\text{CH-PR-HSP-EPR-INFO}} + \text{TOTAL_NUM} \cdot [0.1, 0.3]$

Store $K_{\text{CH-PR-HSP-EPR-INFO}}$

**Fig. 14.** The sequence diagram “Calculation of $K_{\text{CH-PR-HSP-EPR-INFO}}$“
The sequence diagram “Calculation of $K_{NOT-APP-UNAUTH-ACC}$”

<table>
<thead>
<tr>
<th>Component for calculating $K_{NOT-APP-UNAUTH-ACC}$</th>
<th>Sensor $S_{CH-AUTH-LIST}$</th>
<th>Sensor $S_{CH-ACC-INFO-EPR}$</th>
<th>Employee at Client H</th>
<th>Component for calculating $K_{CH-NOT-APP-UNAUTH-ACC}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ref</td>
<td>Calculation of $K_{CH-NOT-APP-UNAUTH-ACC}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$K_{CH-NOT-APP-UNAUTH-ACC}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$K_{KETA-NOT-APP-UNAUTH-ACC}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>$K_{KXR-NOT-APP-UNAUTH-ACC}$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$$K_{NOT-APP-UNAUTH-ACC} = \frac{10 \cdot (K_{CH-NOT-APP-UNAUTH-ACC} + K_{KETA-NOT-APP-UNAUTH-ACC} + K_{KXR-NOT-APP-UNAUTH-ACC})}{\text{Number of years since the monitoring started}}$$

different kinds of data from the three sensors, where this data is used in the sequence diagram “Comparison of data” in Fig. 19 for updating $K_{CH-SP-EPR-INFO}$.

In the sequence diagram in Fig. 19, “Component for calculating $K_{CH-SP-EPR-INFO}$” extracts all accesses to sensitive and private information in EPRs that have occurred in the period of one week backwards. The component also extracts all information items from INFO_LIST_1 and INFO_LIST_2 that both refer to Client H and the medical history of a person. Since information retrieved from the traditional media or the Internet will refer to patients by name, the different accesses are grouped with respect to patient names. In addition, duplicate accesses are removed, since we are not interested in how many times some information has been accessed, but rather whether it has been accessed or not. As can be seen in the sequence diagram, the different items of information retrieved from the traditional media or the Internet are grouped in the same way as for accesses to information in EPRs.

After having grouped the different data, we check for the different information items whether they match information that is retrieved when performing different accesses to information in EPRs. We use software to identify potential matches, while an employee at Client H performs a manual check of the potential matches to determine whether the sensitive and private information obtained from performing an access to information in an EPR is really the source of the information that has been retrieved from the traditional media or the Internet. When evaluating the potential matches, the employee needs to consider other potential sources for the information leakage, such as the patient itself. The employee also needs to consider whether the information retrieved from the traditional media or the Internet really refers to the same patient as the information obtained from an EPR does. If the employee is confident that the information from the EPR is the source, then the basic key indicator $K_{CH-SP-EPR-INFO}$ is incremented by one. In the end, the employee sends the updated basic key indicator to “Component for calculating $K_{CH-SP-EPR-INFO}$,” as illustrated in Fig. 18. The component stores the updated basic key indicator before sending it to “Component for calculating $K_{SP-EPR-INFO}$,” as illustrated in the sequence diagram in Fig. 17.
E. Key indicator designs for $K_{HSP-EPR-INFO}$ and its basic key indicators

The sequence diagram in Fig. 20 specifies how the key indicator $K_{HSP-EPR-INFO}$ is calculated, while the sequence diagram in Fig. 21 describes the calculation of the basic key indicator $K_{CH-HSP-EPR-INFO}$ at Client H. We use the same argument as the one given in Section VIII-A for not presenting sequence diagrams for the two other basic key indicators.

The sequence diagram in Fig. 21 shows that the basic key indicator $K_{CH-HSP-EPR-INFO}$ is updated each week. This sequence diagram is almost identical to the one in Fig. 18, while the sequence diagram “Comparison of data” in Fig. 22, which is referred to in Fig. 21, is almost identical to the one in Fig. 19. Thus, we do not give any further explanations for the two sequence diagrams.

F. Key indicator designs for $K_{ILL-ACC-SC}$ and its basic key indicators

The sequence diagram in Fig. 23 specifies how the key indicator $K_{ILL-ACC-SC}$ is calculated, while the sequence diagram in Fig. 24 describes the calculation of the basic key indicator $K_{CH-ILL-ACC-SC}$ at Client H. We use the same
Calculation of $K_{SP-INFO}$

KSP-INFO = \( \frac{10 \cdot (KCH-SP-INFO + KBTA-SP-INFO + KXR-SP-INFO)}{g187} \)

Number of years since the monitoring started

Fig. 17. The sequence diagram “Calculation of $K_{SP-INFO}$”

Calculation of $K_{CH-SP-INFO}$

KCH-SP-INFO = “The number of times since the start of the monitoring up to the end of the previous week that sensitive and private information about patients have been shared by health-care professionals with others and where this information have ended up in the traditional media or on the Internet”

Fig. 18. The sequence diagram “Calculation of $K_{CH-SP-INFO}$”
Comparison of data

- **Component for calculating KCH-SP-EPR-INFO**

  - **SP_ACC_LIST:** Extract all accesses at Client H in the period of one week backwards to sensitive and private information in EPRs based on ACC_LIST
  - **SP_ACC_GROUPS:** Remove duplicate accesses in SP_ACC_LIST (accesses to the same information) and group accesses wrt patient names (all patients with the same name is put in the same group)
  - **INFO_GROUPS:** Extract all information items from INFO_LIST_1 and INFO_LIST_2 that both refers to Client H and the medical history of people, and remove duplicate items before grouping information items wrt the names of the people (all people with the same name is put in the same group)

**Fig. 19. The sequence diagram “Comparison of data”**
Calculation of KHSP-EPR-INFO

Component for calculating KHSP-EPR-INFO

Sensor SCH-INFO-MRS
Sensor SCH-INFO-GP
EPR system
Employee at Client H
Component for calculating KHSP-EPR-INFO

Calculation of KHSP-EPR-INFO

KHSP-EPR-INFO = (10 · (KCH-HSP-EPR-INFO + KBTA-HSP-EPR-INFO + KXR-HSP-EPR-INFO)) / g
Number of years since the monitoring started

Fig. 20. The sequence diagram “Calculation of KHSP-EPR-INFO”

Calculation of KCH-HSP-EPR-INFO

KCH-HSP-EPR-INFO = "The number of times since the start of the monitoring up to the end of the previous week that highly sensitive and private information about patients have been shared by health-care professionals with others and where this information have ended up in the traditional media or on the Internet"

INFO_LIST_1: Items of information, where each item has been collected from the traditional media or the Internet by "Media retriever service for CH" in the period of one week backwards
INFO_LIST_2: Items of information, where each item is feedback from someone in the general public or information that he/she has collected from the Internet in the period of one week backwards

EPR system

Component for calculating KCH-HSP-EPR-INFO

ACo_LIST: All accesses at Client H in the period of one week backwards to information in EPRs

Fig. 21. The sequence diagram “Calculation of KCH-HSP-EPR-INFO”
Comparison of data

Employee at Client H

Component for calculating KCH-HSP-EPR-INFO

HSP_ACC_LIST: Extract all accesses at Client H in the period of one week backwards to highly sensitive and private information in EPRs based on ACC_LIST

HSP_ACC_GROUPS: Remove duplicate accesses in HSP_ACC_LIST (accesses to the same information) and group accesses wrt patient names (all patients with the same name is put in the same group)

INFO_GROUPS: Extract all information items from INFO_LIST_1 and INFO_LIST_2 that both refers to Client H and the medical history of people, and remove duplicate items before grouping information items wrt the names of the people (all people with the same name is put in the same group)

EPR system

loop (INFO_GROUP = 0, number of elements in INFO_GROUPS)

loop (INFO_GROUP = 0, number of elements in INFO_GROUPS)

opt

loop (X = 0, number of items in INFO_GROUP)

loop (Y = 0, number of items in HSP_ACC_GROUP)

Perform access Y

Info from EPR for access Y

Check whether there is a match between info from EPR for access Y and info item X

opt

[Match between info from EPR for access Y and info item X]

Info item X and info from EPR for access Y

Perform manual check to decide whether info from EPR for access Y is the source of info item X. Other possible sources are also considered during the check

opt

[Info from EPR for access Y is the source of info item X]

KCH-HSP-EPR-INFO = KCH-HSP-EPR-INFO + 1

Fig. 22. The sequence diagram “Comparison of data”
Calculation of $K_{ILL-ACC-SC}$

$$K_{ILL-ACC-SC} = \frac{10 \cdot (K_{CH-ILL-ACC-SC} + K_{STA-ILL-ACC-SC} + K_{XR-ILL-ACC-SC})}{\text{Number of years since the monitoring started}}$$

The sequence diagram in Fig. 24 shows that the basic key indicator $K_{CH-ILL-ACC-SC}$ is updated each week. The first thing that happens is that “Component for calculating $K_{CH-ILL-ACC}$” retrieves the value that was computed for the basic key indicator in the previous week. Afterwards, the component counts for each of the lost/stolen smart cards the number of accesses that have occurred between $TIME_1$ (the time the smart card’s owner used it the last time before noticing that it was missing) and $TIME_2$ (the time when the smart card was registered as missing). In the end, the component stores the basic key indicator $K_{CH-ILL-ACC-SC}$ and sends it to “Component for calculating $K_{ILL-ACC-SC}$,” as illustrated in the sequence diagram in Fig. 23.
Calculation of $K_{CH-ILL-ACC-SC}$

$K_{CH-ILL-ACC-SC} = \text{"The number of illegal accesses at Client H since the start of the monitoring up to the end of the previous week to highly sensitive and private information in EPRs from lost/stolen smart cards"}$

ACC_LIST: All accesses at Client H in the period of one week backwards to information in EPRs

MIS_SC_LIST: All smart cards at Client H that were registered as missing in the period of one week backwards

$\text{TOTAL\_NUM\_ILL\_ACC\_HSP} = 0$

$\text{loop ( MIS\_SC} = 0, \text{number of elements in MIS\_SC\_LIST )}$

Get the time TIME_1 for when the missing smart card MIS_SC was used the last time before its owner noticed that it was missing

Get the time TIME_2 for when the missing smart card MIS_SC was registered as missing and made unusable

From ACC_LIST create ILL\_ACC\_HSP\_LIST which is a list of all illegal accesses to highly sensitive and private info in EPRs that occurred between TIME_1 and TIME_2 from MIS_SC

$\text{NUM\_ILL\_ACC\_HSP} = 0$

$\text{loop ( ILL\_ACC\_HSP} = 0, \text{number of elements in ILL\_ACC\_HSP\_LIST )}$

$\text{NUM\_ILL\_ACC\_HSP} = \text{NUM\_ILL\_ACC\_HSP} + 1$

$\text{TOTAL\_NUM\_ILL\_ACC\_HSP} = \text{TOTAL\_NUM\_ILL\_ACC\_HSP} + \text{NUM\_ILL\_ACC\_HSP}$

$K_{CH-ILL-ACC-SC} = K_{CH-ILL-ACC-SC} \times \text{TOTAL\_NUM\_ILL\_ACC\_HSP}$

Sure $K_{CH-ILL-ACC-SC}$

Fig. 24. The sequence diagram “Calculation of $K_{CH-ILL-ACC-SC}$”
The key indicators $K_{PR-SP-EPR-INFO}$ and $K_{PR-HSP-EPR-INFO}$ are lower or higher than they should be

The key indicator $K_{KNOT-APP-UNAUTH-ACC}$ is lower or higher than it should be

The key indicator $K_{HSP-EPR-INFO}$ is lower or higher than it should be

The key indicator $K_{SP-EPR-INFO}$ is lower or higher than it should be

The key indicator $K_{ILLACC-SC}$ is lower or higher than it should be

**IX. Evaluate construct validity**

To evaluate whether the composite key indicators have construct validity, we re-do the risk analysis from Step 2.2 with the asset “Fulfillment of PBO-A8” replaced by the asset “Correctness of PBO-A8’.” We have established that the monitoring infrastructure described in Step 2–4 is suitable for monitoring the relevant part of business. With the designs of the key indicators specified in the previous step, we want to identify in this step whether the proposed implementation of the monitoring infrastructure results in any new unacceptable risks. More precisely, we want to identify unacceptable risks towards the correctness of the reformulated precise business objective that are the result of threats to criteria for construct validity that the different composite key indicators need to fulfill.

We evaluate the construct validity of the composite key indicators based on the criteria given in Section III-F. A high-level overview of the result of the risk analysis is given in the CORAS threat diagram in Fig. 25. In the referenced threat scenarios in Figs. 26 – 30, risk to the correctness of the different composite key indicators have been documented. For the key indicators $K_{PR-SP-EPR-INFO}$ and $K_{PR-HSP-EPR-INFO}$, Client H is of the opinion that their
The key indicators KPR-SP-EPR-INFO and KPR-HSP-EPR-INFO are lower or higher than they should be

R19: The key indicator KPR-SP-EPR-INFO is higher than it should be

R20: The key indicator KPR-HSP-EPR-INFO is higher than it should be

R21: The key indicator KPR-SP-EPR-INFO is lower than it should be

R22: The key indicator KPR-HSP-EPR-INFO is lower than it should be

The referenced threat scenario “The key indicators KPR-SP-EPR-INFO and KPR-HSP-EPR-INFO are lower or higher than they should be,” referred to in Fig. 25

correctness may be affected if the interval [0.1, 0.3] used to calculate the two key indicators is either too low or too high. This is an example of violation of the stability criterion, since the selection of the interval is the result of human decisions, i.e., expert judgments. For the two composite key indicators, no threats towards the definition and instrument validity of the composite key indicators are identified.

In the case of the key indicator KNOT-APP-UNAUTH-ACC, Client H is of the opinion that its correctness may be affected if the employees who classify unauthorized accesses as approved or not approved at X-ray and Blood test analysis are incompetent and fraudulent, respectively. Both these cases are examples of violation of the stability criterion, since the classification of unauthorized accesses as approved or not approved involves human decisions. Moreover, Client H is worried that the sensor SCH-ACC-INFO-EPR (represented as a non-human threat in Fig. 27) may be unstable with respect to logging of accesses to information in EPRs. This is an example of violation of the instrument validity criterion. Besides the stability and instrument validity criteria, definition validity should also be evaluated. In our case, we say that a key indicator has definition validity if its design is clear and unambiguous so that the key indicator can be implemented correctly. The only thing that is not clear and unambiguous with respect to the design of KNOT-APP-UNAUTH-ACC is how unauthorized accesses should be classified as approved or not approved. Since this has already been covered during the evaluation of the stability criterion, we do not pursue this issue further.

In the case of the key indicators KSP-EPR-INFO and KHSP-EPR-INFO, Client H is worried that the correctness of KSP-EPR-INFO may be affected if employees at Blood test analysis either fail to identify data leakages of sensitive and private information from EPRs or incorrectly classify sensitive and private information obtained from EPRs as the sources of data leakages, when no such data leakages have occurred. Moreover, Client H is worried that
The key indicator \( K_{\text{NOT-APP-UNAUTH-ACC}} \) is lower or higher than it should be

Fig. 27. The referenced threat scenario “The key indicator \( K_{\text{NOT-APP-UNAUTH-ACC}} \) is lower or higher than it should be,” referred to in Fig. 25

the correctness of \( K_{\text{HSP-EPR-INFO}} \) may be affected if employees at X-ray commit the same errors when it comes to highly sensitive and private information in EPRs. Both these cases are examples of violation of the stability criterion. In the case of instrument validity, Client H is worried that the media retriever services employed by Blood test analysis and X-ray are not able to collect the information necessary for detecting data leakages. Client H is also worried that the two composite key indicators may violate the definition validity criterion. The design specifications of the two composite key indicators are not clear and unambiguous with respect to how data leakages should be identified. In both specifications, it is up to the employees investigating potential data leakages to decide. Since this has already been covered during the evaluation of the stability criterion, we do not pursue this issue further.

In the case of the key indicator \( K_{\text{ILL-ACC-SC}} \), Client H is worried that its correctness may be affected by health-
The referenced threat scenario “The key indicator $K_{\text{HSP-EPR-INFO}}$ is lower or higher than it should be,” referred to in Fig. 25.

Fig. 28. The referenced threat scenario “The key indicator $K_{\text{HSP-EPR-INFO}}$ is lower or higher than it should be,” referred to in Fig. 25.

care professionals not having a perfect recollection of when they used their smart cards the last time before losing it. By not having a perfect recollection, accesses to information in EPRs may incorrectly be classified as legal or illegal accesses. This is an example of violation of the stability criterion. For the composite key indicator, no threats towards the definition and instrument validity of the composite key indicator are identified.

In Table XII the risks $R19 - R30$ have been plotted according to their likelihoods and consequences. As we can see from the table, the two risks $R26$ and $R28$ are unacceptable. This means that all the composite key indicators with the exceptions of $K_{\text{SP-EPR-INFO}}$ and $K_{\text{HSP-EPR-INFO}}$ have construct validity. As a first step to making these two risks acceptable, Client H finds it necessary to gain more knowledge on the suitability of the two media retriever services. If the two risks do not become acceptable as a result of this, further treatment will be necessary in order
The key indicator $K_{SP-EPR-INFO}$ is lower or higher than it should be

The employee at Blood test analysis fails to classify sensitive and private information from a patient’s EPR as the source of information retrieved from the traditional media or the Internet

A data leakage of sensitive and private information that has not occurred at Blood test analysis is identified

Lack of quality control

[Unlikely]

0.2

The key indicator $K_{SP-EPR-INFO}$ is higher than it should be

A data leakage of sensitive and private information that has occurred at X-ray is not identified

Lack of quality control

[Unlikely]

0.2

The media retriever service of Blood test analysis fails to collect information relevant for identifying data leakages of sensitive and private information

A data leakage of sensitive and private information that has occurred at Blood test analysis is not identified

The employee at Blood test analysis incorrectly classifies sensitive and private information from a patient’s EPR as the source of information retrieved from the traditional media or the Internet

Lack of quality control

[Unlikely]

0.2

The key indicator $K_{SP-EPR-INFO}$ is lower than it should be

A data leakage of sensitive and private information that has not occurred at X-ray is not identified

Lack of quality control

[Unlikely]

0.2

The media retriever service of X-ray fails to collect information relevant for identifying data leakages of sensitive and private information

A data leakage of sensitive and private information that has occurred at X-ray is not identified

Lack of quality control

[Unlikely]

0.2

The key indicator $K_{SP-EPR-INFO}$ is lower or higher than it should be

Fig. 29. The referenced threat scenario “The key indicator $K_{SP-EPR-INFO}$ is lower or higher than is should be,” referred to in Fig. 25 for the two key indicators $K_{SP-EPR-INFO}$ and $K_{HSP-EPR-INFO}$ to achieve construct validity. Such treatments may involve replacing the media retriever services of Blood test analysis and X-ray, or introducing an additional media retriever service for each of the two hospitals. In the latter case this means that Blood test analysis and X-ray will each identify data leakages based on information which combines results from two media retriever services.
Fig. 30. The referenced threat scenario “The key indicator $K_{\text{ILL-ACC-SC}}$ is lower or higher than is should be,” referred to in Fig. 25

### TABLE XII

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Consequence</th>
<th>Insignificant</th>
<th>Minor</th>
<th>Moderate</th>
<th>Major</th>
<th>Catastrophic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rare</td>
<td></td>
<td>$R_{25}$, $R_{27}$, $R_{29}$</td>
<td>$R_{1}$&amp;$R_{2}^\prime$</td>
<td>$R_{3}$&amp;$R_{4}$’’, $R_{8}$’’, $R_{10}$’’, $R_{15}$&amp;$R_{17}$</td>
<td>$R_{12}$, $R_{16}$&amp;$R_{18}$</td>
<td></td>
</tr>
<tr>
<td>Unlikely</td>
<td></td>
<td>$R_{23}$</td>
<td>$R_{1}$&amp;$R_{2}$’’, $R_{30}$</td>
<td>$R_{3}$&amp;$R_{4}$’’, $R_{8}$’’, $R_{10}$’’, $R_{24}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possible</td>
<td></td>
<td>$R_{6}$, $R_{7}$, $R_{19}$, $R_{20}$</td>
<td>$R_{1}$&amp;$R_{2}$”’, $R_{9}$, $R_{11}$, $R_{21}$, $R_{22}$</td>
<td>$R_{26}$, $R_{28}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Likely</td>
<td>$R_{13}$</td>
<td>$R_{5}$, $R_{14}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
X. RELATED WORK

To the best of our knowledge, there exists no other method for the design of valid key indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of key indicators. There is a tool-framework called Mozart [23] that uses a model-driven approach to create monitoring applications that employ key performance indicators. We do not focus on the implementation of key indicators, but we specify what is needed for implementing them. The work in [23] also differs from our work by not designing indicators from scratch, but by mining them from a data repository during the design cycle.

An important part of our method is the assessment of the validity of the key indicators we design. Our approach to assessing validity is inspired by research conducted within the software engineering domain. As previously explained, there is however no agreement upon what constitutes a valid software metric [8]. A number of the software metrics validation approaches advocate the use of measurement theory [24][25][26] in the validation (see e.g., [9][27][28]). Measurement theory is a branch of applied mathematics that is useful in measurement and data analysis. The fundamental idea of this theory is that there is a difference between measurements and the attribute being measured. Thus, in order to draw conclusions about the attribute, there is a need to understand the nature of the correspondence between the attribute and the measurements. In [29], an approach that relies on measurement theory for the validation of indicators is presented. This approach uses measurement theory to validate the meaningfulness of IT security risk indicators.

Measurement theory has been criticized of being too rigid and restrictive in a practical measurement setting. Briand et al. [27] advocate a pragmatic approach to measurement theory in software engineering. The authors show that even if their approach may lead to violations of the strict prescriptions and proscriptions of measurement theory, the consequences are small compared to the benefits. Another approach that takes a pragmatic approach to measurement theory is [28]. Here, the authors propose a framework for evaluating software metrics. The applicability of the framework is demonstrated by applying it on a bug count metric.

There exist also approaches that assess the validity of specific sets of key indicators. For instance, in [30] the validity of indicators of firm technological capability is assessed, while the validity of indicators of patent value is assessed in [31].

There are several approaches that focus on measuring the achievement of goals. One example is COBIT [32], which is a framework for IT management and IT governance. The framework provides an IT governance model that helps in delivering value from IT and understanding and managing the risks associated with IT. In the governance model, business goals are aligned with IT goals, while metrics, in the form of leading and lagging indicators [33], and maturity models are used to measure the achievement of the IT goals. In our approach we do not focus on the value that the use of IT has with respect to the business objectives. On the other hand, the risk that the use of IT has with respect to the business objectives is important. In our context, IT is relevant in the sense of providing the infrastructure necessary for monitoring the part of business that needs to fulfill the business objectives. In Step 6 of our method we identify risks that may result from the use of the monitoring infrastructure with respect to the business objectives.

Another way to measure the achievement of goals is by the use of the Goal-Question-Metric [34][35] (GQM) approach. Even though GQM originated as an approach for measuring achievement in software development, it can also be used in other contexts where the purpose is to measure achievement of goals. In GQM, business goals are used to drive the identification of measurement goals. These goals do not necessarily measure the fulfillment of the business goals, but they should always measure something that is of interest to the business. Each measurement goal is refined into questions, while metrics are defined for answering each question. No specific method, beyond reviews, is specified for validating whether the correct questions and metrics have been identified. The data provided by the metrics are interpreted and analyzed with respect to the measurement goal in order to conclude whether it is achieved or not. One of the main differences between our method and GQM is that we characterize precisely what it means to achieve a goal/objective. In GQM, however, this may be a question of interpretation.

In the literature, key indicators are mostly referred to in the context of measuring business performance. There exist numerous approaches to performance measurement. Some of these are presented in [36]. Regardless of the approach being used, the organization must translate their business objectives/goals into a set of key performance indicators in order to measure performance. An approach that is widely used [37] is balanced scorecard [5]. This approach translates the company’s vision into four financial and non-financial perspectives. For each perspective a set
of business objectives (strategic goals) and their corresponding key performance indicators are identified. However, the implementation of a balanced scorecard is not necessarily straightforward. In [38], Neely and Bourne identify several reasons for the failure of measurement initiatives such as balanced scorecards. One problem is that the identified measures do not measure fulfillment of the business objectives, while another problem is that measures are identified without putting much thought into how the data must be extracted in order to compute the measures. The first problem can be addressed in Step 4 of our method, while the second problem can be addressed in Step 3 and Step 5 of our method. In Step 3 we identify the sensors to be deployed in the relevant part of business, while in Step 5 we present the kinds of data that needs to be extracted from these sensors in order to compute the measures.

Much research has been done in the field of data quality. The problem of data quality is also recognized within the field of key indicators [39][40]. In [41] a survey on how data quality initiatives are linked with organizational key performance indicators in Australian organizations is presented. This survey shows that a number of organizations do not have data quality initiatives linked to their key indicators. Data quality should be taken into account when designing key indicators, since the use of key indicators based on poor quality data may lead to bad business decisions, which again may greatly harm the organization.

In [42][43] the problem of key indicators computed from uncertain events is investigated. The motivation for this work is to understand the uncertainty of individual key indicators used in business intelligence. The authors use key indicators based on data from multiple domains as examples. In these papers a model for expressing uncertainty is proposed, and a tool for visualizing the uncertain key indicators is presented.

XI. CONCLUSION

In [1] we presented the method ValidKI (Valid Key Indicators) for designing key indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of key indicators. ValidKI facilitates the design of a set of key indicators that is valid with respect to a business objective. In this report we have presented the improved and consolidated version of the method.

To the best of our knowledge, there exists no other method for the design of valid key indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of key indicators. The applicability of our method has been demonstrated on a large, realistic example case addressing the use of electronic patient records in a hospital environment.

Even though ValidKI has been demonstrated on a large, realistic example case there is still a need to apply ValidKI in a real-world industrial setting in order to evaluate properly to what extent it has the characteristics specified in the introduction. By applying ValidKI in such a setting we will for instance gain more knowledge regarding whether it is time and resource efficient.

ACKNOWLEDGMENTS

The research on which this report describes has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER and NESSoS projects, both funded from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreements FP7-216917 and FP7-256980, respectively.

REFERENCES


Chapter 10

Paper B: Using indicators to monitor risk in interconnected systems: How to capture and measure the impact of service dependencies on the quality of provided services
Using Indicators to Monitor Risk in Interconnected Systems: How to Capture and Measure the Impact of Service Dependencies on the Quality of Provided Services

Author(s)
Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen
Using Indicators to Monitor Risk in Interconnected Systems: How to Capture and Measure the Impact of Service Dependencies on the Quality of Provided Services

VERSION
Final version

DATE
2012-10-01

AUTHOR(S)
Ole Skjelkvåle Løigaarden, Atle Refsdal, and Ketil Stålen

CLIENT(S)
Research Council of Norway

PROJECT NO.
98245

CLIENT'S REF.
180052/510

NUMBER OF PAGES/APENDICES:
126/4

ABSTRACT
Interconnected systems are collections of systems that interact through the use of services. Their often complex service dependencies and very dynamic nature make them hard to analyze and predict with respect to quality attributes. In this report we put forward a method for the capture and monitoring of impact of service dependencies on the quality of provided services. The method is divided into four main steps focusing on documenting the interconnected systems and the service dependencies, establishing the impact of service dependencies on risk to quality of provided services, identifying measurable indicators for dynamic monitoring, and specifying their design and deployment, respectively. We illustrate the method in an example-driven fashion based on a case study from the domain of power supply.

PREPARED BY
Ole Skjelkvåle Løigaarden

CHECKED BY
Mass Soldal Lund

APPROVED BY
Bjarn Skjellevag, Research Director

REPORT NO.
SINTEF A22301

ISBN
978-82-14-05279-4

CLASSIFICATION
Unrestricted

CLASSIFICATION THIS PAGE
Unrestricted
Contents

1 Introduction 5

2 Basic terminology and definitions 6
2.1 System of systems and related concepts 6
2.2 Risk and related concepts 8

3 Methodological approach 9
3.1 Step 1: Document interconnected systems 9
3.1.1 Step 1.1: Model interconnected systems 9
3.1.2 Step 1.2: Capture service dependencies 10
3.1.3 Step 1.3: Capture trust relations 12
3.2 Step 2: Analyze the impact of service dependencies on risk to quality of provided services 15
3.2.1 Step 2.1: Identify quality assets to be analyzed 15
3.2.2 Step 2.2: Construct high-level threat diagrams of the impact of service dependencies on identified quality assets 15
3.2.3 Step 2.3: Construct detailed threat diagrams of the impact of service dependencies on identified quality assets 20
3.3 Step 3: Identify indicators for interconnected systems 22
3.3.1 Step 3.1: Identify risks to be monitored 22
3.3.2 Step 3.2: Identify relevant indicators for the risks to be monitored 22
3.4 Step 4: Specify design and deployment of identified indicators for interconnected systems 23
3.4.1 Step 4.1: Specify design of indicators for risk monitoring 23
3.4.2 Step 4.2: Specify deployment of indicators for risk monitoring 24

4 Demonstration of Step 1: Document interconnected systems 24
4.1 Step 1.1: Model interconnected systems 24
4.2 Step 1.2: Capture service dependencies 27
4.3 Step 1.3: Capture trust relations 27

5 Demonstration of Step 2: Analyze the impact of service dependencies on risk to quality of provided services 27
5.1 Step 2.1: Identify quality assets 27
5.2 Step 2.2: Construct high-level threat diagrams of the impact of service dependencies on identified quality assets 30
5.3 Step 2.3: Construct detailed threat diagrams of the impact of service dependencies on identified quality assets 30

6 Demonstration of Step 3: Identify indicators for interconnected systems 41
6.1 Step 3.1: Identify risks to be monitored 41
6.2 Step 3.2: Identify relevant indicators for the risks to be monitored 42

7 Demonstration of Step 4: Specify design and deployment of identified indicators for interconnected systems 42
7.1 Step 4.1: Specify design of indicators for risk monitoring 42
7.2 Step 4.2: Specify deployment of indicators for risk monitoring 44

8 Related work 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>48</td>
</tr>
<tr>
<td><strong>Conclusion</strong></td>
<td></td>
</tr>
<tr>
<td>References</td>
<td>49</td>
</tr>
<tr>
<td>A Assets to be analyzed for provided services</td>
<td>51</td>
</tr>
<tr>
<td><strong>B Schematic construction of threat diagrams for provided services</strong></td>
<td>53</td>
</tr>
<tr>
<td>B.1 Control instructions service provided to Public telecom system</td>
<td>53</td>
</tr>
<tr>
<td>B.2 Electricity service provided to Distribution line 2</td>
<td>53</td>
</tr>
<tr>
<td>B.3 Electricity service provided to Distribution line 3</td>
<td>54</td>
</tr>
<tr>
<td>B.4 Electricity service provided to Transmission line</td>
<td>55</td>
</tr>
<tr>
<td><strong>C Capture and measure impact of service dependencies on quality assets of provided services</strong></td>
<td>58</td>
</tr>
<tr>
<td>C.1 Control instructions service provided to Public telecom system</td>
<td>58</td>
</tr>
<tr>
<td>C.1.1 Detailed threat diagrams</td>
<td>58</td>
</tr>
<tr>
<td>C.1.2 Relevant indicators for risk monitoring</td>
<td>65</td>
</tr>
<tr>
<td>C.1.3 Design and deployment of indicators for risk monitoring</td>
<td>67</td>
</tr>
<tr>
<td>C.2 Electricity service provided to Distribution line 2</td>
<td>72</td>
</tr>
<tr>
<td>C.2.1 Detailed threat diagrams</td>
<td>72</td>
</tr>
<tr>
<td>C.2.2 Relevant indicators for risk monitoring</td>
<td>80</td>
</tr>
<tr>
<td>C.2.3 Design and deployment of indicators for risk monitoring</td>
<td>80</td>
</tr>
<tr>
<td>C.3 Electricity service provided to Distribution line 3</td>
<td>86</td>
</tr>
<tr>
<td>C.3.1 Detailed threat diagrams</td>
<td>86</td>
</tr>
<tr>
<td>C.3.2 Relevant indicators for risk monitoring</td>
<td>87</td>
</tr>
<tr>
<td>C.3.3 Design and deployment of indicators for risk monitoring</td>
<td>87</td>
</tr>
<tr>
<td>C.4 Electricity service provided to Transmission line</td>
<td>88</td>
</tr>
<tr>
<td>C.4.1 Detailed threat diagrams</td>
<td>88</td>
</tr>
<tr>
<td>C.4.2 Relevant indicators for risk monitoring</td>
<td>98</td>
</tr>
<tr>
<td>C.4.3 Design and deployment of indicators for risk monitoring</td>
<td>100</td>
</tr>
<tr>
<td><strong>D Monitor risk values based on identified indicators</strong></td>
<td>104</td>
</tr>
<tr>
<td>D.1 Likelihood calculation rules</td>
<td>104</td>
</tr>
<tr>
<td>D.2 Sensor data service provided to Public telecom system</td>
<td>107</td>
</tr>
<tr>
<td>D.3 Control instructions service provided to Public telecom system</td>
<td>110</td>
</tr>
<tr>
<td>D.4 Electricity services provided to Distribution line 2 and Distribution line 3</td>
<td>114</td>
</tr>
<tr>
<td>D.5 Electricity service provided to Transmission line</td>
<td>117</td>
</tr>
</tbody>
</table>
Using Indicators to Monitor Risk in Interconnected Systems: How to Capture and Measure the Impact of Service Dependencies on the Quality of Provided Services

Olav Skjelkvåle Ligaarden\textsuperscript{1,2}, Atle Refsdal\textsuperscript{1}, and Ketil Stølen\textsuperscript{1,2}

\textsuperscript{1} Department for Networked Systems and Services, SINTEF ICT, Norway
\textsuperscript{2} Department of Informatics, University of Oslo, Norway

Abstract

Interconnected systems are collections of systems that interact through the use of services. Their often complex service dependencies and very dynamic nature make them hard to analyze and predict with respect to quality attributes. In this report we put forward a method for the capture and monitoring of impact of service dependencies on the quality of provided services. The method is divided into four main steps focusing on documenting the interconnected systems and the service dependencies, establishing the impact of service dependencies on risk to quality of provided services, identifying measurable indicators for dynamic monitoring, and specifying their design and deployment, respectively. We illustrate the method in an example-driven fashion based on a case study from the domain of power supply.

1 Introduction

In today’s business environment, businesses/organizations co-operate with each other by providing and/or requiring different kinds of services. The systems facilitating such co-operation are often so-called system of systems (SoS). An SoS may be thought of as a kind of “super system” comprising a set of interconnected systems that work together towards some common goal.

An SoS is challenging from a quality perspective. Firstly, the provided services may require other services in order to function. Such requirements result in so-called service dependencies. Change in the quality attributes of one service may easily cause the quality attributes of its dependent services to change as well. Secondly, the different systems may be under different managerial control and within different jurisdictions. For the systems that are outside our control, we have limited knowledge of their risks, structure, and behavior. Thirdly, such a large number of systems, controlled and operated by different parties, evolve rapidly in a manner that may be difficult to predict.

To cope with this situation we propose the use of detailed dependency models to capture the impact of service dependencies, trust relations as a basis for analysis in the case of insufficient documentation, and monitoring to cope with evolution. Our main result is a method facilitating the set-up of such monitoring. The method is divided into four steps. Service dependencies and
trust relations are identified and documented in the first step. In the second step we conduct a risk analysis to capture the impact of service dependencies on risk to quality of a set of provided services. These services will not be provided according to their quality requirements if services that depend on are not delivered according to their quality requirements. The focus of the risk analysis is therefore on assessing how service dependencies may result in risks, and how these risks may result in the provided services not being delivered according to their quality requirements. During this step, the identified trust relations are used when analyzing service dependencies involving systems of which we have insufficient documentation. In the third step we identify the risks to be monitored, as well as measurable indicators for monitoring their risk values. In the fourth and final step we specify how these indicators should be designed, i.e., how they should be calculated, and deployed in the interconnected systems, i.e., how data needed in the calculations should be extracted and transmitted within the interconnected systems in question. The result of applying the method is a risk picture parameterized by indicators, each defined by design and deployment specifications.

The rest of the report is organized as follows: in Section 2 we introduce basic terminology and definitions. Section 3 presents the methodological approach, while the four steps of the approach are demonstrated on an example case from the domain of power supply in Sections 4–7. In Section 8 we present related work, while we conclude and indicate further research in Section 9. For the sake of simplicity, the approach is only demonstrated for one provided service in Sections 5–7. In Appendices A–C we demonstrate the approach on the remaining provided services. In Appendix D we show how to monitor risk values for the different provided services based on indicators.

2 Basic terminology and definitions

In this section we provide basic terminology, definitions, and conceptual models for system of systems, risk, and related concepts.

2.1 System of systems and related concepts

As already explained, an SoS is basically a set of interconnected systems that work together towards some common goal. Our definition of SoS is based on the definitions of [1] and [2]. We define SoS as follows: “A system of systems (SoS) is a set or arrangement of systems that are related or connected to fulfill common goals. The different systems may be controlled, operated, and maintained by different parties and within different jurisdictions. The loss of any system may seriously impact the other systems and the process of fulfilling the common goals.”

An SoS may arise naturally from the interconnection of individual systems, or it may be built specifically for the purpose of achieving goals that the individual systems cannot achieve alone. An example of the former is the interconnection of critical infrastructures, while a sensor network, constructed for the purpose of gathering low-level data to be aggregated, is an example of the latter.

We focus on SoS where the systems interact through the use of services. In Figure 1 is a conceptual model, in the form of a UML [3] class diagram, relating system, system of systems, and other concepts. The associations between the different concepts have cardinalities that specify how many instances of one concept that may be associated to an instance of another concept. The filled diamond specifies composition, while the hollow diamond specifies aggregation.

As shown in Figure 1, a System of Systems consists of at least two Systems. In this report, we divide a SoS into two parts; a Target and a Environment. The target consists of one or more Target Systems, and it is the fragment of the SoS which is controlled by the client enterprise
on whose behalf our method is applied. We refer to this client as the Trustor. The target depends on the rest of the SoS that is controlled by other enterprises that may be thought of as Trustees of our client enterprise. We refer to the rest of the SoS as the environment of the target. The environment consists of a number of Environment Systems; each controlled by one of the trustees.

In this report, we only consider services where each service is provided by one system and required by another. Each service represents the exchange of some commodity (electricity, information, etc.). A Service Dependency describes a relationship between a service provided by a system and services required by the system. A service depends on other services if it requires the other services in order to be provided according to its requirements. In Figure 1, Service Dependencies are shown by the use of an association class. Service dependencies help us to better understand the importance of the individual services that are provided and required by the different systems in the SoS.

Typically, a service will have one or more Required service levels. Each required service level describes a requirement to one area of service scope. Availability, integrity, etc., are all examples of areas of service scope. The different required service levels may for instance be specified in a service-level agreement. Thus, one or more Required Service Levels are associated with each service. For each required service level, the Trustor may have a certain amount of trust in that the service delivers the required level of service. Inspired by [4, 5], Lysemose et al. [6] defines trust as “the subjective probability by which an actor (the trustor) expects that another entity (the trustee) performs a given transition on which its welfare depends.” The level of trust may vary from 0 (complete distrust) to 1 (complete trust). In our case, trust assessment is only of relevance for required service levels associated with services provided by trustees’ environment systems to the trustor’s target systems. Trust is discussed in more detail in Section 3.1.3.
2.2 Risk and related concepts

Figure 2 shows a conceptual model for risk and closely related concepts. A Risk involves an Unwanted Incident, such as “System operator is unable to control and operate the power plant.” The unwanted incident may occur with a certain Likelihood. When the incident occurs, an Asset will be damaged (and its value reduced). This is the Consequence of the risk. An asset is owned by a Trustor and it is something of value that the trustor seeks to protect. It can be a physical thing, e.g., “Power plant,” or conceptual, e.g., “Reputation of trustor.” Since the consequence of an incident depends on the particular asset in question, the same incident may have different consequences for different assets.

By conducting a risk analysis we obtain a Risk Picture, consisting of zero or more risks, for the Target of analysis, i.e., the subject of the risk analysis. The Target in Figure 2 is the same as the Target in Figure 1. This is also true for the Environment. In [7], the environment of the target is defined as “the surrounding things of relevance that may affect or interact with the target; in the most general case, the rest of the world.” In our case, the environment of the target is limited to those systems of the trustees that are of relevance to the risk analysis.

In order to choose and prioritize between treatments, we assign a Risk Value to each risk. A risk function calculates the risk value by taking the likelihood of the unwanted incident and its consequence for the asset in question as input. Typically, likelihood is measured in terms of frequency or probability, while the measure of consequence depends on the asset in question.

Zero or more Indicators may be used to measure likelihood and consequence values. Hammond et al. [8] defines indicator as “something that provides a clue to a matter of larger significance or makes perceptible a trend or phenomenon that is not immediately detectable.” For example, an unexpected rise in the traffic load of a web server may signal a denial of service attack in progress. Thus, the significance of an indicator extends beyond what is actually measured to a larger phenomenon of interest. Moreover, an indicator is either basic or composite. Thus, an abstract class (name in italic) is used to represent Indicator in the conceptual model. By Basic Indicator we mean a measure such as the number of times a specific event generated by the ICT infrastructure has been observed within a given time interval, the average time between each generation of a specific event, the load on the network at a particular point in time, or similar. A Composite Indicator is the aggregation of two or more basic indicators.
Step 1 – Document interconnected systems
  1.1 – Model interconnected systems
  1.2 – Capture service dependencies
  1.3 – Capture trust relations

Step 2 – Analyze the impact of service dependencies on risk to quality of provided services
  2.1 – Identify quality assets
  2.2 – Construct high-level threat diagrams of the impact of service dependencies on identified quality assets
  2.3 – Construct detailed threat diagrams of the impact of service dependencies on identified quality assets

Step 3 – Identify indicators for interconnected systems
  3.1 – Identify risks to be monitored
  3.2 – Identify relevant indicators for the risks to be monitored

Step 4 – Specify design and deployment of identified indicators for interconnected systems
  4.1 – Specify design of indicators for risk monitoring
  4.2 – Specify deployment of indicators for risk monitoring

Figure 3: Overview of the methodological approach

3 Methodological approach

An overview of the methodological approach is presented in Figure 3. In the following we describe each of the four main steps as well as their sub-steps in terms of a detailed guideline. Throughout this section we exemplify different steps of the method. It should be noticed that the examples presented in this section are not used in the continuation of this report.

As already explained in Section 2, our intended client enterprise corresponds to the trustor in Figure 2. The trustor controls a fragment of the SoS which we refer to as the target. The target depends on the rest of the SoS that is controlled by other enterprises that may be thought of as trustees of our client enterprise. Our task is to establish a dynamic risk picture that captures the impact of service dependencies on risk to the quality of the services that the trustor provides to the trustees’ systems.

The methodological approach presented in this section is closely related to the method ValidKI [9] (Valid Key Indicators). ValidKI is a method for designing indicators to monitor the fulfillment of business objectives with particular focus on quality and ICT-supported monitoring of indicators. The ValidKI method is particularly relevant for further detailing of Step 3 and Step 4 as presented below. In Step 3, ValidKI supports the identification of indicators, while it supports the specification of the design and the deployment of indicators in Step 4.

3.1 Step 1: Document interconnected systems

3.1.1 Step 1.1: Model interconnected systems

Objective: Model the interconnected systems.

Rationale: To capture the impact of service dependencies on risk to quality of provided services, we need to document the services interactions between the different interconnected systems. In particular, it is essential to understand the dependencies between the target and the target’s environment, i.e., the interconnected systems that are not controlled by the trustor. We also need to document the requirements to the different services. We are only concerned with the impact of services on risk when they are not delivered according to requirements.
How conducted: A target model is created by the analysis team based on input documentation provided by the trustor. The target model describes the systems of the target as well as the systems in the target’s environment. It also captures the systems’ service interactions and the required service levels of the different services. Each required service level is specified for one area of service scope. We can for instance specify the required level of availability, integrity, etc., for the same service.

Input documentation: The trustor provides information on the interconnected systems, their service interactions, and the requirements, in the form of required levels of service, for each service.

Output documentation: A target model documenting:

- the systems of the target and its environment;
- the service interactions between the systems; and
- the required service levels for each service.

Modeling guideline: The interconnected systems are modeled in the form of a graph, as illustrated by Figure 4. The system elements (vertices) in the graph represent systems, while service relations (edges) represent interactions in the form of services. The bold rectangular container with rounded corners separates the target from its environment. Each system element is annotated with the party controlling and operating the system represented by the element, while each service relation is annotated with the service in question and its required levels of service. In Figure 4 this has only been shown for two service relations, in order to save space. For one of the service relations, a required service level has been specified for one area of service scope, while required service levels have been specified for two areas of service scope for the other service. Here, $A$ stands for availability, while $I$ stands for integrity.

The source of a service relation represents the provider of the service, while the target of the relation represents the consumer of the service. A system may need to consume services in order to provide other services. If one system provides two or more services to another system, then the model is a multigraph, i.e., a graph which allows multiple edges, meaning edges with the same pair of source and target vertices.

3.1.2 Step 1.2: Capture service dependencies

Objective: Identify and document service dependencies within the interconnected systems.

Rationale: In Step 1.1 we documented the service interactions between the different systems. In this step we identify the service dependencies resulting from the interactions. This enables us to analyze the impact of service dependencies on risk to quality of provided services.

How conducted: The target model from Step 1.1 is annotated with service dependencies, based on input documentation provided by the trustor. The annotated model shows how provided services depend on required services.
Figure 4: Target model
Input documentation:
- The target model from Step 1.1.
- The trustor provides information on the relations between required and provided services for the different systems documented in the target model.

Output documentation: The target model from Step 1.1 annotated with service dependencies.

Modeling guideline: Figure 5 shows the target model in Figure 4 annotated with service dependency constructs. The constructs describe dependencies between the provided and the required services of the systems. Dependencies between required and provided services are combined with “and” (\(\land\)) or “or” (\(\lor\)) operators. For an operator we refer to services that enter the operator as incoming, while we refer to services that leave the operator as outgoing. The meaning of the “and” operator is that all the incoming services are required to provide each of the outgoing services, while the meaning of the “or” operator is that only one of the incoming services is required to provide each of the outgoing services. Operators may be combined to express dependencies that cannot be expressed by a single operator alone. This has not been exemplified in Figure 5. For examples of this, we refer to Figure 11 on page 28.

Figure 5 also shows examples of service dependency constructs that do not rely on operators for expressing dependencies. If only one service is required to provide one or more services, then it is of course not necessary to use “and” or “or” operators to describe the dependencies.

3.1.3 Step 1.3: Capture trust relations

Objective: Document the trustor’s trust in the required levels of services being delivered by its trustees.

Rationale: A trustor will normally not have detailed knowledge of the interior of systems owned by its trustees. Moreover, they may be changed and updated in a manner not controlled by the trustor. Hence, services provided by environment systems are difficult to analyze due to lack of documentation as well as control. To cope with this lack of knowledge we capture trust levels with respect to the failure of environment systems to provide their services with the required service levels. Each trust level states the degree to which the trustor trusts the required service level of a service to be delivered by the environment system of a trustee.

How conducted: The target model from Step 1.2 is annotated with trust relations. Each trust relation relates a trust level (in the interval \([0, 1]\)) determined by the trustor to a required service level of a service provided by an environment system to a target system.

Input documentation: The target model from Step 1.2.

Output documentation: The target model from Step 1.2 annotated with trust relations.

Modeling guideline: Figure 6 shows the target model in Figure 5 annotated with trust relations. The trust relations are shown with dotted clouds. Each cloud is assigned to a required service level of a service provided by an environment system to a target system.
Figure 5: Target model annotated with service dependencies
Figure 6: Target model annotated with trust relations
3.2 Step 2: Analyze the impact of service dependencies on risk to quality of provided services

3.2.1 Step 2.1: Identify quality assets to be analyzed

Objective: Identify the quality assets for which impact of service dependencies should be analyzed.

Rationale: The trustor wants to protect the quality of the services provided to its trustees, i.e., ensure that they are provided according to their required service levels. By identifying quality assets we restrict the identification of risks caused by service dependencies to only those risks that may harm the quality of the services provided by the trustor to its trustees. By doing so, we ensure that the available time and resources are spent identifying the most critical and important risks for the trustor in question.

How conducted: For each provided service, the trustor identifies the quality assets for which protection is required. A quality asset is identified for each area of service scope of a provided service for which a required service level has been defined. The value of a quality asset is reduced if the service level becomes less than the required service level.

Input documentation: Target model from Step 1.3.

Output documentation: A list of quality assets for each provided service.

3.2.2 Step 2.2: Construct high-level threat diagrams of the impact of service dependencies on identified quality assets

Objective: Achieve an initial high-level understanding of the impact of service dependencies on the identified quality assets by schematically constructing threat diagrams from the target model.

Rationale: In order to conduct a detailed analysis of the impact of service dependencies on risk to quality of provided services, we first establish an initial high-level understanding of how the failure of individual systems to deliver their services according to requirements may lead to the failure of other individual systems to deliver their services according to requirements. Moreover, we establish how this eventually may lead to unwanted incidents that harm the identified quality assets. Such an initial high-level understanding is achieved by schematically constructing a threat diagram for each provided service.

How conducted: Figure 7 presents a threat diagram that provides an initial overview of how the quality asset “Availability of Service 5 delivered to System 5” may be harmed if the different services represented by the referring threat scenarios are not delivered according to their required service levels. The threat diagram has been schematically constructed from the target model in Figure 6.

We use CORAS [7], which is a model-driven approach to asset-oriented risk analysis, for the modeling and analysis of risk. The threat diagram is expressed in the CORAS language. The referring threat scenarios, vulnerabilities, and the referring unwanted incident have been given names following the conventions “Service X, Z, and Y not delivered according to requirements,” “Service X depends on Service Y,” and “Incident with impact on the A,” (where A is the name...
Figure 7: Threat diagram, constructed schematically from the target model in Figure 6, which provides a high-level outline of the impact of service dependencies on the quality asset “Availability of Service 5 delivered to System 5”
of the asset) respectively. It can also be seen that the vulnerability names only describe direct dependencies between services. Indirect dependencies may be identified by consulting the target model.

For all leads-to relations in the threat diagram, the source and target of the relation is an out-gate and in-gate, respectively. The gates are connected to referring threat scenarios and unwanted incidents. Moreover, the source of each impacts relation is an out-gate, where the out-gate is connected to a referring unwanted incident. In-gates and out-gates are explained in more detail in Step 2.3 of the demonstration of the methodological approach on the example case.

Before we present the schematic procedure used to construct the threat diagram in Figure 7 from the target model in Figure 6, we provide a number of definitions needed for this purpose.

A dependency gate is either the provider gate !s or the consumer gate ?s of a service s. A dependency is a pair of dependency gates. This means that a dependency is either of the form (!s, ?s) for some service s, or of the form (?s, !t) where s and t are different services. A dependency path is a totally ordered finite set of dependencies

\( \{(g_1, h_1), (g_2, h_2), \ldots, (g_n, h_n)\} \)

such that for all \( 0 < j < n, h_j = g_{j+1} \). The gate \( g' \) is dependent on the gate \( g \) if there is a dependency path

\( \{(g_1, h_1), (g_2, h_2), \ldots, (g_n, h_n)\} \)

such that \( g = g_1 \) and \( g' = h_n \). We then write \( g \sim g' \).

In the following we illustrate the relations between dependency constructs, dependency gates, dependencies, and dependency paths. In Figure 8 is an excerpt of the target model in Figure 6, where dependency gates have been high-lighted. We use the short-hand notation \( s_X \) to refer to “Service X” in the following. The excerpt has the following dependency gates, dependencies, and dependency paths:

- **Dependency gates:** !s_{16}, ?s_{16}, !s_{17}, ?s_{17}, !s_{18}, and ?s_{18}.
- **Dependencies:** (!s_{16}, ?s_{16}), (!s_{17}, ?s_{17}), (!s_{18}, ?s_{18}), (?s_{16}, !s_{17}), and (?s_{18}, !s_{17}).
- **Dependency paths**
We then write \( g \) or equivalence class with respect to \( \sim \). We use the short-hand notation presented above when constructing the high-level threat diagram in Figure 7 from the target and \( g \) level threat diagram from the target model as follows:

- of length one: \( \{(s_{18}, s_{16})\}, \{(s_{17}, s_{17})\}, \{(s_{18}, s_{18})\}, \{(s_{16}, s_{17})\}, \) and \( \{(s_{18}, s_{17})\} \).
- of length two: \( \{(s_{16}, s_{16}), (s_{16}, s_{17})\}, \{(s_{16}, s_{17}), (s_{17}, s_{17})\}, \{(s_{18}, s_{18}), (s_{18}, s_{17})\}, \) and \( \{(s_{18}, s_{17}), (s_{17}, s_{17})\} \).
- of length three: \( \{(s_{16}, s_{16}), (s_{16}, s_{17}), (s_{17}, s_{17})\}, \) and \( \{(s_{18}, s_{18}), (s_{18}, s_{17}), (s_{17}, s_{17})\} \).

If we had replaced the “or” operator in the dependency construct in Figure 8 with an “and” operator, then we would have ended up with the same dependencies and dependencies paths. We do not distinguish between “and” and “or” operators when identifying dependencies and dependency paths. These operators are only of importance when capturing the impact of dependencies on risk to quality of provided services.

Two gates \( g_1 \) and \( g_2 \) are mutually dependent iff

\[
g_1 \sim g_2 \land g_2 \sim g_1
\]

or

\[
g_1 = g_2
\]

We then write \( g_1 \sim \sim \sim g_2 \). Moreover, we write \( g_1 \sim \sim \sim \sim \sim \sim g_2 \) to state that \( g_1 \sim \sim \sim g_2 \) and \( g_1 \sim g \) and \( g_2 \sim g \). Since \( \sim \sim \sim \) is a reflexive, symmetric, and transitive relation of the set of gates it follows that \( \sim \sim \sim \) is an equivalence relation. The same holds for \( \sim \sim \sim \sim \sim \sim \). For any gate \( g \), let \( [g] \) be its equivalence class with respect to \( \sim \sim \sim \sim \sim \sim \). Moreover, we use \( [g]_{\sim \sim \sim \sim \sim \sim} \) to denote its restriction to \( \sim \sim \sim \sim \sim \sim \).

For each service \( s \) provided by a target system to an environment system, construct a high-level threat diagram from the target model as follows:

1. Introduce the quality assets identified in Step 2.1 for the provided service \( s \).
2. For each of these quality assets, introduce a high-level unwanted incident and connect this to the asset by an impacts relation.
3. Let \( G_T \) be the set of all provider gates \( !s' \) within the target such that \( !s' \sim ?s \).
4. Introduce a high-level threat scenario for each equivalence class \( [g]_{\sim \sim \sim \sim \sim \sim} \) where \( g \in G_T \).
5. Only one of these equivalence classes contains \( !s \). Connect its high-level threat scenarios to the high-level unwanted incidents introduced under 2 using leads-to relations.
6. For each pair of different equivalence classes \( [g_1]_{\sim \sim \sim \sim \sim \sim} \) and \( [g_2]_{\sim \sim \sim \sim \sim \sim} \) connect their high-level threat scenarios with a leads-to relation decorated by a vulnerability if there is a dependency path \( \{(g_1, g), (g, g_2)\} \).
7. Let \( G_E \) be the set of all provider gates \( !s' \) within the environment such that \( \{(!s', g_1), (g_1, g_2)\} \), where \( g_2 \in G_T \).
8. Introduce a high-level threat scenario for each \( !s' \in G_E \), and connect the scenario to the high-level threat scenario representing the equivalence class \( [g_2]_{\sim \sim \sim \sim \sim \sim} \) using a leads-to relation decorated by a vulnerability.

In the following we present the results of executing the different steps of the procedure presented above when constructing the high-level threat diagram in Figure 7 from the target model in Figure 6. We use the short-hand notation \( s_X \) to refer to “Service X” in Figure 6.
1. The quality asset “Availability of Service 5 delivered to System 5” is introduced.

2. The unwanted incident “Incident with impact on the availability of Service 5 delivered to System 5” is introduced, and connected to the quality asset by an impacts relation.

3. The set $G_T = \{s_{1}, \ldots , s_{13}, s_{15}, \ldots , s_{18}\}$ is identified.

4. The following equivalence classes and their respective high-level threat scenarios are identified and introduced, respectively:

   - $[s_{1}]_s_{5} = [s_{2}]_s_{5} = [s_{3}]_s_{5} = [s_{4}]_s_{5} = \{s_{1}, s_{2}, s_{3}, s_{4}\}$: “Service 1, 2, 3, and 4 are not delivered according to requirements”
   - $[s_{5}]_s_{5} = \{s_{5}\}$: “Service 5 is not delivered according to requirements”
   - $[s_{6}]_s_{5} = \{s_{6}\}$: “Service 6 is not delivered according to requirements”
   - $[s_{7}]_s_{5} = [s_{8}]_s_{5} = [s_{9}]_s_{5} = \{s_{7}, s_{8}, s_{9}\}$: “Service 7, 8, and 9 are not delivered according to requirements”
   - $[s_{10}]_s_{5} = [s_{11}]_s_{5} = [s_{12}]_s_{5} = [s_{13}]_s_{5} = \{s_{10}, s_{11}, s_{12}, s_{13}\}$: “Service 10, 11, 12, and 13 are not delivered according to requirements”
   - $[s_{14}]_s_{5} = \{s_{14}\}$: “Service 15 is not delivered according to requirements”
   - $[s_{16}]_s_{5} = \{s_{16}\}$: “Service 16 is not delivered according to requirements”
   - $[s_{17}]_s_{5} = [s_{18}]_s_{5} = \{s_{17}, s_{18}\}$: “Service 17 and 18 are not delivered according to requirements”

5. The high-level threat scenario “Service 5 is not delivered according to requirements” is connected by a leads-to relation to the unwanted incident.

6. The high-level threat scenarios of the following pairs of equivalence classes are connected by leads-to relations decorated by vulnerabilities:

   - $[s_{17}]_s_{5}$ and $[s_{15}]_s_{5}$ as a result of $\{(s_{17}, s_{17}), (s_{15}, s_{15})\}$
   - $[s_{15}]_s_{5}$ and $[s_{12}]_s_{5}$ as a result of $\{(s_{15}, s_{15}), (s_{12}, s_{12})\}$
   - $[s_{10}]_s_{5}$ and $[s_{9}]_s_{5}$ as a result of $\{(s_{10}, s_{10}), (s_{9}, s_{9})\}$
   - $[s_{8}]_s_{5}$ and $[s_{6}]_s_{5}$ as a result of $\{(s_{8}, s_{8}), (s_{6}, s_{6})\}$
   - $[s_{6}]_s_{5}$ and $[s_{3}]_s_{5}$ as a result of $\{(s_{6}, s_{6}), (s_{3}, s_{3})\}$
   - $[s_{2}]_s_{5}$ and $[s_{5}]_s_{5}$ as a result of $\{(s_{2}, s_{2}), (s_{5}, s_{5})\}$

7. The set $G_E = \{s_{14}\}$ is identified.

8. The high-level threat scenario “Service 14 is not delivered according to requirements” is introduced and connected to the high-level threat scenario “Service 16 is not delivered according to requirements” by a leads-to relation decorated by a vulnerability as a result of $\{(s_{14}, s_{14}), (s_{16}, s_{16})\}$.

**Input documentation:**

- The target model from Step 1.3.
- The identified quality assets from Step 2.1.
Output documentation: One high-level threat diagram outlining the impact of service dependencies on the quality assets for each provided service.

3.2.3 Step 2.3: Construct detailed threat diagrams of the impact of service dependencies on identified quality assets

Objective: Achieve a detailed understanding of the impact of service dependencies on the identified quality assets.

Rationale: The threat diagrams from Step 2.2 provide only a high-level outline of the impact of service dependencies on the identified quality assets. To establish a risk picture that can be monitored, we need to detail those diagrams.

How conducted: In Figure 9 is a threat diagram (where some of the details have been suppressed) that shows part of the result of detailing the high-level threat diagram in Figure 7.

We detail the high-level constructs, one by one, by following the instructions given in [7]. We only deviate from these instructions when detailing leads-to relations. A leads-to relation between two high-level constructs is detailed by decomposing it. If vulnerabilities are assigned to the leads-to relation being detailed, then the detailing also involves the decomposition of those vulnerabilities. It should be noticed that if the vulnerability represents the dependency of target services on an environment service, then the vulnerability is decomposed into as many vulnerabilities as there are required service levels associated with the environment service. For example, the vulnerability “Service 16 depends on Service 14” in Figure 7 has been decomposed into the two vulnerabilities “Service 16 depends on availability of Service 14” and “Service 16 depends on integrity of Service 14”; one for each of the required service levels associated with “Service 14.”

As a result of the decomposition of the high-level vulnerabilities, the referring threat scenarios, and the referring unwanted incident in Figure 7, the high-level in-gates and out-gates and the impacts relation in Figure 7 have been decomposed, and likelihood values and consequences values have been assigned to the gates and impacts relations, respectively. For each out-gate being the source of a leads-to relation associated with a vulnerability representing the dependence of target services on a particular area of service scope of an environment service, we estimate the likelihood of the required service level not being delivered. This is done by first calculating the worst-case service level of the particular area of service scope. The worst-case service level specifies our minimum expectation to the particular area of service scope. It is calculated based on the required service level and the trust level calculated in Step 1.3. The likelihood is then estimated based on the difference between the required service level and the worst case service level.

As part of this step, we also specify scales for measuring likelihood and consequence, and functions for calculating risk values. The risk functions are used after we have created the detailed threat diagrams to determine the risk values of the different risks to quality of provided services. A risk value is determined based on the likelihood of an unwanted incident and its consequence with respect to a quality asset.

Input documentation:

- The high-level threat diagrams from Step 2.2.
- Target model from Step 1.3.
Figure 9: Threat diagram that shows part of the result of detailing the threat diagram in Figure 7
Output documentation:

- Detailed threat diagrams documenting the impact of service dependencies on the quality assets.
- Worst-case service levels.
- Scales for measuring likelihood and consequence.
- Risk functions for calculating risk values.
- A list of risks to quality of provided services.

3.3 Step 3: Identify indicators for interconnected systems

3.3.1 Step 3.1: Identify risks to be monitored

Objective: Identify the risks to quality of provided services that should be monitored.

Rationale: A risk analysis will often result in a number of identified risks to quality of provided services. We need to identify the risks that should be monitored, since it is often not in the trustor’s interest to monitor all the risks. Moreover, there may be risks for which monitoring is not feasible.

How conducted: For each risk resulting from Step 2.3, we must decide whether it should be monitored. Typically, a risk to quality of provided services is selected for monitoring if it is believed that the likelihood and/or consequence value determining its risk value is likely to change in a manner that will considerably harm the trustor. A risk may also be selected for monitoring if we are uncertain about the risk value.

Input documentation:

- The detailed threat diagrams from Step 2.3.
- The list of risks to quality of provided services from Step 2.3.

Output documentation: A list of risks to quality of provided services to be monitored.

3.3.2 Step 3.2: Identify relevant indicators for the risks to be monitored

Objective: Identify relevant indicators for monitoring the risk values of the risks to be monitored.

Rationale: To monitor changes in risk values we need to identify indicators. The indicators are calculated from measurable properties of the interconnected systems.
How conducted: For the risks identified to be monitored in Step 3.1, we identify relevant indicators. Indicators for monitoring consequence are related to impacts relations between unwanted incidents and quality assets. On the other hand, indicators for monitoring likelihood may not only be related to unwanted incidents, but also to vulnerabilities and threat scenarios leading up to an incident, since the likelihoods of vulnerabilities being exploited and threat scenarios occurring will affect the likelihood of the unwanted incident occurring.

Basic indicators are identified for the different likelihood and consequence values to be monitored. If more than one basic indicator is needed for monitoring a consequence or likelihood value, then a composite indicator, aggregating the basic indicators, is also identified.

Input documentation:
- The list of risks to quality of provided services to be monitored from Step 3.1.
- The detailed threat diagrams from Step 2.3.

Output documentation: A set of relevant basic and composite indicators for monitoring likelihood and consequence.

3.4 Step 4: Specify design and deployment of identified indicators for interconnected systems

3.4.1 Step 4.1: Specify design of indicators for risk monitoring

Objective: Specify how basic and composite indicators for monitoring likelihood and consequence values should be designed.

Rationale: We need to specify how the identified basic and composite indicators from Step 3.2 should be designed, i.e., how they should be calculated, in order to be useful for monitoring.

How conducted: A design specification, in the form of an algorithm, is provided for each indicator identified in Step 3.2. It specifies the data needed for calculating the indicator, how the indicator should be calculated, and the output from the calculation. Assuming the likelihood and consequence intervals obtained in Step 2.3 are correct, the algorithm should yield likelihoods and consequences in these intervals when applied to the basic indicator values at the time these intervals were determined.

Input documentation:
- The list of risks to quality of provided services to be monitored from Step 3.1.
- The relevant indicators identified in Step 3.2.
- The detailed threat diagrams from Step 2.3.
- Basic indicator values from the time when the detailed threat diagrams were constructed.

Output documentation: A design specification for each indicator identified in Step 3.2.
3.4.2 Step 4.2: Specify deployment of indicators for risk monitoring

Objective: Specify how basic and composite indicators for monitoring likelihood and consequence values should be deployed in the interconnected systems.

Rationale: We need to specify how the identified basic and composite indicators from Step 3.2 should be deployed in the interconnected systems, i.e., how the data needed to calculate the different indicators should be extracted and transmitted within the interconnected systems, in order to be useful for monitoring.

How conducted: A deployment specification is provided for each indicator identified in Step 3.2. It specifies how the data needed to calculate the indicator should be extracted and transmitted within the interconnected systems.

Input documentation: The design specifications from Step 4.1.

Output documentation: A deployment specification for each indicator.

4 Demonstration of Step 1: Document interconnected systems

We consider an SoS consisting of an electrical power production infrastructure (EPP), a public telecom infrastructure (PTI), and an electrical power grid (EPG). In the following we assume that we as analysts have been hired by the company in charge of the electrical power production infrastructure, Client EPP, to help capture and monitor the impact of service dependencies on the quality of the services that Client EPP provides to the parties in charge of the public telecom infrastructure and the electrical power grid.

4.1 Step 1.1: Model interconnected systems

Figure 10 documents the electrical power production infrastructure and its environment. The different systems provide and/or require electricity (elec), control instructions (cinstr), and sensor data (sdata). All the services with the exception of the electricity services are data services. For each electricity service, we provide a required service level for availability. Each required service level is a conjunction of availability with respect to time and availability with respect to the amount of electricity (in megawatt hours (MWh)) that needs to be delivered. Both these availability requirements are for the period of one year. The required service levels for electricity services take into account that service disruptions may occur. For instance, consider the electricity service provided by “Distribution line 3” to “Private telecom system.” The “Private telecom system” will not experience any disruptions of the service if the availability with respect to time is 100% (available 8760 hours per year) and if the availability with respect to electricity delivered is 22 MWh. The latter is an estimate for the amount of electricity that “Private telecom system” needs during the period of one year.

For the data services, the required service levels (also for the period of one year) are specified in terms of percentages of all sensor data/control instructions messages that are sent. We can for instance specify the percentages of all sent data messages that need to be delivered (availability), be delivered with integrity, and comply with the data confidentiality policy of Client EPP. In Section 5.2 we explain what it means to comply with the data confidentiality policy. An integrity requirement cannot be higher than the availability requirement for the same service, since each integrity requirement specifies the percentage of all sent data messages that needs
Figure 10: Target model for the electrical power production infrastructure and its environment
to be delivered with integrity. Thus, the integrity requirement is equal to or less than the availability requirement for each data service in Figure 10. For Client EPP, the integrity of a data message is only important if it is delivered. On the other hand, confidentiality is important for both data messages that are delivered and data messages that are lost during transmission. A data message can for instance be intercepted by an outsider before it is lost during transmission. Thus, the confidentiality requirement may be higher than the availability requirement for the same service.

In the electrical power production infrastructure there is a “Large hydro power plant.” The electrical power produced by this plant is transmitted on a high-voltage “Transmission line” to a “Power substation.” Here, the power is transformed to low-voltage power by a transformer, before being distributed to its end-users by distribution lines. “Distribution line 1” provides electrical power to the “Public telecom system.” The infrastructure also consists of a “Small hydro power plant.” This power plant distributes power directly to its end-users by the use of “Distribution line 2” and “Distribution line 3.” “Private telecom system” and “Control system,” both located within the electrical power production infrastructure, are two of the end-users that receive electrical power from these two distribution lines. These two systems share a “Backup power system,” which is used when the electrical power grid fails to provide electricity to one or both systems.

The “Control system” is used to operate the “Large hydro power plant.” By the use of the “Private telecom system” it sends control instructions to the plant, while sensors at the plant send data to the “Control system” through the same telecom system. The “Control system” responds to errors arising at the plant. If it cannot resolve the errors, it will shut down the plant to protect equipment. If the connection to the “Control system” is lost, the plant will automatically shut down if it cannot resolve errors by itself. The required service level with respect to availability is 99% for all the data services exchanged between the “Control system” and the “Large hydro power plant,” since the plant has some ability of operating independently of the “Control system.” Moreover, the required service level with respect to integrity is 99% for all the data services.

Due to its size, the “Small hydro power plant” is operated by a system operator from his “Home office computer.” The operator uses a computer that is dedicated to this task. He sends encrypted control instructions to the plant through the “Public telecom system,” while the sensors at the plant sends encrypted data to the operator through the same telecom system. The encrypted communication is achieved through the use of symmetric-key cryptography. The system operator responds to errors arising at the plant. If he cannot resolve the errors, he will shut down the plant to protect equipment. If the connection to the “Public telecom system” is lost, the plant will automatically shut down to protect equipment. This is done as a precautionary step, since the plant is not able to resolve errors by itself. Since the availability of the data services exchanged between the “Small hydro power plant” and the “Home office computer” are crucial for the operation of the “Small hydro power plant,” the required service level for all the data services with respect to availability is 99.99%. It should be noticed that the integrity and confidentiality requirements for data services provided by “Public telecom system” to “Home office computer” and “Small hydro power plant” do not specify explicit requirements that “Public telecom system” needs to fulfill when providing the data services. It is more correct to say that these requirements are to the data messages themselves. Client EPP requires that data messages’ compliance with the data confidentiality policy and data messages’ integrity should not be changed while at “Public telecom system” or during transmission to its destinations. Notice that only the confidentiality requirements have been set to 100% in Figure 10. The integrity requirements would have been set to 100% too, but this is not possible since the availability requirements equal 99.99% in both cases.
4.2 Step 1.2: Capture service dependencies

In Figure 11, the target model in Figure 10 is annotated with the service dependencies. Most of the service dependencies are self-explanatory, but note especially that “Small hydro power plant” depends on the availability of control instructions, provided by “Home office computer,” to produce electricity. The “Large hydro power plant” is less dependent on control instructions than the “Small hydro power plant,” but since it depends on control instructions in situations where it cannot resolve errors, there is a dependency between the required control instructions service and the electricity service provided to “Transmission line.” It should also be noticed that both “Private telecom system” and “Control system” can require electricity from the “Backup power system” if the electrical power grid fails to provide electricity, and that incoming sensor data messages may affect the outgoing control instructions messages, and vice versa. The dependencies between incoming and outgoing messages are a result of control instructions messages often being created based on the incoming sensor data messages, and that control instructions messages affect the operation of “Small hydro power plant” and its data sensors, which again affect the outgoing sensor data messages.

4.3 Step 1.3: Capture trust relations

In Figure 12, the target model in Figure 11 is annotated with trust relations. As can be seen in the figure, trust levels have been assigned to the required service levels for those services that are provided by systems of the environment to systems of the target.

All the services for which trust levels should be assigned are considered very reliable by Client EPP. Thus, it is expected that they should achieve their required service levels. Even so, Client EPP is aware that the services can fail. After having considered both the high reliability of the services and the possibility of service failures, Client EPP assigns high trust levels to the different required service levels.

For the control instructions service provided by “Public telecom system” to “Small hydro power plant,” Client EPP has a trust of:

- 0.97 in that the control service is delivered according to the confidentiality requirement;
- 0.95 in that the control service is delivered according to the integrity requirement; and
- 0.99 in that the control service is delivered according to the availability requirement.

5 Demonstration of Step 2: Analyze the impact of service dependencies on risk to quality of provided services

5.1 Step 2.1: Identify quality assets

For the sake of simplicity, we demonstrate the method by only identifying quality assets for one of the provided services. In Appendices A–C we demonstrate the method on the other provided services.

A concern of Client EPP is that services dependencies in the SoS may affect the ability of “Small hydro power plant” to provide the sensor data service according to the quality requirements associated with the service. If this service is affected, then the ability of “Home office computer” to control and operate the “Small hydro power plant” may be affected as well, which again may impact the electricity services provided to “Distribution line 2” and “Distribution line 3.” Client EPP therefore seeks to protect the quality assets “Confidentiality of sensor data delivered to Public telecom system,” “Integrity of sensor data delivered to Public telecom system,”
Figure 11: Target model annotated with service dependencies
Figure 12: Target model annotated with trust relations
Sensor data service is not delivered according to requirements

Control instructions service is not delivered according to requirements

Incident with impact on confidentiality of the sensor data service delivered to Public telecom system

Incident with impact on integrity of the sensor data service delivered to Public telecom system

Incident with impact on availability of the sensor data service delivered to Public telecom system

Confidentiality of sensor data delivered to Public telecom system

Integrity of sensor data delivered to Public telecom system

Availability of sensor data delivered to Public telecom system

Figure 13: Threat diagram, constructed schematically from the target model in Figure 12, which provides a high-level outline of the impact of service dependencies on the quality of the sensor data service provided by “Small hydro power plant” to “Public telecom system” and “Availability of sensor data delivered to Public telecom system,” and wants to identify the service dependencies’ impact on these quality assets.

5.2 Step 2.2: Construct high-level threat diagrams of the impact of service dependencies on identified quality assets

For the sensor data service provided to “Public telecom system,” the high-level threat diagram in Figure 13 has been constructed schematically from the target model in Figure 12. The threat diagram provides a high-level description of the impact of service dependencies on the quality of the sensor data service provided to “Public telecom system.” In the threat diagram we use the abbreviations “sensor data service” and “control instructions service” to refer to the sensor data service provided by “Small hydro power plant” to “Public telecom system” and the control instructions service provided by “Public telecom system” to “Small hydro power plant,” respectively.

5.3 Step 2.3: Construct detailed threat diagrams of the impact of service dependencies on identified quality assets

Before we perform the detailed risk analysis of how target systems may fail to provide services according to requirements, we need to establish how to measure likelihood and consequence, as well as defining the risk function. Table 1 shows how likelihood is measured, while Table 2 shows how consequence is measured for the different quality assets.
Table 1: Likelihood scale

<table>
<thead>
<tr>
<th>Likelihood</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certain</td>
<td>Fifty times or more per year ([500, \infty)) : 10 years</td>
</tr>
<tr>
<td>Very likely</td>
<td>Ten to fifty times per year ([100, 499]) : 10 years</td>
</tr>
<tr>
<td>Likely</td>
<td>Five times to ten times per year ([50, 99]) : 10 years</td>
</tr>
<tr>
<td>Possible</td>
<td>Two to five times per year ([20, 49]) : 10 years</td>
</tr>
<tr>
<td>Unlikely</td>
<td>Once a year ([6, 19]) : 10 years</td>
</tr>
<tr>
<td>Very unlikely</td>
<td>Less than once per year ([2, 5]) : 10 years</td>
</tr>
<tr>
<td>Rare</td>
<td>Less than once per ten years ([0, 1]) : 10 years</td>
</tr>
</tbody>
</table>

There is need to clarify what we mean with “lack of integrity” and “do not comply with the data confidentiality policy.” ISO/IEC 27000 [10] defines confidentiality as the “property that information is not made available or disclosed to unauthorized individuals, entities, or processes,” while it defines integrity as the “property of protecting the accuracy and completeness of assets.” In our case, asset refers to information. In the case of confidentiality, it may be extremely difficult to detect whether information contained in a sensor data message or control instructions message have been made available or been disclosed to unauthorized individuals, entities, or processes. Instead of focusing on whether the information has been disclosed, we focus on how the information is protected against disclosure. If a sensor data message or control instructions message comes with strong protection against disclosure of the information contained in the message, then it likely that the information will remain confidential during transmission. Client EPP has a data confidentiality policy that defines what it means for information in a sensor data message or in a control instructions message to be well-enough protected against disclosure. At Client EPP, all sent messages should comply with this policy. The information is, for instance, not well-enough protected if: the message is sent in clear text; the cryptographic algorithm used has flaws which makes it vulnerable to attacks; the cryptographic key used has been disclosed, has a long life-span, or has been incorrectly generated; etc.

In the case of “lack of integrity,” we say that a sensor data message or a control instructions message has lack of integrity if: the information contained in the message has been changed deliberately or by accident during transmission, processing, or storage of the message; the message has not been created and sent by one of Client EPP’s systems; or the message has been created based on data that is not correct with respect to the true state of the object represented by the data. With respect to the latter, a sensor data message may be created based on incorrect sensor data, while control instructions may be created based on sensor data that is not correct with respect to the true state of a power plant.

To calculate the number of sensor data messages that are not delivered, delivered with lack of integrity, or that do not comply with the data confidentiality policy, it is helpful to have an estimate of the number of sensor data messages sent from “Small hydro power plant” in the period of one year. Client EPP estimates this number to be 5000.
Table 2: How consequence is measured for the three quality assets

<table>
<thead>
<tr>
<th>Availability of sensor data delivered to Public telecom system</th>
<th>Number of sensor data messages that are not delivered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality of sensor data delivered to Public telecom system</td>
<td>Number of sensor data messages sent that do not comply with the data confidentiality policy</td>
</tr>
<tr>
<td>Integrity of sensor data delivered to Public telecom system</td>
<td>Number of sensor data messages that are delivered with lack of integrity</td>
</tr>
</tbody>
</table>

For all the risks, the risk is classified as acceptable or unacceptable as follows:

\[
Expected \text{ service level} = \frac{\text{Maximum service level} - (\text{Likelihood} \cdot \text{Consequence})}{\text{Maximum service level}}
\]  

(1)

\[
\begin{align*}
\text{if } Expected \text{ service level} & \geq \frac{\text{Required service level}}{\text{Maximum service level}} \text{ then} \\
\text{Risk value} & = \text{Acceptable} \\
\text{else} \\
\text{Risk value} & = \text{Unacceptable}
\end{align*}
\]

(2)

Here, the Maximum service level is the highest achievable service level for the area of service scope associated with the quality asset in question. For example, the highest achievable service level for the integrity of the sensor data service is 5000. This means that all the 5000 sensor data messages sent during the period of one year are delivered with integrity. A risk associated with a quality asset is Unacceptable if the Expected service level is less than Required service level Maximum service level.

In Figure 14 is the detailed version of the high-level threat diagram in Figure 13. The referring elements in Figure 14 refer to the referenced threat scenarios provided in Figures 15 and 16, and the referenced unwanted incidents provided in Figure 20. Moreover, the referenced threat scenario in Figure 16 contains three referring threat scenarios, which refer to the referenced threat scenarios provided in Figures 17–19. Client EPP has estimated all the likelihood and consequence values in the different figures.

We refer to \(i_x\) and \(o_y\) of the referring threat scenarios and unwanted incidents as in-gate and out-gate, respectively. Relations to an element inside a referenced threat scenario must go through an in-gate, while relations to an element outside the referenced threat scenario must go through an out-gate. The likelihood value of an in-gate \(i_x\) documents the contribution of an element outside the referenced threat scenario via gate \(i_x\) to the likelihood of an element inside the referenced threat scenario, while the likelihood of the out-gate \(o_y\) documents the contribution of the likelihood of an element inside the referenced threat scenario via gate \(o_y\) to the likelihood of an element outside the referenced threat scenario.

Below we provide some examples of the semantics of elements and relations in the different figures. For more information on the semantics of the CORAS language, see [7].

- **Threat scenario**: Threat scenario “Control instructions message is not delivered” occurs with likelihood “Very likely” (Figure 17).

- **Leads-to relation (with conditional likelihood)**: “Invalid control instructions are used by the Small hydro power plant” leads to “Small hydro power plant starts to operate in an incorrect state” with conditional likelihood “0.1” (Figure 18).
Figure 14: Detailed version of the high-level threat diagram in Figure 13
• **Leads-to relation (with vulnerability):** “Control instructions message is delivered, where its integrity has been changed during transmission or while at Public telecom system” leads to “Re-transmission of control instructions message is not requested” with conditional likelihood “0.001,” due to vulnerability “Possible that checksum algorithm fails to detect integrity violations” (Figure 18).

• **In-gate (with likelihood):** $i_1$ is an in-gate with likelihood “Very likely” (Figure 14).

• **Out-gate (with likelihood):** $o_1$ is an out-gate with likelihood “Very likely” (Figure 14).

• **Leads-to relations (between elements of referenced threat scenarios):** “Control instructions service is not delivered by Public telecom system according to the availability requirement that Public telecom system is required to fulfill” leads to “Control instructions message is not delivered” via gates $o_{11}, i_{11},$ and $i_{10}$, due to vulnerability “Small hydro power plant depends on availability of control instructions” (Figures 14–17).

• **Unwanted incident:** Unwanted incident “Sensor data is sent in plain text from Small hydro power plant to an outsider” occurs with likelihood “Unlikely” (Figure 20).

• **Impacts relation (between element of referenced unwanted incident and asset):** “Sensor data is sent in plain text from Small hydro power plant to an outsider” impacts “Confidentiality of sensor data delivered to Public telecom system” via gate $o_{10}$ with consequence “96” (Figures 14 and 20).

As can be seen in Figure 14, the vulnerability “Sensor data service depends on control instructions service” in Figure 13 has been decomposed into three vulnerabilities. The referenced threat scenario in Figure 15 is a detailing of the referring threat scenario “Control instructions service is not delivered according to requirements” in Figure 13. Since “Public telecom system” is only required to deliver the control instructions service according to the availability requirement, the referenced threat scenario distinguish between the failure of not achieving the availability requirement, and the failures of not achieving the confidentiality and integrity requirements.

Client EPP estimates that 1000 control instructions messages are sent each year to “Small hydro power plant.” Before we can estimate the likelihoods of the control instructions service not being delivered according to the confidentiality, integrity, and availability requirements, we need to calculate the worst-case service levels (required service level × trust level) of the control instructions service delivered by “Public telecom system.” These are as follows:

- $100\% \cdot 0.97 = 97\%$ of the sent control instructions messages do comply with the data confidentiality policy;
- $99.99\% \cdot 0.95 = 94.99\%$ of the sent control instructions messages are delivered with integrity; and
- $99.99\% \cdot 0.99 = 98.99\%$ of the sent control instructions messages are delivered.

To estimate the likelihoods we use the estimated number of control instructions messages sent each year in combination with the required and worst-case service levels of the control instructions service delivered by “Public telecom system.” The required service levels specify that:

- $1000 \cdot 100\% = 1000$ of the sent control instructions messages should comply with the data confidentiality policy;
Control instructions service is not delivered according to requirements

Figure 15: The referenced threat scenario “Control instructions service is not delivered according to requirements,” referred to in Figure 14

- $1000 \cdot 99.99\% = 999.9$ of the sent control instructions messages should be delivered with integrity; and
- $1000 \cdot 99.99\% = 999.9$ of the sent control instructions messages should be delivered.

On the other hand, our expectations according to the worst-case service levels are that:

- $1000 \cdot 97\% = 970$ out of the required 1000 control instructions messages comply with the data confidentiality policy;
- $1000 \cdot 94.99\% = 949.9$ out of the required 999.9 control instructions messages are delivered with integrity; and
- $1000 \cdot 98.99\% = 989.9$ out of the required 999.9 control instructions messages are delivered.

Based on the calculations for required and worst-case service levels, we end up with the following likelihoods:

- The likelihood of the control instructions service not being delivered according to the confidentiality requirement is “Very likely” ($1000 – 970 = 30$ control instructions messages in the period of a year).
- The likelihood of the control instructions service not being delivered according to the integrity requirement is “Certain” ($999.9 – 949.9 = 50$ control instructions messages in the period of a year).
- The likelihood of the control instructions service not being delivered according to the availability requirement is “Very likely” ($999.9 – 989.9 = 10$ control instructions messages in the period of a year).

The referenced threat scenario “Sensor data service is not delivered according to requirements” is given in Figure 16. The internal threat behavior of “Small hydro power plant” is
Figure 16: The referenced threat scenario “Sensor data service is not delivered according to requirements,” referred to in Figure 14

described by the referenced threat scenarios in Figures 17–19. The different referenced threat scenarios describe how “Small hydro power plant” may fail to deliver the sensor data service according to requirements as a result of “Public telecom system” failing to deliver the control instructions service according to its requirements.

Figure 20 contains the referenced unwanted incidents referred to in Figure 14. For each of the unwanted incidents, Client EPP believes that more than one sensor data message is affected by the incident. For the incident “No sensor data messages are sent due to Small hydro power plant being unavailable due to lack of control instructions or use of invalid control instructions,” Client EPP estimates a down time of one day, while a down time of 3 days is estimated for the incident “No sensor data messages are sent due to Small hydro power plant being unavailable due to malicious software.” For the incident “Incorrect sensor data is sent to Public telecom system due to invalid control instructions being used by Small hydro power plant,” Client EPP estimates that “Small hydro power plant” sends incorrect sensor data messages for a period of 12 hours as a result of using incorrect control instructions. For the incident “Sensor data is sent in plain text from Small hydro power plant to an outsider,” Client EPP believes that this can go on undetected for at much as seven days. The same is believed for the incident “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus.” With an average number of 13.7 (\(2^5 \times 0.001\)) sensor data messages being sent each day, we get the consequence values documented in Figure 14.

The result of the detailed analysis is five risks, where each risk consists of an unwanted incident, its likelihood of occurring, and the consequence of the unwanted incident with respect to a quality asset. Based on the risk function, defined in Equations (1) and (2), the estimated number of sensor data messages sent each year (5000), and the required service levels for the sensor data service, we can calculate the risk values of the five risks.
Control instructions message is not delivered to Small hydro power plant

**Figure 17:** The referenced threat scenario “Control instructions message is not delivered to Small hydro power plant,” referred to in Figure 16

Control instructions message with lack of integrity is delivered to Small hydro power plant

**Figure 18:** The referenced threat scenario “Control instructions message with lack of integrity is delivered to Small hydro power plant,” referred to in Figure 16
Figure 19: The referenced threat scenario “Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant,” referred to in Figure 16
Incident with impact on availability of the sensor data service delivered to Public telecom system

No sensor data messages are sent due to Small hydro plant being unavailable due to lack of control instructions or use of invalid control instructions
[Unlikely]

Incident with impact on confidentiality of the sensor data service delivered to Public telecom system

Sensor data is sent in plain text from Small hydro power plant to an outsider
[Unlikely]

Incident with impact on integrity of the sensor data service delivered to Public telecom system

Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus
[Unlikely]

Incorrect sensor data is sent to Public telecom system due to invalid control instructions being used by Small hydro power plant
[Rare]

Figure 20: The referenced unwanted incidents “Incident with impact on confidentiality of the sensor data service delivered to Public telecom system,” “Incident with impact on integrity of the sensor data service delivered to Public telecom system,” and “Incident with impact on availability of the sensor data service delivered to Public telecom system,” referred to in Figure 14
Interval arithmetic needs to be used during the calculation of risk values, since likelihoods in the form of intervals are used in the calculations. For two intervals \([a, b]\) and \([c, d]\), where both are subsets of the positive real line \(\mathbb{R}^+\), the basic operations of interval arithmetic are:

- **Addition**: \([a, b] + [c, d] = [a + c, b + d]\)
- **Subtraction**: \([a, b] - [c, d] = [\max(0, a - d), \max(0, b - c)]\)
- **Multiplication**: \([a, b] \cdot [c, d] = [a \cdot c, b \cdot d]\)
- **Division**: \([a, b] \div [c, d] = [a \div d, b \div c]\) when 0 is not in \([c, d]\)

In addition, a positive real number \(e\) may be written as the interval \([e, e]\). Notice that the application of all the basic operations result in intervals that are subsets of the positive real line. For instance, \([a, b] - [c, d]\) results in the interval \([0, 0]\) if \(d > a\) and \(c > b\). In our case, it does not make any sense to produce intervals that contains negative values.

A risk value is acceptable if \(\text{Expected service level} \geq \text{Required service level}\), while it is unacceptable in the opposite case. We need some additional interval arithmetic rules to determine whether the risk value is acceptable or not. We let \([a, b]\) and \([c, d]\) represent \(\text{Expected service level}\) and \(\text{Required service level}\) respectively. Both intervals are subsets of the positive real line \(\mathbb{R}^+\). The rules are as follows:

- Risk value is **Acceptable**: \([a, b] \geq [c, d]\) if \(a \geq c\)
- Risk value is **Unacceptable**: \([a, b] < [c, d]\) if \(a < c\)

In the following we calculate the risk values for the five risks. In all the equations for \(\text{Expected service level}\), \(\text{Likelihood}\) is given for the period of one year, since both \(\text{Required service level}\) and \(\text{Maximum service level}\) are given for the period of one year.

The risk value of “Sensor data is sent in plain text from Small hydro power plant to an outsider” is Unacceptable since \(\text{Expected service level}\) is less than \(\text{Required service level}\). In this case, the calculations are as follows:

\[
\text{Expected service level} = \frac{\text{Maximum service level} - (\text{Likelihood} \cdot \text{Consequence})}{\text{Maximum service level}}
\]

\[
= \frac{5000 - ([0.6, 1.9] \cdot [96, 96])}{5000}
\]

\[
= \frac{[5000, 5000] - ([0.6, 1.9] \cdot [96, 96])}{[5000, 5000]}
\]

\[
= \frac{[5000, 5000] - [57.6, 182.4]}{[5000, 5000]}
\]

\[
= [4817.6, 4942.4]
\]

\[
= [5000, 5000]
\]

\[
= [0.9635, 0.9885]
\]

\[
\text{Required service level} = \frac{\text{5000} \cdot 0.995}{\text{Maximum service level}}
\]

\[
= \frac{5000 \cdot [0.999, 0.999]}{5000}
\]

\[
= \frac{[4975, 4975]}{[5000, 5000]}
\]

\[
= [0.995, 0.995]
\]
For the other risks, we end up with the following risk values:

- The risk value of “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus” is Unacceptable since

  \[ Expected \text{ service level} = [0.9635, 0.9885] \]

  is less than

  \[ \frac{Required \text{ service level}}{Maximum \text{ service level}} = [0.999, 0.999] \]

- The risk value of “Incorrect sensor data is sent to Public telecom system due to invalid control instructions being used by Small hydro power plant” is Acceptable since

  \[ Expected \text{ service level} = [0.9999, 1] \]

  is greater than

  \[ \frac{Required \text{ service level}}{Maximum \text{ service level}} = [0.999, 0.999] \]

- The risk value of “No sensor data messages are sent due to Small hydro power plant being unavailable due to lack of control instructions or use of invalid control instructions” is Unacceptable since

  \[ Expected \text{ service level} = [0.9723, 0.986] \]

  is less than

  \[ \frac{Required \text{ service level}}{Maximum \text{ service level}} = [0.9999, 0.9999] \]

- The risk value of “No sensor data messages are sent due to Small hydro power plant being unavailable due to malicious software” is Unacceptable since

  \[ Expected \text{ service level} = [0.9844, 0.9951] \]

  is less than

  \[ \frac{Required \text{ service level}}{Maximum \text{ service level}} = [0.9999, 0.9999] \]

6 Demonstration of Step 3: Identify indicators for interconnected systems

6.1 Step 3.1: Identify risks to be monitored

Client EPP believes that the likelihood values used to calculate the risk values of the risks “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus” and “Sensor data is sent in plain text from Small hydro power plant to an outsider” may be subject to change. We therefore decide to monitor these risks.
6.2 Step 3.2: Identify relevant indicators for the risks to be monitored

Indicators should be used to monitor likelihood values, since the likelihood values used to calculate the risk values of the two risks may be subject to change. Client EPP does not find it feasible to directly monitor the likelihoods of the unwanted incidents occurring, and has therefore decided to monitor the conditional likelihoods of two leads-to relations in the referenced threat scenario in Figure 19 that affect the likelihoods of the two unwanted incidents occurring. The relevant indicators for the two leads-to relations are presented in Figure 21. In Appendix D.2 we show how to use the conditional likelihoods we now address as well as other factors to monitor the resulting likelihoods of the risks identified for monitoring in Step 3.1.

One composite indicator $c_1$, which aggregates the two basic indicators $b_1$ and $b_2$, has been identified for one leads-to relation. $c_1$ makes a prediction about the percentage of computer viruses that “Small hydro power plant” is not protected against. For the other leads-to relation, we have identified the composite indicator $c_2$, which aggregates the two basic indicators $b_3$ and $b_4$. $c_2$ makes a prediction about the percentage of Trojan horses that “Small hydro power plant” is not protected against.

To calculate the indicators, Client EPP relies on data from the security vendor that delivers the security solutions and patches that are used in the control system of “Small hydro power plant.” At the “Small hydro power plant” it may take some time between each upgrade of the security solutions and patching of the control system. This is due to that the updates and patches need to be inspected and tested before they can be introduced into the control system in order to ensure the stability of the control system of “Small hydro power plant.” The consequence is that “Small hydro power plant” may be unprotected for some time against well-known computer viruses and Trojan horses.

7 Demonstration of Step 4: Specify design and deployment of identified indicators for interconnected systems

7.1 Step 4.1: Specify design of indicators for risk monitoring

In Figure 21 the composite indicators $c_1$ and $c_2$ are associated to one leads-to relation each. Conditional likelihoods were assigned to these leads-to relations during the detailed analysis described in Section 5. Values are therefore obtained for all the basic indicators from the time when the referenced threat scenario in Figure 19 was constructed. For $b_1$ and $b_2$ we obtain the values 750000 and 450000, respectively, while for $b_3$ and $b_4$ we obtain the values 500000 and 200000, respectively.

In Tables 3 and 4 are the design specifications for the different basic and composite indicators. All the specifications have been given in the form of algorithms. The four algorithms are to be used by a risk monitor within the electrical power production infrastructure. The indicators are updated every week. Afterwards, the risk picture is updated based on the updated composite indicators.

To calculate the two composite indicators, Client EPP uses data gathered in its infrastructure to update six lists. These lists are maintained by the risk monitor, and they are used to calculate the basic indicators. Client EPP takes into account that there may be computer viruses and Trojan horses that the security vendor is not aware of. Client EPP thinks it is reasonable to assume that the total number of computer viruses is 0.1 – 0.5% higher than the sum $b_1 + b_2$, and that the total number of Trojan horses is 0.1 – 0.3% higher than the sum $b_3 + b_4$. For both composite indicators we end up with an interval. By using the obtained values for the basic indicators as input to the algorithms of $c_1$ and $c_2$ in Tables 3 and 4, respectively, we get
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

- b_1: Number of computer viruses that Small hydro power plant is protected against
- b_2: Number of computer viruses that Small hydro power plant is not protected against
- b_3: Number of Trojan horses that Small hydro power plant is protected against
- b_4: Number of Trojan horses that Small hydro power plant is not protected against

Figure 21: Relevant indicators, assigned to leads-to relations in the referenced threat scenario in Figure 19, for monitoring the risks “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus” and “Sensor data is sent in plain text from Small hydro power plant to an outsider”
for \( c_1 \), while we get \([0.4008, 0.4025]\) for \( c_2 \). These numbers are almost identical to the initial estimates of 0.6 and 0.4.

### 7.2 Step 4.2: Specify deployment of indicators for risk monitoring

In Table 5 is the deployment specification for the basic and composite indicators. The specification describes how data needed in the calculations of the indicators should be extracted and transmitted within the SoS.

### 8 Related work

The methodological approach presented in this report is a specialization of the approach presented in [11]. The approach in [11] is general in the sense that it only restricts the risk identification to the identified assets and nothing else. In our approach, the risk identification focuses entirely on risk to quality of provided services that have been caused by service dependencies. The approach in [11] can of course be used to identify indicators for the purpose of measuring the impact of service dependencies on risk to quality of provided services, because of its generality. Compared to our approach, however, it is inferior. The approach in [11] does not offer any support for dealing with interconnected systems or service dependencies. In addition, it focuses to a much lesser extent on the calculations of indicators, and it cannot be used to specify how the indicator calculations should be embedded in the systems to be monitored.

We are not aware of other approaches targeting the capture and measure of impact of service dependencies on risks to the quality of provided services. In [12], which is an approach for constructing formal models of services dependencies in information systems, the dependency models are used in security policy-based management. The dependency models are used to find enforcement points for security rules, which then support countermeasure deployment, and for computing the impact of attacks and countermeasures that propagate over the information system.

Service dependencies are also used in fault analysis [13] and dependability analysis [14], as well as in analyses targeting critical infrastructures. A number of the approaches that address service dependencies within critical infrastructures focus primarily on the consequences of infrastructure services not being provided. One such approach is [15]. This approach is used to create models of infrastructure systems and their interactions. The models are used in computer simulations where the main purpose is to investigate how the functionality of infrastructure systems and interconnections react to different attack scenarios (“what if” scenarios where one or two systems are removed), and how mechanisms for strengthening the underlying dependency graph can be used. Svendsen’s approach differs, in particular, from our approach in that the likelihoods of incidents (systems failing to provide services according to requirements) are not considered.

Even though a lot of work has been done within the SoS field, there is still no single accepted definition of what an SoS is. Examples of different definitions may be found in [2]. With different understandings of what an SoS is, we also get different understandings of what should be addressed with respect to risk and security. For instance, some definitions state that an SoS only consists of systems that operate independently of each other, i.e., that the different systems do not rely on services from other systems in order to function. This is quite different from our understanding of an SoS. In the literature, SoS has received relatively little coverage when it comes to risk and security analysis. Papers like [16], [17], [18], and [19], focus primarily on the challenges and relatively little on actual approaches.
Table 3: Design specifications, in the form of algorithms, for the basic indicators $b_1$ and $b_2$ and the composite indicator $c_1$

### Algorithm for $b_1$ and $b_2$

**Input:** $data_1$: “Data on security updates/patches that have been applied in the control system at Small hydro power plant,” $data_2$: “Data on the threat picture, the security updates and patches that are available from the security vendor of Client EPP, and malware that the security vendor is aware of but does not yet offer protection against”

**Data maintained by the risk monitor:** $list_1$: “List of names of computer viruses that the control system at Small hydro power plant is protected against,” $list_2$: “List of names of all computer viruses that the security vendor of Client EPP offers protection against,” $list_3$: “List of names of computer viruses that the security vendor of Client EPP is aware of but does not yet offer protection against”

Based on $data_1$, check whether the security updates/patches applied in the control system have resulted in protection against new computer viruses. Add the names of the new computer viruses to $list_1$, if applicable.

Based on $data_2$, check whether the security vendor offers protection against any new computer viruses. Add the names of the new computer viruses to $list_2$, if applicable. Remove names of computer viruses from $list_3$, if applicable.

Based on $data_2$, check whether there are any new computer viruses that the security vendor is aware of, but does not yet offer protection against. Add the names of the new computer viruses to $list_3$, if applicable.

$b_1 := \text{"The number of items in } list_1\text{"}$

$b_2 := \text{"The number of items in } list_2\text{, where each item is not in } list_1\text{"} +$

$\text{"The number of items in } list_3\text{"}$

**Output:** $b_1, b_2$

### Algorithm for $c_1$

**Input:** $b_1$: “Number of computer viruses that Small hydro power plant is protected against,” $b_2$: “Number of computer viruses that Small hydro power plant is not protected against”

$var_1 := b_2 + ((b_1 + b_2) \cdot [0.001, 0.005])$

$var_2 := b_1 + var_1$

$c_1 := \frac{var_1}{var_2}$

**Output:** $c_1$
Algorithm for $b_3$ and $b_4$

**Input:** $data_1$: “Data on security updates/patches that have been applied in the control system at Small hydro power plant,” $data_2$: “Data on the threat picture, the security updates and patches that are available from the security vendor of Client EPP, and malware that the vendor is aware of but does not yet offer protection against”

**Data maintained by the risk monitor:** $list_4$: “List of names of Trojan horses that Small hydro power plant is protected against,” $list_5$: “List of names of all Trojan horses that the security vendor of Client EPP offers protection against,” $list_6$: “List of names of Trojan horses that the security vendor of Client EPP is aware of but does not yet offer protection against”

Based on $data_1$, check whether the security updates/patches applied in the control system have resulted in protection against new Trojan horses. Add the names of the new Trojan horses to $list_4$, if applicable.

Based on $data_2$, check whether the security vendor offers protection against any new Trojan horses. Add the names of the new Trojan horses to $list_5$, if applicable. Remove names of Trojan horses from $list_6$, if applicable.

Based on $data_2$, check whether there are any new Trojan horses that the security vendor is aware of, but does not yet offer protection against. Add the names of the new Trojan horses to $list_6$, if applicable.

$b_3 := \text{“The number of items in } list_4\text{”}$

$b_4 := \text{“The number of items in } list_5, \text{ where each item is not in } list_4\text{”} + \text{“The number of items in } list_6\text{”}$

**Output:** $b_3, b_4$

Algorithm for $c_2$

**Input:** $b_3$: “Number of Trojan horses that Small hydro power plant is protected against,” $b_4$: “Number of Trojan horses that Small hydro power plant is not protected against”

$var_3 := b_4 + ((b_3 + b_4) \cdot [0.001, 0.003])$

$var_4 := b_3 + var_3$

$c_2 := \frac{var_3}{var_4}$

**Output:** $c_2$
Table 5: Deployment specification for the basic indicators $b_1$, $b_2$, $b_3$, and $b_4$ and the composite indicators $c_1$ and $c_2$

<table>
<thead>
<tr>
<th>Deployment specification for $b_1$, $b_2$, $b_3$, $b_4$, $c_1$, and $c_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of $data_1$:</strong> Client EPP maintains a security database that contains different kinds of information, including how the control system of “Small hydro power plant” is protected against malware. Each time the security solutions of the control system are updated or patches are installed, the information stored about the control system will be updated based on information that comes with the updates/patches. Every week, an automated ICT process extracts data for the control system that the database has been updated with in the period of one week backwards. It should be noticed that the process will extract all the available data for the control system the first time it is executed. We refer to the extracted data as $data_1$. The process transmits $data_1$ to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
<tr>
<td><strong>Extraction and transmission of $data_2$:</strong> The security database of Client EPP also contains information that has been provided by the security vendor used by Client EPP. As part of delivering security solutions to Client EPP, the security vendor provides Client EPP with regular information updates on the threat picture, security updates and patches that are available from the vendor, and malware that the vendor is aware of but does not yet offer protection against. The security database is updated with this information. Every week, an automated ICT process extracts the information that the database has been updated with in the period of one week backwards. It should be noticed that the process will extract all the available information the first time it is executed. We refer to the extracted data as $data_2$. The process transmits $data_2$ to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>
Dependent CORAS [7] is an approach for modular risk modeling, which can be used to document and reason about risk in SoS. It extends the CORAS risk modeling language with facilities for documenting and reasoning about risk analysis assumptions. It was motivated by the need to deal with mutual dependencies in risk analysis of SoS. By employing dependent CORAS we may document risk separately for the individual systems in an SoS. In addition, we document the risk analysis assumptions for the different systems, i.e., how threat scenarios and unwanted incidents, documented for other systems, may lead to threat scenarios and unwanted incidents, documented for the system in question. These assumptions are due to some form of dependencies, not necessarily service dependencies, between the different systems. Thus, dependent CORAS deal with dependencies in a general way compared to our approach, which only focus on service dependencies. The different risk models may be combined in the end, if the dependencies between them are well-founded, i.e., not circular.

Many services need to fulfill quality requirements that are requirements to information security. There exist a number of approaches for measuring information security. One of those is the NIST Performance Measurement Guide for Information Security [20]. This approach aims to assist in the development, selection, and implementation of suitable measures. It also provides a number of candidate measures. Unlike our approach, it is not specialized towards using these measures for the purpose of calculating explicit likelihood and consequence values.

9 Conclusion

In this report we have addressed the issue of how to capture and measure the impact of service dependencies on risk to quality of provided services by the use of measurable indicators. To this end we have put forward a method consisting of four steps. To the best of our knowledge, there exists no similar approach. The applicability of the approach has been demonstrated on an example case within power supply.

In Step 1 of the approach, dependencies due to service interactions between the different interconnected systems are captured. Their impact on risk to quality of provided services is established in Step 2. In Step 3 we identify relevant indicators for monitoring the risks arising from service dependencies, while in Step 4 we specify how likelihood and consequence values associated with the risks should be calculated from sets of indicators and how these calculations should be embedded in the interconnected systems. The result of applying the method is a risk picture capturing the impact of service dependencies on quality of provided services that can be dynamically monitored via the specified indicators.

An interesting topic for further research is the use of leading indicators [21] to monitor the impact of service dependencies on risk to quality of provided services. Many indicators can be viewed as lagging indicators [21]. A lagging indicator that focuses on quality measures something that exists after a shift in quality, e.g. occurrence of unwanted incidents that affects quality assets. Leading indicators, on the other hand, measures something that exists before a shift in quality. In the case of service dependencies, the leading indicators may be used to predict their future impact on risk to quality of provided services. By employing leading indicators, countermeasures may be implemented prior to the risks occurring.

Acknowledgements

The research on which this report describes has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER and NESSoS projects, both funded from the European Community’s Seventh Framework Programme.
(FP7/2007-2013) under grant agreements FP7-216917 and FP7-256980, respectively.

References


A  Assets to be analyzed for provided services

In this appendix and Appendices B and C we demonstrate the use of the methodological approach on the four provided services of Client EPP that the approach was not demonstrated on in Sections 5–7. This appendix focuses on identifying quality assets for the four provided services, specifying scales for measuring likelihood and consequence, and specifying functions for calculating risk values.

On behalf of Client EPP we aim to capture and measure the impact of service dependencies on the quality of the following provided services:

- The control instructions service provided to “Public telecom system.”
- The electricity service provided to “Distribution line 2.”
- The electricity service provided to “Distribution line 3.”
- The electricity service provided to “Transmission line.”

The CORAS asset diagram in Figure 22 presents the relevant quality assets. The control instructions service provided to “Public telecom system” has been assigned three quality assets, while the different electricity services have for the sake of simplicity only been assigned one quality asset each.

Table 6 shows how consequence is measured for the different assets. The meaning of “lack of integrity” and “do not comply with the data confidentiality policy” was explained in Section 5.3. Client EPP has an estimate for the number of control instructions messages to be sent in the period of one year. We will present this estimate later. This estimate is used to calculate consequence values for the control instructions service.

Likelihood is measured as defined in Table 1 on page 31. For the control instructions service, we classify risks as acceptable or unacceptable by the use of Equations (1) and (2) on page 32. For the electricity services, we need to take into account that the required service level for availability is the conjunction of two availability requirements. We classify risks towards these services as follows:

\[
Expected\ service\ level_T = \frac{Maximum\ service\ level_T - (Likelihood \cdot Consequence_T)}{Maximum\ service\ level_T} \tag{3}
\]

Figure 22: Asset diagram presenting the quality assets for which impact of service dependencies should be captured and measured
Table 6: How consequence is measured for the different quality assets

<table>
<thead>
<tr>
<th>Availability of control instructions delivered to Public telecom system</th>
<th>Number of control instructions messages that are not delivered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Confidentiality of control instructions delivered to Public telecom system</td>
<td>Number of control instructions messages sent that do not comply with the data confidentiality policy</td>
</tr>
<tr>
<td>Integrity of control instructions delivered to Public telecom system</td>
<td>Number of control instructions messages that are delivered with lack of integrity</td>
</tr>
<tr>
<td>Availability of electricity delivered to Distribution line 2</td>
<td>Number of hours that the electricity service is unavailable and the amount of electricity (in kilowatt hours) that is not delivered</td>
</tr>
<tr>
<td>Availability of electricity delivered to Distribution line 3</td>
<td>Number of hours that the electricity service is unavailable and the amount of electricity (in kilowatt hours) that is not delivered</td>
</tr>
<tr>
<td>Availability of electricity delivered to Transmission line</td>
<td>Number of hours that the electricity service is unavailable and the amount of electricity (in kilowatt hours) that is not delivered</td>
</tr>
</tbody>
</table>

\[ E = \frac{\text{Maximum service level}_E - (\text{Likelihood} \cdot \text{Consequence}_E)}{\text{Maximum service level}_E} \]  

\[ \text{if } \text{Expected service level}_T \geq \frac{\text{Required service level}_T}{\text{Maximum service level}_T} \text{ and } \text{Expected service level}_E \geq \frac{\text{Required service level}_E}{\text{Maximum service level}_E} \text{ then } \text{Risk value} = \text{Acceptable} \]

\[ \text{else } \text{Risk value} = \text{Unacceptable} \text{ endif} \]

In Equations (3)–(5), \( T \) refers to the requirement that focus on availability with respect to time, while \( E \) refers to the requirement that focus on availability with respect to the electricity delivered. In Appendix C.2.1, we provide an example of the use of the three equations given above.
B Schematic construction of threat diagrams for provided services

This appendix presents high-level threat diagrams for the four provided services of Client EPP that the approach was not demonstrated on in Sections 5–7.

B.1 Control instructions service provided to Public telecom system

For the control instructions service provided by “Home office computer” to “Public telecom system,” the high-level threat diagram in Figure 23 has been schematically constructed from the target model in Figure 12 on page 29. The threat diagram provides a high-level description of the impact of service dependencies on the quality of the control instructions service provided to “Public telecom system.” In the threat diagram we use the abbreviations “sensor data service” and “control instructions service” to refer to the sensor data service provided by “Public telecom system” to “Home office computer” and the control instructions service provided by “Home office computer” to “Public telecom system,” respectively.

B.2 Electricity service provided to Distribution line 2

For the electricity service provided by “Small hydro power plant” to “Distribution line 2,” the high-level threat diagram in Figure 24 has been schematically constructed from the target model.
Figure 24: Threat diagram, which has been schematically constructed from the target model in Figure 12 on page 29, which provides a high-level outline of the impact of service dependencies on the quality of the electricity service provided by “Small hydro power plant” to “Distribution line 2.”

In Figure 12 on page 29, the threat diagram provides a high-level description of the impact of service dependencies on the quality of the electricity service provided to “Distribution line 2.” In the threat diagram we use the abbreviations “control instructions service” and “electricity service” to refer to the control instructions service provided by “Public telecom system” to “Small hydro power plant” and the electricity service provided by “Small hydro power plant” to “Distribution line 2,” respectively.

B.3 Electricity service provided to Distribution line 3

For the electricity service provided by “Small hydro power plant” to “Distribution line 3,” the high-level threat diagram in Figure 25 has been schematically constructed from the target model in Figure 12 on page 29. The threat diagram provides a high-level description of the impact of service dependencies on the quality of the electricity service provided to “Distribution line 3.” In the threat diagram we use the abbreviations “control instructions service” and “electricity service” to refer to the control instructions service provided by “Public telecom system” to “Small hydro power plant” and the electricity service provided by “Small hydro power plant” to
Control instructions service is not delivered according to requirements.

Electricity service is not delivered according to requirements.

Electricity service depends on control instructions service.

Control instructions service is not delivered according to requirements.

Availability of electricity delivered to Distribution line 3

Incident with impact on the availability of the electricity service delivered to Distribution line 3

Figure 25: Threat diagram, which has been schematically constructed from the target model in Figure 12 on page 29, which provides a high-level outline of the impact of service dependencies on the quality of the electricity service provided by “Small hydro power plant” to “Distribution line 3” respectively.

B.4 Electricity service provided to Transmission line

For the electricity service provided by “Large hydro power plant” to “Transmission line,” the high-level threat diagram in Figure 26 has been schematically constructed from the target model in Figure 12 on page 29. The threat diagram provides a high-level description of the impact of service dependencies on the quality of the electricity service provided to “Transmission line.” In the threat diagram we use the following abbreviations for the different services:

- “DL2-CS electricity service” refers to the electricity service provided by “Distribution line 2” to “Control system.”
- “PBS-CS electricity service” refers to the electricity service provided by “Backup power system” to “Control system.”
- “PBS-PTS electricity service” refers to the electricity service provided by “Backup power system” to “Private telecom system.”
• “DL3-PTS electricity service” refers to the electricity service provided by “Distribution line 3” to “Private telecom system.”

• “PTS-LHPP control instructions service” refers to the control instructions service provided by “Private telecom system” to “Large hydro power plant.”

• “LHPP-PTS sensor data service” refers to the sensor data service provided by “Large hydro power plant” to “Private telecom system.”

• “PTS-CS sensor data service” refers to the sensor data service provided by “Private telecom system” to “Control system.”

• “CS-PTS control instructions service” refers to the control instructions service provided by “Control system” to “Private telecom system.”

• “LHPP-TL electricity service” refers to the electricity service provided by “Large hydro power plant” to “Transmission line.”
PTS-LHPP control instructions service and PTS-CS sensor data service depend on DL3-PTS electricity service.

BPS-PTS electricity service is not delivered according to requirements.

BPS-CS electricity service is not delivered according to requirements.

CS-LHPP control instructions service depends on BPS-CS electricity service.

CS-LHPP control instructions service depends on DL2-CS electricity service.

Availability of electricity delivered to Transmission line.

Figure 26: Threat diagram, which has been schematically constructed from the target model in Figure 12 on page 29, which provides a high-level outline of the impact of service dependencies on the quality of the electricity service provided by “Large hydro power plant” to “Transmission line.”
C Capture and measure impact of service dependencies on quality assets of provided services

This appendix presents detailed threat diagrams for the four provided services of Client EPP that the approach was not demonstrated on in Sections 5–7. In addition, it presents relevant indicators for monitoring risk to the quality of the different provided services, and design and deployment specifications for these indicators.

C.1 Control instructions service provided to Public telecom system

C.1.1 Detailed threat diagrams

In Figure 27 is the detailed version of the high-level threat diagram in Figure 23. The referring elements in Figure 27 refer to the referenced threat scenarios provided in Figures 28 and 29, and the referenced unwanted incidents provided in Figure 33. Moreover, the referenced threat scenario in Figure 29 contains three referring threat scenarios, which refer to the referenced

![Figure 27: Detailed version of the high-level threat diagram in Figure 23](image-url)
threat scenarios provided in Figures 30–32. Client EPP has estimated all the likelihood and consequence values in the different figures.

As can be seen in Figure 27, the vulnerability “Control instructions service depends on sensor data service” in Figure 23 has been decomposed into three vulnerabilities. The referenced threat scenario in Figure 28 is a detailing of the referring threat scenario “Sensor data service is not delivered according to requirements” in Figure 23. Since “Public telecom system” is only required to deliver the sensor data service according to the availability requirement, the referenced threat scenario distinguish between the failure of not achieving the availability requirement, and the failures of not achieving the confidentiality and integrity requirements.

Client EPP estimates that 5000 sensor data messages are sent each year to “Home office computer.” Moreover, Client EPP estimates the number of control instructions sent by “Home office computer” in the period of one year to be 1000. Before we can estimate the likelihoods of the sensor data service not being delivered according to the confidentiality, integrity, and availability requirements, we need to calculate the worst-case service levels of the sensor data service delivered by “Public telecom system.” These are as follows:

- $100\% \cdot 0.97 = 97\%$ of the sent sensor messages do comply with the data confidentiality policy;
- $99.99\% \cdot 0.95 = 94.99\%$ of the sent sensor messages are delivered with integrity; and
- $99.99\% \cdot 0.99 = 98.99\%$ of the sent sensor data messages are delivered.

To estimate the likelihoods we use the estimated number of sensor data messages sent each year in combination with the required and worst-case service levels of the sensor data service delivered by “Public telecom system.” The required service levels specify that:

- $5000 \cdot 100\% = 5000$ of the sent sensor data messages should comply with the data confidentiality policy;
- $5000 \cdot 99.99\% = 4999.5$ of the sent sensor data messages should be delivered with integrity; and
Control instructions service is not delivered according to requirements

Sensor data message is not delivered to Home office computer

Sensor data message with lack of integrity is delivered to Home office computer

Sensor data message that do not comply with the data confidentiality policy is sent to Home office computer

Sensor data message with lack of integrity is delivered to Home office computer

Figure 29: The referenced threat scenario “Control instructions service is not delivered according to requirements,” referred to in Figure 27

- \(5000 \cdot 99.99\% = 4999.5\) of the sent sensor data messages should be delivered.

On the other hand, our expectations according to the worst-case service levels are that:

- \(5000 \cdot 97\% = 4850\) out of the 5000 required sensor data messages comply with the data confidentiality policy;
- \(5000 \cdot 94.99\% = 4749.5\) out of the 4999.5 required sensor data messages are delivered with integrity; and
- \(5000 \cdot 98.99\% = 4949.5\) out of the 4999.5 required sensor data messages are delivered.

Based on the calculations for required and worst-case service levels, we end up with the following likelihoods:

- The likelihood of the sensor data service not being delivered according to the confidentiality requirement is “Certain” \((5000 - 4850 = 150\) sensor data messages in the period of a year).
- The likelihood of the sensor data service not being delivered according to the integrity requirement is “Certain” \((4999.5 - 4749.5 = 250\) sensor data messages in the period of a year).
- The likelihood of the sensor data service not being delivered according to the availability requirement is “Certain” \((4999.5 - 4949.5 = 50\) sensor data messages in the period of a year).
Sensor data message is not delivered to Home office computer

Figure 30: The referenced threat scenario “Sensor data message is not delivered to Home office computer,” referred to in Figure 29

The referenced threat scenario “Control instructions service is not delivered according to requirements” is given in Figure 29. The internal threat behavior of “Home office computer” is described by the referenced threat scenarios in Figures 30–32. The different referenced threat scenarios describe how “Home office computer” may fail to deliver the control instructions service according to requirements as a result of “Public telecom system” failing to deliver the sensor data service according to its requirements.

Figure 33 contains the referenced unwanted incidents referred to in Figure 27. For each of the unwanted incidents, with the exception of “No control instructions messages are sent to Public telecom system due to Home office computer being unavailable,” Client EPP assigns the consequence value 1, since each of these incidents only affects one control instructions message. In the case of the incident “No control instructions messages are sent to Public telecom system due to Home office computer being unavailable,” Client EPP believes that the “Home office computer” may be unavailable for as long as 3 days. With an average number of 2.74 (1000/365) control instructions being sent each day, the consequence with respect to the quality asset is 8.

The result of the detailed analysis is five risks. Based on the risk function, defined in Equations (1) and (2) on page 32, the estimated number of control instructions sent each year (1000), and the required service levels for the control instructions service, we can calculate the risk values of the five risks. These are as follows:

- The risk value of “Outsider decrypts control instructions message sent to Public telecom system and discloses the control instructions contained in the message” is Acceptable since

  \[
  \text{Expected service level} = [0.9981, 0.9994]
  \]

  is greater than

\[
\]
Sensor data message with lack of integrity is delivered to Home office computer.

Figure 31: The referenced threat scenario “Sensor data message with lack of integrity is delivered to Home office computer,” referred to in Figure 29

\[
\frac{\text{Required service level}}{\text{Maximum service level}} = [0.995, 0.995]
\]

- The risk value of “Incorrect control instructions are sent to Public telecom system due to fake sensor data message sent by outsider to Home office computer” is Unacceptable

\[
\text{Expected service level} = [0.9981, 0.9994]
\]

is less than

\[
\frac{\text{Required service level}}{\text{Maximum service level}} = [0.999, 0.999]
\]

- The risk value of “Incorrect control instructions are sent to Public telecom system due to use of invalid sensor data” is Acceptable

\[
\text{Expected service level} = [0.9999, 1]
\]

is greater than

\[
\frac{\text{Required service level}}{\text{Maximum service level}} = [0.999, 0.999]
\]

- The risk value of “No control instructions messages are sent to Public telecom system due to Home office computer being unavailable” is Unacceptable

\[
\text{Expected service level} = [0.996, 0.9984]
\]

is less than
Figure 32: The referenced threat scenario “Sensor data message that do not comply with the data confidentiality policy is sent to Home office computer,” referred to in Figure 29
Incident with impact on availability of the control instructions service delivered to Public telecom system

No control instructions messages are sent to Public telecom system due to lack of sensor data or use of invalid sensor data

[Very likely]

Incident with impact on confidentiality of the control instructions service delivered to Public telecom system

Outsider decrypts control instructions message sent to Public telecom system and discloses the control instructions contained in the message

[Unlikely]

Incident with impact on integrity of the control instructions service delivered to Public telecom system

Incorrect control instructions are sent to Public telecom system due to fake sensor data message sent by outsider to Home office computer

[Likely]

Incorrect control instructions are sent to Public telecom system due to use of invalid sensor data

[Rare]

Figure 33: The referenced unwanted incidents “Incident with impact on confidentiality of the control instructions service delivered to Public telecom system,” “Incident with impact on integrity of the control instructions service delivered to Public telecom system,” and “Incident with impact on availability of the control instructions service delivered to Public telecom system,” referred to in Figure 27

\[
\frac{\text{Required service level}}{\text{Maximum service level}} = [0.9999, 0.9999]
\]

- The risk value of “No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data” is Unacceptable

\[
\text{Expected service level} = [0.9501, 0.99]
\]

is less than

\[
\frac{\text{Required service level}}{\text{Maximum service level}} = [0.9999, 0.9999]
\]
Sensor data message is not delivered to Home office computer

No action is taken by the system operator since he/she fails to detect the alarm [Rare]

An alarm is generated as a result of the control instructions message being discarded [Rare]

Control instructions message is discarded by the system because the control instructions are not valid [Rare]

Figure 34: Relevant indicators, assigned to leads-to relations in the referenced threat scenario in Figure 30 for monitoring the risk “No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data”

C.1.2 Relevant indicators for risk monitoring

Client EPP believes that the likelihood value used to calculate the risk value of the risk “No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data” may be subject to change. We therefore decide to monitor this risk.

Indicators should be used to monitor likelihood values, since the likelihood value used to calculate the risk value of the risk may be subject to change. Client EPP does not find it feasible to directly monitor the likelihood of the unwanted incident occurring, and has therefore decided to monitor the conditional likelihoods of three leads-to relations in the referenced threat scenarios in Figures 30 and 31 that affect the likelihood of the unwanted incident occurring. The relevant indicators for the three leads-to relations are presented in Figures 34 and 35. In Appendix D.3 we show how to use the conditional likelihoods we now address as well as other factors to monitor the resulting likelihood of the risk identified for monitoring.

One composite indicator $c_3$, which aggregates the two basic indicators $b_5$ and $b_6$, has been
Sensor data message with lack of integrity is delivered to Home office computer.

Figure 35: Relevant indicators, assigned to a leads-to relation in the referenced threat scenario in Figure 31, for monitoring the risk “No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data.”

\( b_5 \): Number of sensor data messages that were sent by Small hydro power plant to Home office computer in the last three months, where each message required a response in the form of a control instructions message from Home office computer.

\( b_6 \): Total Number of sensor data messages that were sent by Small hydro power plant to Home office computer in the last three months.

---

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor data message is delivered, its integrity has been changed during transmission or while at Public telecom system</td>
<td>[Certain]</td>
</tr>
<tr>
<td>The sensor data message requires a response in the form of a control instructions message</td>
<td>[Very likely]</td>
</tr>
<tr>
<td>Control instructions message is created based on invalid sensor data</td>
<td>[Rare]</td>
</tr>
<tr>
<td>Insufficient check of whether the control instructions are valid</td>
<td>0.2</td>
</tr>
<tr>
<td>Insufficient check of whether the sensor data is valid</td>
<td>0.5</td>
</tr>
<tr>
<td>Re-transmission of sensor data message is not requested</td>
<td>[Rare]</td>
</tr>
<tr>
<td>Possible that checksum algorithm fails to detect integrity violations</td>
<td>0.001</td>
</tr>
<tr>
<td>Control instructions message is not discarded by the system</td>
<td>[Rare]</td>
</tr>
</tbody>
</table>

---

66
identified for two of the leads-to relations. \(c_3\) calculates the ratio of sensor data messages that required responses in form of control instructions messages to all sensor data messages. For the third leads-to relation, we have identified the composite indicator \(c_4\), which aggregates the two basic indicators \(b_7\) and \(b_8\). \(c_4\) calculates the ratio of alarms where appropriate actions were taken by the system operator to all alarms generated.

### C.1.3 Design and deployment of indicators for risk monitoring

In Figure 34, the composite indicators \(c_3\) and \(c_4\) are associated with one leads-to relation each. Moreover, \(c_3\) is also associated with one leads-to relation in Figure 35. Conditional likelihoods were assigned to all of these leads-to relations during the detailed analysis described in Appendix C.1.1. Values are therefore obtained for all the basic indicators from the time when the referenced threat scenarios in Figures 30 and 31 were constructed. For \(b_5\) and \(b_6\) we obtain the values 590 and 1255, respectively, while for \(b_7\) and \(b_8\) we obtain the value 20 for both.

In Tables 7–9 are the design specifications for the different basic and composite indicators. All the specifications have been given in the form of algorithms. The four algorithms are to be used by a risk monitor within the electrical power production infrastructure. The indicators are updated each month. Afterwards, the risk picture is updated based on the updated composite indicators.

The input \(\text{data}_1\) is used by the algorithm for \(b_5\) and \(b_6\), while the inputs \(\text{data}_2\) and \(\text{data}_3\) are used by the algorithm for \(b_7\) and \(b_8\). It should be noticed that the first time the two algorithms are executed, the input \(\text{data}_1\) consists of events that have been generated during the last three months, while the inputs \(\text{data}_2\) and \(\text{data}_3\) consist of events that have been generated during the last six months. This has been done in order to ensure correct calculation of the indicators. For all other executions of the two algorithms, the inputs will only consist of events that have been generated during the last month.

The composite indicator \(c_3\) aggregates the two indicators \(b_5\) and \(b_6\). As can be seen in Figures 30 and 31, Client EPP has estimated that between 20% and 50% of the sensor data messages require responses in the form of control instructions messages. Thus, the probability interval \([0.2, 0.5]\). Client EPP finds it very likely that most values of \(c_3\) should be contained in this interval, but is also aware of that some values may be lower than 0.2 or higher than 0.5. In Client EPP’s opinion, the minimum value for \(c_3\) should be 0.1. Thus, if the aggregation of \(b_5\) and \(b_6\) results in a value less than 0.1, then \(c_3\) is assigned the value 0.1. It should be noticed that we do not perform any checks of whether \(b_6\) is zero in the design specification in Table 7. This is due to that \(b_6\) will never be zero. By using the obtained values for the basic indicators as input to the algorithm we get 0.47. This number is in accordance with the initial estimate of \([0.2, 0.5]\).

The composite indicator \(c_4\) aggregates the two indicators \(b_7\) and \(b_8\). Client EPP is of the opinion that the system operator fails to notice at least 1% of the alarms. Thus, the minimum value for \(c_4\) should be 0.01. If \(b_7\) does not equal zero, then \(c_4\) is 1 minus the ratio of \(b_8\) to \(b_7\). If the result of this calculation is less than 0.01, then \(c_4\) is assigned the minimum value of 0.01. By using the obtained values for the basic indicators as input to the algorithm we get 0.01. This number is in accordance with the initial estimate of 0.01.

In Tables 10 and 11 are the deployment specifications for the basic and composite indicators.
Table 7: Design specifications, in the form of algorithms, for the basic indicators \( b_5 \) and \( b_6 \) and the composite indicator \( c_3 \)

### Algorithm for \( b_5 \) and \( b_6 \)

**Input:** \( data_1 \): “Events generated by the control system at Small hydro power plant, where each event was generated as a result of sending a sensor data message”

**Data maintained by the risk monitor:** \( event \ log_1 \): “Events generated by the control system at Small hydro power plant during the last three months, where each event represents the sending of a sensor data message which required a response in the form of a control instructions message,” \( event \ log_2 \): “Events generated by the control system at Small hydro power plant during the last three months, where each event represents the sending of a sensor data message”

Remove all events from \( event \ log_1 \) that were generated for more than three months ago. Extract all events from \( data_1 \) that required a response in the form of a control instructions message. Add the extracted events to \( event \ log_1 \).

Remove all events from \( event \ log_2 \) that were generated for more than three months ago. Extract all events from \( data_1 \). Add the extracted events to \( event \ log_2 \).

\( b_5 := \) “The number of events in \( event \ log_1 \)”
\( b_6 := \) “The number of events in \( event \ log_2 \)”

**Output:** \( b_5, b_6 \)

### Algorithm for \( c_3 \)

**Input:** \( b_5 \): “Number of sensor data messages that were sent by Small hydro power plant to Home office computer in the last three months, where each message required a response in the form of a control instructions message from Home office computer,” \( b_6 \): “Total number of sensor data messages that were sent by Small hydro power plant to Home office computer in the last three months”

\( c_3 := \frac{b_5}{b_6} \)

\( \text{if } c_3 < 0.1 \text{ then } \)
\( c_3 := 0.1 \)

**end if**

**Output:** \( c_3 \)
Table 8: Design specification, in the form of an algorithm, for the basic indicators \( b_7 \) and \( b_8 \)

<table>
<thead>
<tr>
<th>Algorithm for ( b_7 ) and ( b_8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> ( data_2 ): “Events generated by the Home office computer, where each event represents an alarm,” ( data_3 ): “Events generated by the Home office computer, where each event represents the response to an alarm”</td>
</tr>
<tr>
<td><strong>Data maintained by the risk monitor:</strong> ( event \ log_3 ): “Events generated by the Home office computer during the last six months, where each event represents an alarm that was generated as a result of a control instructions message being discarded due to not being valid,” ( event \ log_4 ): “Events generated by the Home office computer during the last six months, where each event represents the system operator responding to an alarm generated as a result of a control instructions message being discarded due to not being valid”</td>
</tr>
<tr>
<td>Remove all events from ( event \ log_3 ) that were generated for more than six months ago. Remove all events from ( event \ log_4 ) that were generated for more than six months ago.</td>
</tr>
<tr>
<td>Extract all events from ( data_2 ) where each event represents the generation of an alarm as a result of a control instructions message being discarded due to not being valid. Add the extracted events to ( event \ log_3 ). Extract all events from ( data_3 ) where each event represents that the system operator responded to an alarm generated as a result of a control instructions message being discarded due to the control instructions not being valid. Add the extracted events to ( event \ log_4 ).</td>
</tr>
<tr>
<td>( b_7 := ) “The number of events in ( event \ log_3 )” ( b_8 := ) “The number of events in ( event \ log_4 )”</td>
</tr>
<tr>
<td><strong>Output:</strong> ( b_7, b_8 )</td>
</tr>
</tbody>
</table>
Table 9: Design specification, in the form of an algorithm, for the composite indicator \( c_4 \)

<table>
<thead>
<tr>
<th>Algorithm for ( c_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> ( b_7 ): “Number of alarms generated by Home office computer in the last six months as a result of control instructions messages being discarded due to not being valid,” ( b_8 ): “Number of alarms generated by Home office computer in the last six months as a result of control instructions messages being discarded due to not being valid, where the system operator detected the alarms and took appropriate actions”</td>
</tr>
<tr>
<td>( \text{if } b_7 \neq 0 \text{ then} )</td>
</tr>
<tr>
<td>( c_4 := 1 - \frac{b_8}{b_7} )</td>
</tr>
<tr>
<td>( \text{else} )</td>
</tr>
<tr>
<td>( c_4 := 0.01 )</td>
</tr>
<tr>
<td>( \text{end if} )</td>
</tr>
<tr>
<td>( \text{if } c_4 &lt; 0.01 \text{ then} )</td>
</tr>
<tr>
<td>( c_4 := 0.01 )</td>
</tr>
<tr>
<td>( \text{end if} )</td>
</tr>
<tr>
<td><strong>Output:</strong> ( c_4 )</td>
</tr>
</tbody>
</table>

Table 10: Deployment specification for the basic indicators \( b_5 \) and \( b_6 \) and the composite indicator \( c_3 \)

<table>
<thead>
<tr>
<th>Deployment specification for ( b_5 ), ( b_6 ), and ( c_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of ( data_1 ):</strong> The control system at the “Small hydro power plant” has an event log that contains different events generated by the control system. At the start of each month, an automated ICT process extracts all events from the event log that have been generated as a result of sending sensor data messages to “Home office computer” and where each event was generated during the last month. It should be noticed that the process will extract all events that have been generated during the last three months the first time it is executed. We refer to the extracted data as ( data_1 ). The process transmits ( data_1 ) to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>
Table 11: Deployment specification for the basic indicators $b_7$ and $b_8$ and the composite indicator $c_4$

<table>
<thead>
<tr>
<th>Deployment specification for $b_7$, $b_8$, and $c_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of $data_2$:</strong> The “Home office computer” has an event log that contains different events generated by the computer. At the start of each month, an automated ICT process extracts all events from the event log that represent alarms and where each event was generated during the last month. It should be noticed that the process will extract all events that have been generated during the last six months the first time it is executed. We refer to the extracted data as $data_2$. The process transmits $data_2$ to the risk monitor by using the public telecom infrastructure.</td>
</tr>
<tr>
<td><strong>Extraction and transmission of $data_3$:</strong> At the start of each month, an automated ICT process extracts all events from the event log that represent responses to alarms and where each event was generated during the last month. It should be noticed that the process will extract all events that have been generated during the last six months the first time it is executed. We refer to the extracted data as $data_3$. The process transmits $data_3$ to the risk monitor by using the public telecom infrastructure.</td>
</tr>
</tbody>
</table>
C.2 Electricity service provided to Distribution line 2

C.2.1 Detailed threat diagrams

In Figure 36 is the detailed version of the high-level threat diagram in Figure 24 on page 54. The referring elements in Figure 36 refer to the referenced threat scenarios provided in Figures 37 and 38, and the referenced unwanted incident provided in Figure 42. Moreover, the referenced threat scenario in Figure 38 contains three referring threat scenarios, which refer to the referenced threat scenarios provided in Figures 39–41. Client EPP has estimated all the likelihood and consequence values in the different figures.

As can be seen in Figure 36, the vulnerability “Electricity service depends on control instructions service” in Figure 24 has been decomposed into three vulnerabilities. The referenced threat scenario in Figure 37 is a detailing of the referring threat scenario “Control instructions service is not delivered according to requirements” in Figure 24. Since “Public telecom sys-
Control instructions service is not delivered according to requirements

Figure 37: The referenced threat scenario “Control instructions service is not delivered according to requirements,” referred to in Figure 36

tem” is only required to deliver the control instructions service according to the availability requirement, the referenced threat scenario distinguish between the failure of not achieving the availability requirement, and the failures of not achieving the confidentiality and integrity requirements.

Client EPP estimates that 1000 control instructions messages are sent each year to “Small hydro power plant.” Moreover, Client EPP estimates the maximum amount of electricity delivered in the period of one year to each of “Distribution line 2” and “Distribution line 3” to be 37 MWh. The likelihoods of the control instructions service not being delivered according to the confidentiality, integrity, and availability requirements are identical to the ones calculated in Section 5.3. The likelihoods are as follows:

- The likelihood of the control instructions service not being delivered according to the confidentiality requirement is “Very likely.”
- The likelihood of the control instructions service not being delivered according to the integrity requirement is “Certain.”
- The likelihood of the control instructions service not being delivered according to the availability requirement is “Very likely.”

The referenced threat scenario “Electricity service is not delivered according to requirements” is given in Figure 38. The internal threat behavior of “Small hydro power plant” is described by the referenced threat scenarios in Figures 39–41. The different referenced threat scenarios describe how “Small hydro power plant” may fail to deliver the electricity service according to requirements as a result of “Public telecom system” failing to deliver the control instructions service according to its requirements.

Figure 42 contains the referenced unwanted incident referred to in Figure 36. For most of the unwanted incidents, with the exceptions of “Small hydro power plant is shut down due to malicious software” and “Small hydro power plant is shut down due to damage to unstable power generator,” Client EPP assign the consequence value of “24 h and 101 kWh” (h is hours and kWh is kilowatt hours) with respect to the quality asset. Client EPP believes that the electricity
Electricity service is not delivered according to requirements

Figure 38: The referenced threat scenario “Electricity service is not delivered according to requirements,” referred to in Figure 36

Control instructions message is not delivered to Small hydro power plant

Control instructions message with lack of integrity is delivered to Small hydro power plant

Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Figure 39: The referenced threat scenario “Control instructions message is not delivered to Small hydro power plant,” referred to in Figure 38
Service will not be provided for 24 hours. 101 kWh ($\frac{37000}{365}$) is the average amount of electricity produced for “Distribution line 2” in one day. For the incident “Small hydro power plant is shut down due to malicious software,” Client EPP believes that “Small hydro power plant” will be shut down for three days (72 hours). Moreover, for the unwanted incident “Small hydro power plant is shut down due to damage to unstable power generator,” Client EPP believes that such an incident may result in a down time of 31 days (744 hours), since it is very likely that the power generator needs to be replaced as a result of the incident.

The result of the detailed analysis is five risks. Based on the risk function, defined in Equations (3)–(5) on pages 51 and 52, the maximum service levels $Maximum service level_T$ (24 hours $\cdot$ 365 days = 8760 hours) and $Maximum service level_E$ (37 MWh per year), and the two availability requirements specified in the required service level of the electricity service, we can calculate the risk values of the five risks.

In the case of the risk “Small hydro power plant is shut down due to lack of control instructions for correcting errors,” the Expected service level $T$ is less than $Required service level_T$, while the Expected service level $E$ is less than $Required service level_E$. This means that the risk value is Unacceptable. Below we present the calculations for this case. Notice that all the values are for

![Diagram](image-url)
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Very likely

The outsider constructs his own encrypted control instructions message

Unlikely

The outsider sends an encrypted control instructions message to Small hydro power plant that contains malicious code

Unlikely

The outsider sends an encrypted control instructions message to Small hydro power plant that contains control instructions for increasing the output of the power generator

Unlikely

Power generator becomes unstable

Unlikely

Uncontrolled shutdown of the power generator due to safety system failure

Rare

An outsider intercepted the control instructions message that do not comply with the data confidentiality policy during transmission

Likely

Control instructions message format is easy to understand

Malware protection not up to date

0.5

The outsider constructs his own encrypted control instructions message

0.7

The outsider sends an encrypted control instructions message to Small hydro power plant that contains malicious code

Unlikely

Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Figure 41: The referenced threat scenario “Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant,” referred to in Figure 38
Incident with impact on the availability of the electricity service delivered to Distribution line 2

Small hydro power plant is shut down due to malicious software [Unlikely]

Small hydro power plant is shut down due to damage to unstable power generator [Rare]

Small hydro power plant is shut down due to an automatic shutdown of unstable power generator [Unlikely]

Small hydro power plant is shut down due to use of invalid control instructions [Rare]

Small hydro power plant is shut down due to lack of control instructions for correcting errors [Likely]

Figure 42: The referenced unwanted incident “Incident with impact on the availability of the electricity service delivered to Distribution line 2,” referred to in Figure 36
the period of one year.

\[
\text{Expected service level}_T = \frac{\text{Maximum service level}_T - (\text{Likelihood} \cdot \text{Consequence}_T)}{\text{Maximum service level}_T} \\
= \frac{8760 - ([5, 9.9] \cdot 24)}{8760} \\
= \frac{8760, 8760) - ([5, 9.9] \cdot [24, 24])}{[8760, 8760]} \\
= \frac{8760, 8760) - 120, 237.6}{[8760, 8760]} \\
= [8522.4, 8640] \\
= [0.9729, 0.9863]
\]

\[
\text{Required service level}_T = \frac{8760 \cdot 0.995}{8760} \\
= \frac{[8760, 8760] \cdot [0.999, 0.999]}{[8760, 8760]} \\
= \frac{[8751.24, 8751.24]}{[8760, 8760]} \\
= [0.9999, 0.999]
\]

\[
\text{Expected service level}_E = \frac{\text{Maximum service level}_E - (\text{Likelihood} \cdot \text{Consequence}_E)}{\text{Maximum service level}_E} \\
= \frac{37000 - ([5, 9.9] \cdot 101)}{37000} \\
= \frac{[37000, 37000] - ([5, 9.9] \cdot [101, 101])}{[37000, 37000]} \\
= \frac{[37000, 37000] - [505, 999.9]}{[37000, 37000]} \\
= [36000.1, 36495] \\
= [0.973, 0.9864]
\]

\[
\text{Required service level}_E = \frac{36980}{37000} \\
= 0.9995 \\
= [0.9995, 0.9995]
\]

For the other risks, we end up with the following risk values:

- The risk value of “Small hydro power plant is shut down due to use of invalid control instructions” is **Acceptable** since

\[
\text{Expected service level}_T = [0.9997, 1]
\]

is greater than
\[
\frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999]
\]

and since
\[
\text{Expected service level}_E = [0.9997, 1]
\]
is greater than
\[
\frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995]
\]

- The risk value of “Small hydro power plant is shut down due to an automatic shutdown of unstable power generator” is \textit{Unacceptable} since
  \[
  \text{Expected service level}_T = [0.9948, 0.9984]
  \]
is less than
\[
\frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999]
\]
and since
\[
\text{Expected service level}_E = [0.9948, 0.9984]
\]
is less than
\[
\frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995]
\]

- The risk value of “Small hydro power plant is shut down due to damage to unstable power generator” is \textit{Unacceptable} since
  \[
  \text{Expected service level}_T = [0.9915, 1]
  \]
is less than
\[
\frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999]
\]
and since
\[
\text{Expected service level}_E = [0.9915, 1]
\]
is less than
\[
\frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995]
\]

- The risk value of “Small hydro power plant is shut down due to malicious software” is \textit{Unacceptable} since
  \[
  \text{Expected service level}_T = [0.9844, 0.9951]
  \]
is less than
\[
\frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999]
\]
and since
Expected service level \( E \) = [0.9844, 0.9951]

is less than

\[
\frac{\text{Required service level}}{\text{Maximum service level}_E} = [0.9995, 0.9995]
\]

C.2.2 Relevant indicators for risk monitoring

Client EPP believes that both the likelihood value and the consequence value used to calculate the risk value of the risk “Small hydro power plant is shut down due to damage to unstable power generator” may be subject to change. We therefore decide to monitor this risk.

Client EPP does not find it feasible to directly monitor the likelihood of the unwanted incident occurring, and has therefore decided to monitor the conditional likelihood of a leads-to relation in the referenced threat scenario in Figure 41 that affects the likelihood of the unwanted incident occurring. In Figure 43 are relevant indicators for monitoring the conditional likelihood of the leads-to relation in Figure 41, while in Figure 44 are relevant indicators for monitoring the consequence of the impacts relation between the unwanted incident and the quality asset in the detailed high-level threat diagram in Figure 36. In Appendix D.4 we show how to use the conditional likelihood and the consequence we now address as well as other factors to monitor the risk value of the risk identified for monitoring.

One composite indicator \( c_5 \), which aggregates the two basic indicators \( b_9 \) and \( b_{10} \), has been identified for the leads-to relation, while two composite indicators \( c_6 \) and \( c_7 \), where both aggregate the two basic indicators \( b_{11} \) and \( b_{12} \), have been identified for the impacts relation. Client EPP relies on simulations to test the stability of the safety system. The software simulator uses data provided by sensors that monitors the state of the power generator and the safety system. In addition, the software simulator uses the ages of the power generator and the safety system as well as data on their previous failures as input. To monitor the basic indicators of the composite indicators \( c_6 \) and \( c_7 \), Client EPP relies on information from the vendor producing the power generators used by Client EPP, and information from the company maintaining and installing these generators.

C.2.3 Design and deployment of indicators for risk monitoring

In Figure 43 the composite indicator \( c_5 \) is associated with a leads-to relation, while in Figure 44 the composite indicators \( c_6 \) and \( c_7 \) are associated with an impacts relation. A conditional likelihood was assigned to the leads-to relation during the detailed analysis described in Appendix C.2.1, while a consequence value was assigned to the impacts relation associated with \( c_6 \) and \( c_7 \) during the same detailed analysis. Values are therefore obtained for all the basic indicators from the time when the detailed high-level threat diagram and the referenced threat scenario in Figures 36 and 41, respectively, were constructed. For \( b_9 \) and \( b_{10} \) we obtain the values 3 and 9997, respectively, while for \( b_{11} \) and \( b_{12} \) we obtain the values 28 and 3, respectively.

In Tables 12–14 are the design specifications for the different basic and composite indicators with the exception of the basic indicators \( b_{11} \) and \( b_{12} \). These two basic indicators are so simple that no design specifications are needed. All the specifications have been given in the form of algorithms. The three algorithms are to be used by a risk monitor within the electrical power production infrastructure. The indicators \( b_9 \), \( b_{10} \), and \( c_5 \) are updated each week, while the indicators \( b_{11} \), \( b_{12} \), \( c_6 \), and \( c_7 \) are updated every two weeks. The risk picture is updated after each composite indicator has been updated.

The algorithm for the two basic indicators \( b_9 \) and \( b_{10} \) is given in Table 12. It should be noticed that the two inputs \( data_5 \) and \( data_6 \) are only provided the first time the algorithm is
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

An outsider intercepted the control instructions message that do not comply with the data confidentiality policy during transmission [Likely]

The outsider sends an encrypted control instructions message to Small hydro power plant that contains control instructions for increasing the output of the power generator [Unlikely]

Control instructions message format is easy to understand [Very likely]

The outsider constructs his own encrypted control instructions message [Unlikely]

Malware protection not up to date [Unlikely]

Small hydro power plant starts to operate in an incorrect state [Unlikely]

Unstable safety system [Rare]

Power generator becomes unstable [Unlikely]

Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Figure 43: Relevant indicators, assigned to leads-to relations in the referenced threat scenario in Figure 41, for monitoring the risk “Small hydro power plant is shut down due to damage to unstable power generator”
executed. The reason is that the two inputs will not change as long as the power generator and the safety system are not replaced. As can be seen in Table 12, the two inputs are used to initialize the two data items $time_1$ and $time_2$.

The composite indicator $c_5$ aggregates the two indicators $b_9$ and $b_{10}$. Client EPP understands that simulations cannot provide perfect predictions about the future, and decides therefore to come up with a minimum and a maximum value for $c_5$. Client EPP is of the opinion that the minimum value of $c_5$ should be 0.0001 (1 out of 10000 shutdowns of the unstable power generator results in an uncontrolled shutdown), and that the maximum value of $c_5$ should be 0.001 (10 out of 10000 shutdowns of the unstable power generator results in an uncontrolled shutdown). $c_5$ is calculated as the ratio of $b_9$ to $b_9 + b_{10}$. If $c_5$ is less than 0.0001 or greater than 0.001, then $c_5$ is assigned the value 0.0001 or the value 0.001, respectively. By using the obtained values for the basic indicators as input to the algorithm we get 0.0003, which is in accordance with the initial estimate of [0.0001, 0.001].

The composite indicators $c_6$ and $c_7$ both aggregate the two indicators $b_{11}$ and $b_{12}$. It should be noticed that neither of the indicators $b_{11}$ and $b_{12}$ can be equal to zero. The number 101.37 used to calculate $c_7$ is the average amount of electricity in kilowatt hours that is produced in one day by “Small hydro power plant” for “Distribution line 2.” By using the obtained values for the basic indicators as input to the algorithm we get 744 for $c_6$, while we get 3142 for $c_7$. This is of course in accordance with the initial consequence estimate.

In Tables 15 and 16 are the deployment specifications for the basic and composite indicators.
Table 12: Design specifications, in the form of algorithms, for the basic indicators $b_9$ and $b_{10}$

<table>
<thead>
<tr>
<th>Algorithm for $b_9$ and $b_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $data_1$: “Sensor data for the period of one week backwards that describes the state of the power generator at Small hydro power plant,” $data_2$: “Sensor data for the period of one week backwards that describes the state of the safety system at Small hydro power plant,” $data_3$: “Data on previous failures for the power generator at Small hydro power plant,” $data_4$: “Data on previous failures for the safety system at Small hydro power plant,” $data_5$: “The installation time for the power generator at Small hydro power plant,” $data_6$: “The installation time for the safety system at Small hydro power plant”</td>
</tr>
<tr>
<td><strong>Data maintained by the risk monitor:</strong> $list_1$: “List containing data on all previous failures for the power generator at Small hydro power plant,” $list_2$: “List containing data on all previous failures for the safety system at Small hydro power plant,” $time_1$: “The installation time for the power generator at Small hydro power plant,” $time_2$: “The installation time for the safety system at Small hydro power plant”</td>
</tr>
<tr>
<td>if First time the algorithm is executed then</td>
</tr>
<tr>
<td>$time_1 := data_5$, $time_2 := data_6$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>Based on $time_1$, calculate the age $age_1$ of the power generator</td>
</tr>
<tr>
<td>Based on $time_2$, calculate the age $age_2$ of the safety system</td>
</tr>
<tr>
<td>Update $list_1$ based on $data_3$</td>
</tr>
<tr>
<td>Update $list_2$ based on $data_4$</td>
</tr>
<tr>
<td>Initialize the software simulator with $data_1, data_2, list_1, list_2, age_1, age_2$</td>
</tr>
<tr>
<td>$i := 0$, $b_9 := 0$, $b_{10} := 0$</td>
</tr>
<tr>
<td>Start software simulator</td>
</tr>
<tr>
<td><strong>while</strong> $i &lt; 10000$ <strong>do</strong></td>
</tr>
<tr>
<td>Simulate a shutdown of an unstable power generator by the use of the safety system</td>
</tr>
<tr>
<td>if Uncontrolled shutdown of the unstable power generator then</td>
</tr>
<tr>
<td>$b_9 := b_9 + 1$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>$b_{10} := b_{10} + 1$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>$i := i + 1$</td>
</tr>
<tr>
<td>end while</td>
</tr>
<tr>
<td>Shutdown software simulator</td>
</tr>
<tr>
<td><strong>Output:</strong> $b_9, b_{10}$</td>
</tr>
</tbody>
</table>
Table 13: Design specification, in the form of an algorithm, for the composite indicator $c_5$

<table>
<thead>
<tr>
<th>Algorithm for $c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $b_9$: “Number of computer simulations that simulated a shutdown of an unstable power generator by the use of the safety system, where all the simulations resulted in an uncontrolled shutdown,” $b_{10}$: “Number of computer simulations that simulated a shutdown of an unstable power generator by the use of the safety system, where all the simulations resulted in a controlled shutdown”</td>
</tr>
<tr>
<td>$c_5 := \frac{b_9}{b_9 + b_{10}}$</td>
</tr>
<tr>
<td>if $c_5 &lt; 0.0001$ then</td>
</tr>
<tr>
<td>$c_5 := 0.0001$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>if $c_5 &gt; 0.001$ then</td>
</tr>
<tr>
<td>$c_5 := 0.001$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td><strong>Output:</strong> $c_5$</td>
</tr>
</tbody>
</table>

Table 14: Design specification, in the form of an algorithm, for the composite indicators $c_6$ and $c_7$

<table>
<thead>
<tr>
<th>Algorithm for $c_6$ and $c_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $b_{11}$: “Number of days it takes to get a new power generator delivered,” $b_{12}$: “Number of days it takes to get a new power generator installed”</td>
</tr>
<tr>
<td>$c_6 := 24 \cdot (b_{11} + b_{12})$</td>
</tr>
<tr>
<td>$c_7 := 101.37 \cdot (b_{11} + b_{12})$</td>
</tr>
<tr>
<td><strong>Output:</strong> $c_6, c_7$</td>
</tr>
</tbody>
</table>
Table 15: Deployment specification for the basic indicators $b_9$ and $b_{10}$ and the composite indicator $c_5$

<table>
<thead>
<tr>
<th>Deployment specification for $b_9$, $b_{10}$, and $c_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of data</strong> $1$, $data_2$, $data_3$, $data_4$, $data_5$, and $data_6$: Client EPP has a maintenance database that contains information about different components and systems in the electrical power production infrastructure, including the power generator and the safety system at “Small hydro power plant.” In the case of the power generator and the safety system, the database is updated at least on a daily basis by sensors that monitor the state of power generator and the safety system. Besides being updated by sensors, the database is also updated manually by humans. At the start of each week, an automated ICT process extracts all new sensor data that the database has been updated with in the period of one week backwards. We refer to the extracted sensor data for the power generator and the safety system as $data_1$ and $data_2$, respectively. If the power generator and/or the safety system experienced failures in the previous week, then the process extracts the data describing these failures. We refer to the extracted data describing previous failures for the power generator and the safety system as $data_3$ and $data_4$, respectively. The first time the automated ICT process is executed, it will extract all available data on previous failures for the power generator ($data_3$) and the safety system ($data_4$). It will also extract the installation time for the power generator and the safety system from the database. We refer to the former and the latter as $data_5$ and $data_6$, respectively. After having extracted the different data, the process transmits the data to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>

Table 16: Deployment specification for the basic indicators $b_{11}$ and $b_{12}$ and the composite indicators $c_6$ and $c_7$

<table>
<thead>
<tr>
<th>Deployment specification for $b_{11}$, $b_{12}$, $c_6$, and $c_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of</strong> $b_{11}$ and $b_{12}$: Every two weeks, an employee of Client EPP obtains the expected delivery time in days for a new power generator from the vendor producing the power generators used in the electrical power production infrastructure. The number obtained is the basic indicator $b_{11}$. The employee also obtains the expected installation time in days for a new power generator from the company that Client EPP uses for installing and maintaining power generators. The number obtained is the basic indicator $b_{12}$. The employee updates the maintenance database of Client EPP with these two numbers. After the database has been updated, an automated ICT process extracts $b_{11}$ and $b_{12}$. The process transmits $b_{11}$ and $b_{12}$ to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>
C.3 Electricity service provided to Distribution line 3

C.3.1 Detailed threat diagrams

In Figure 45 is the detailed version of the high-level threat diagram in Figure 25 on page 55. With the only exceptions of the quality assets and the names of the referring unwanted incidents of Figures 36 and 45 being different, Figure 45 is identical to Figure 36.

The two electricity services provided to “Distribution line 2” and “Distribution line 3” share the referenced threat scenarios in Figures 37–41, since electricity cannot be provided by “Small hydro power plant” to “Distribution line 3” if it cannot be provided to “Distribution line 2” and vice versa. In Figure 46 is the referenced unwanted incident referred to in Figure 45. With the only exception of the names of the two referenced unwanted incidents in Figures 42 and 46 being different, Figure 46 is identical to Figure 42.

Figure 45: Detailed version of the high-level threat diagram in Figure 25 on page 55
Figure 46: The referenced unwanted incident “Incident with impact on the availability of the electricity service delivered to Distribution line 3,” referred to in Figure 45

C.3.2 Relevant indicators for risk monitoring

The relevant indicators for monitoring risk to quality of the electricity service provided to “Distribution line 3” are given in Appendix C.2.2.

C.3.3 Design and deployment of indicators for risk monitoring

Design and deployment specifications for the indicators for monitoring risk to quality of the electricity service provided to “Distribution line 3” are given in Appendix C.2.3.
C.4 Electricity service provided to Transmission line

C.4.1 Detailed threat diagrams

In Figure 47 is the detailed version of the high-level threat diagram in Figure 26 on page 57. The referring elements in Figure 47 refer to the referenced threat scenarios provided in Figures 48–52 and 57, and the referenced unwanted incident provided in Figure 58. Moreover, the referenced threat scenario in Figure 52 contains four referring threat scenarios, which refer to the referenced threat scenarios provided in Figures 53–56. Client EPP has estimated all the likelihood and consequence values in the different figures.
DL3-PTS electricity service is not delivered according to requirements

Figure 48: The referenced threat scenario “DL3-PTS electricity service is not delivered according to requirements,” referred to in Figure 47

DL2-CS electricity service is not delivered according to requirements

Figure 49: The referenced threat scenario “DL2-CS electricity service is not delivered according to requirements,” referred to in Figure 47

As can be seen in Figure 47, the five vulnerabilities in Figure 26 have been decomposed into 10 vulnerabilities. The referenced threat scenarios in Figures 48 and 49 are the detailed versions of the referring threat scenarios “DL3-PTS electricity service is not delivered according to requirements” and “DL2-CS electricity service is not delivered according to requirements” in Figure 26, respectively. Both “Distribution line 3” and “Distribution line 2” need to fulfill the availability requirement when delivering electricity to “Private telecom system” and “Control system,” respectively.

Client EPP estimates the maximum amount of electricity delivered in the period of one year to “Transmission line” to be 365 MWh. Before we can estimate the likelihoods of the DL3-PTS electricity service and the DL2-CS electricity service not being delivered according to their availability requirements, we need to calculate the worst-case service levels of the two services. These are as follows:

- **DL3-PTS electricity service (availability with respect to time):** 99.7% · 0.99 = 98.7% – The service is available 98.7% of the time for “Private telecom system.”

- **DL3-PTS electricity service (availability with respect to electricity delivered):** 10980 · 0.99 = 10870.2 kWh of electricity is delivered to “Private telecom system.”

- **DL2-CS electricity service (availability with respect to time):** 99.7% · 0.99 = 98.7% –
The service is available 98.7% of the time for “Control system.”

- DL2-CS electricity service (availability with respect to electricity delivered): $21960 \cdot 0.99 = 21740.4$ kWh of electricity is delivered to “Control system.”

The required service levels specify that:

- DL3-PTS electricity service (availability with respect to time): The service should be available 99.7% of the time for “Private telecom system.”
- DL3-PTS electricity service (availability with respect to electricity delivered): 10980 kWh of electricity should be delivered to “Private telecom system.”
- DL2-CS electricity service (availability with respect to time): The service should be available 99.7% of the time for “Control system.”
- DL2-CS electricity service (availability with respect to electricity delivered): 21960 kWh of electricity should be delivered to “Control system.”

To estimate likelihoods, we need to look at the differences between the required service levels and the worst-case service levels. The differences are as follows:

- DL3-PTS electricity service (availability with respect to time):
  
  \[(8760 \cdot 0.997) - (8760 \cdot 0.987) = 87.6\text{ hours in the period of one year.}\]

- DL3-PTS electricity service (availability with respect to electricity delivered):
  
  \[10980 - 10870.2 = 109.8\text{ kWh of electricity in the period of one year.}\]

- DL2-CS electricity service (availability with respect to time):
  
  \[(8760 \cdot 0.997) - (8760 \cdot 0.987) = 87.6\text{ hours in the period of one year.}\]

- DL2-CS electricity service (availability with respect to electricity delivered):
  
  \[21960 - 21740.4 = 219.6\text{ kWh of electricity in the period of one year.}\]

The “Private telecom system” uses an amount of about 30 kWh daily, while the “Control system” uses an amount of about 60 kWh daily. If electricity is not provided for three days to “Private telecom system” and “Control system,” then the amounts not delivered will be close to 109.8 kWh and 219.6 kWh. Moreover, the time the two services are not available will also be close to 87.6 hours. Client EPP does however not find it reasonable that electricity is not delivered for three full days to “Private telecom system” and “Control system.” Thus, Client EPP believes that the likelihood for both services should be higher than three. Based on the differences between the required service levels and the worst-case service levels, Client EPP estimates the likelihood of the DL3-PTS electricity service not being delivered according to the availability requirements to be between 5 and 10 times per year (“Likely”). Moreover, Client makes the same estimate for the DL2-CS electricity service.

The referenced threat scenarios in Figures 50 and 51 are the detailed versions of the referring threat scenarios “BPS-PTS electricity service is not delivered according to requirements” and “BPS-CS electricity service is not delivered according to requirements” in Figure 26, respectively, while the referenced threat scenario in Figure 52 is the detailed version of the referring threat scenario “PTS-LHPP control instructions service, LHPP-PTS sensor data service, PTS-CS sensor data service, and CS-PTS control instruction service are not delivered according to requirements” in Figure 26. The referenced threat scenario consists of four referring threat scenarios that refer to the referenced threat scenarios in Figures 53–56. Moreover, the referenced threat scenario in Figure 57 is the detailed version of the referring threat scenario “LHPP-TL electricity service is not delivered according to requirements” in Figure 26.
BPS-PTS electricity service is not delivered according to requirements

Lack of fuel
Old components
The Backup power system fails due to component failure [Rare]
Lack of maintenance [Very unlikely]

The tank connected to the Backup power system does not contain the amount of fuel required for providing electrical power during a disruption in the electrical power grid [Very unlikely]

BPS-PTS electricity service is not delivered according to requirements

Figure 50: The referenced threat scenario “BPS-PTS electricity service is not delivered according to requirements,” referred to in Figure 47

BPS-CS electricity service is not delivered according to requirements

Lack of fuel
Old components
The Backup power system fails due to component failure [Rare]
Lack of maintenance [Very unlikely]

The tank connected to the Backup power system does not contain the amount of fuel required for providing electrical power during a disruption in the electrical power grid [Very unlikely]

BPS-CS electricity service is not delivered according to requirements

Figure 51: The referenced threat scenario “BPS-CS electricity service is not delivered according to requirements,” referred to in Figure 47

91
Figure 52: The referenced threat scenario “PTS-LHPP control instructions service, LHPP-PTS sensor data service, PTS-CS sensor data service, and CS-PTS control instructions service are not delivered according to requirements,” referred to in Figure 47
Figure 53: The referenced threat scenario “Failure to create and/or send control instructions message,” referred to in Figure 52
Figure 54: The referenced threat scenario “Sensor data message is changed,” referred to in Figure 52
Control instructions message is changed

System failure at Private telecom system

Control instructions message is changed while at Private telecom system due to system failure [Rare]

Network failure

Control instructions message is changed during transmission from Control system to Large hydro power plant due to network failure [Very unlikely]

Error-prone network connection

Control instructions message is changed during transmission or while at Private telecom system [Very unlikely]

Insufficient protection of messages while in buffer queue

Control instructions message is changed while at Private telecom system due to system failure [Rare]

Invalid control instructions are created [Very unlikely]

Sensor data is interpreted incorrectly [Unlikely]

Lack of training

Invalid control instructions are created [Very unlikely]

Unstable sensors at Large hydro power plant results in invalid sensor data [Very unlikely]

System operator at Control system

Re-transmission of sensor data message is not requested [Rare]

Insufficient check of whether the sensor data is valid [Very unlikely]

Insufficient quality control

Sensor data is interpreted incorrectly [Unlikely]

Invalid control instructions are created [Very unlikely]

Insufficient check of whether the sensor data is valid [Very unlikely]

Figure 55: The referenced threat scenario “Control instructions message is changed,” referred to in Figure 52

Figure 56: The referenced threat scenario “Incorrect control instructions are created,” referred to in Figure 52
LHPP-TL electricity service is not delivered according to requirements

Figure 57: The referenced threat scenario “LHPP-TL electricity service is not delivered according to requirements,” referred to in Figure 47
Figure 58: The referenced unwanted incident “Incident with impact on the availability of the electricity service delivered to Transmission,” referred to in Figure 47.

Figure 58 contains the referenced unwanted incident referred to in Figure 47. For each of the unwanted incidents, Client EPP believes that the “Large hydro power plant” will be shut down for a period of one day each time one of the incidents occurs. With an average production of 1000 kWh \((\frac{365000}{365})\) of electricity each day, the consequence for all incidents with respect to the quality asset is 24 hours and 1000 kWh.

The result of the detailed analysis is three risks. Based on the risk function, defined in Equations (3)–(5) on pages 51 and 52, the maximum service levels \(\text{Maximum service level}_T\) (8760 hours) and \(\text{Maximum service level}_E\) (365 MWh per year), and the two availability requirements specified in the required service level of the electricity service, we can calculate the risk values of the three risks. The risk values are as follows:

- The risk value of “Large hydro power plant is shut down due to use of invalid control instructions from control instructions message with correct checksum” is Unacceptable since

\[
\text{Expected service level}_T = [0.9986, 0.9995]
\]

is less than

\[
\frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999]
\]

and since

\[
\text{Expected service level}_E = [0.9986, 0.9995]
\]

is less than

\[
\frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995]
\]
The risk value of “Large hydro power plant is shut down due to lack of control instructions for correcting errors” is 

\[ \text{Acceptable} \]

since

\[ \text{Expected service level}_T = [0.9997, 1] \]

is greater than

\[ \frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999] \]

and since

\[ \text{Expected service level}_E = [0.9997, 1] \]

is greater than

\[ \frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995] \]

The risk value of “Large hydro power plant is shut down due to use of invalid control instructions from control instructions message with incorrect checksum” is 

\[ \text{Acceptable} \]

since

\[ \text{Expected service level}_T = [0.9997, 1] \]

is greater than

\[ \frac{\text{Required service level}_T}{\text{Maximum service level}_T} = [0.999, 0.999] \]

and since

\[ \text{Expected service level}_E = [0.9997, 1] \]

is greater than

\[ \frac{\text{Required service level}_E}{\text{Maximum service level}_E} = [0.9995, 0.9995] \]

C.4.2 Relevant indicators for risk monitoring

Client EPP believes that the likelihood value used to calculate the risk value of the risk “Large hydro power plant is shut down due to lack of control instructions for correcting errors” may be subject to change. We therefore decide to monitor this risk.

The indicators should be used to monitor likelihood values, since the likelihood value used to calculate the risk value of the risk may be subject to change. Client EPP does not find it feasible to directly monitor the likelihood of the unwanted incident occurring, and has therefore decided to monitor the conditional likelihoods of four leads-to relations in the detailed high-level threat diagram in Figure 47 that affect the likelihood of the unwanted incident occurring. The relevant indicators for the four leads-to relations are presented in Figure 59. In Appendix D.5 we show how to use the conditional likelihoods we now address as well as other factors to monitor the resulting likelihood of the risk identified for monitoring.

One composite indicator \( c_8 \), which aggregates the two basic indicators \( b_{13} \) and \( b_{14} \), has been identified for two leads-to relations that have the same vulnerability, while another composite indicator \( c_9 \), which aggregates the three basic indicators \( b_{15}, b_{16}, \) and \( b_{17} \), has been identified for the two other leads-to relations that also have the same vulnerability. \( c_8 \) calculates the ratio of
PTS-LHPP control instructions service, LHPP-PTS sensor data service, PTS-CS sensor data service, and CS-PTS control instructions service are not delivered according to requirements.

BPS-PTS electricity service is not delivered according to requirements.

BPS-CS electricity service is not delivered according to requirements.

Lack of reserve fuel

Lack of reserve components

Lack of reserve components

Figure 59: Relevant indicators, assigned to leads-to relations in an excerpt of the detailed high-level threat diagram in Figure 47, for monitoring the risk “Large hydro power plant is shut down due to lack of control instructions for correcting errors”.
critical components for which reserve components cannot be installed if the critical components fails to all critical components, while \( c_9 \) makes a prediction about the likelihood of the “Backup power system” running out of fuel.

C.4.3 Design and deployment of indicators for risk monitoring

In Figure 59 the composite indicators \( c_8 \) and \( c_9 \) are both associated with two leads-to relations each. Conditional likelihoods were assigned to these leads-to relations during the detailed analysis described in Appendix C.4.1. We therefore obtain values for the different basic indicators from the time when the detailed high-level threat diagram in Figure 47 was constructed. For \( b_{13} \) and \( b_{14} \) we obtain the values 15 and 20, respectively, while for \( b_{15} \), \( b_{16} \), and \( b_{17} \) we obtain the values 15000, 10000, and 8, respectively.

In Tables 17 and 18 are the design specifications for the different basic and composite indicators with the exception of the basic indicators \( b_{15} \), \( b_{16} \), and \( b_{17} \). These three basic indicators are so simple that no design specifications are needed. All the specifications have been given in the form of algorithms. The three algorithms are to be used by a risk monitor within the electrical power production infrastructure. The indicators are updated every two weeks. Afterwards, the risk picture is updated based on the updated composite indicators.

The composite indicator \( c_8 \) aggregates the two indicators \( b_{13} \) and \( b_{14} \). Client EPP is of the opinion that \( c_8 \) should never be less than 0.1. Thus, if the aggregation of \( b_{13} \) and \( b_{14} \) results in a value that is less than 0.1, then \( c_8 \) is assigned the value 0.1. By using the obtained values for the basic indicators as input to the algorithm we get 0.25, which is close to the initial estimates. It should be noticed that the basic indicator \( b_{14} \) will never be equal to zero. Thus, we do not apply any check for this in the design specification in Table 17.

The composite indicator \( c_9 \) aggregates the three indicators \( b_{15} \), \( b_{16} \), and \( b_{17} \). The number 20000 used in the algorithm of \( c_9 \) is the liters of fuel needed for running the “Backup power system” during disruptions of average length in the electrical power grid. The average length is eight hours. The value 0.1 is assigned to \( c_9 \) if there is at least 20000 liters of fuel available. If less than 20000 liters of fuel is available, then the value of \( c_8 \) is determined by the delivery time of the fuel. By using the obtained values for the basic indicators as input to the algorithm we get 0.1, which is of course in accordance with the initial estimates.

In Tables 19 and 20 are the deployment specifications for the basic and composite indicators.
Table 17: Design specifications, in the form of algorithms, for the basic indicators $b_{13}$ and $b_{14}$ and the composite indicator $c_8$

<table>
<thead>
<tr>
<th>Algorithm for $b_{13}$ and $b_{14}$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $data_1$: “Data on the components of Backup power system”</td>
</tr>
<tr>
<td><strong>Data maintained by the risk monitor:</strong> $list_1$: “List of names of components in the Backup power system that are critical for the functioning of the system, and where each component has a reserve component that can be installed if it fails,” $list_2$: “List of names of components in the Backup power system that are critical for the functioning of the system”</td>
</tr>
<tr>
<td>Based on $data_1$, check whether $list_1$ should be updated. Add names of components to $list_1$, if applicable. Remove names of components from $list_1$, if applicable.</td>
</tr>
<tr>
<td>Based on $data_1$, check whether $list_2$ should be updated. Add names of components to $list_2$, if applicable. Remove names of components from $list_2$, if applicable.</td>
</tr>
</tbody>
</table>
| $b_{13} := “The number of elements in $list_1$”$  

$b_{14} := “The number of elements in $list_2”$ |
| **Output:** $b_{13}, b_{14}$ |

<table>
<thead>
<tr>
<th>Algorithm for $c_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $b_{13}$: “Number of components in the Backup power system that are critical for the functioning of the system, and where each component has a reserve component that can be installed if it fails,” $b_{14}$: “Total number of components in the Backup power system that are critical for the functioning of the system”</td>
</tr>
</tbody>
</table>
| $c_8 := 1 - \frac{b_{13}}{b_{14}}$  

$if c_8 < 0.1 then  

$c_8 := 0.1  

end if$ |
| **Output:** $c_8$ |
Table 18: Design specification, in the form of an algorithm, for the composite indicator $c_9$

<table>
<thead>
<tr>
<th>Algorithm for $c_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $b_{15}$: “Number of liters of fuel left in the tank connected to the Backup power system,” $b_{16}$: “Number of liters of fuel in reserve,” $b_{17}$: “Number of days it takes to get fuel delivered”</td>
</tr>
</tbody>
</table>
| if $b_{15} + b_{16} \geq 20000$ then 
  $c_9 := 0.1$ 
else 
  if $b_{15} + b_{16} < 20000$ and $0 < b_{17} \leq 7$ then 
    $c_9 := 0.3$ 
  else 
    if $b_{15} + b_{16} < 20000$ and $b_{17} > 7$ then 
      $c_9 := 0.5$ 
    end if 
  end if 
end if |
| **Output:** $c_9$ |

Table 19: Deployment specification for the basic indicators $b_{13}$ and $b_{14}$ and the composite indicator $c_8$

<table>
<thead>
<tr>
<th>Deployment specification for $b_{13}$, $b_{14}$, and $c_8$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of data</strong>$_1$: Client EPP has a maintenance database that contains information about different components and systems in the infrastructure, including the “Backup power system.” Every two weeks, an automated ICT process extracts data for components of the “Backup power system” that the database has been updated with in period of two weeks backwards. It should be noticed that the process will extract all the data that is available for the components the first time it is executed. The process transmits data$_1$ to the risk monitor by the use of the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>

102
Table 20: Deployment specification for the basic indicators $b_{15}$, $b_{16}$, and $b_{17}$ and the composite indicator $c_9$

<table>
<thead>
<tr>
<th>Deployment specification for $b_{15}$, $b_{16}$, $b_{17}$, and $c_9$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Extraction and transmission of $b_{15}$</strong>: A sensor is used to keep track of the number of liters of fuel left in the tank connected to the “Backup power system.” The sensor measures the number of liters left at least daily. The measurements are transmitted to the maintenance database of Client EPP. Every two weeks, an automated ICT process extracts the latest measurement from the database. The number extracted is the basic indicator $b_{15}$. The process transmits $b_{15}$ to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
<tr>
<td><strong>Extraction and transmission of $b_{16}$</strong>: A sensor is used to keep track of the number of liters of fuel left in the tank that stores the reserve fuel of the “Backup power system.” The sensor measures the number of liters left at least daily. The measurements are transmitted to the maintenance database of Client EPP. Every two weeks, an automated ICT process extracts the latest measurement from the database. The number extracted is the basic indicator $b_{16}$. The process transmits $b_{16}$ to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
<tr>
<td><strong>Extraction and transmission of $b_{17}$</strong>: Every two weeks an employee of Client EPP obtains the expected delivery time in days for fuel from the company delivering fuel to Client EPP. The number obtained is the basic indicator $b_{17}$. The employee updates the maintenance database with this number. After the database has been updated, an automated ICT process extracts $b_{17}$ and transmits it to the risk monitor by using the internal data network of the electrical power production infrastructure.</td>
</tr>
</tbody>
</table>
D Monitor risk values based on identified indicators

In this appendix we show how to monitor risk values for provided services based on the indicators identified in Section 6 and Appendix C. The risk values are monitored only indirectly via monitored likelihood and consequence values.

The rest of the appendix is structured as follows: in Appendix D.1 we present rules for calculating likelihoods, while in Appendices D.2–D.5 we show how to monitor risk values for the five provided services. Appendix D.4 covers both of the electricity services provided by “Small hydro power plant” because the monitoring of their risk values are identical.

In Appendices D.2–D.5 we present CORAS diagrams based on the CORAS diagrams presented in Section 5 and Appendices C.1.1, C.2.1, and C.4.1. The CORAS diagrams in Appendices D.2–D.5 contain only the fragments that are necessary for calculating frequencies of risks. The fragments have been assigned some additional conditional likelihoods that were left out earlier to keep things simple. We have also included a shorthand notation in the description of each threat scenario (TS) and unwanted incident (UI) to make it easier to refer to them during the likelihood calculation.

D.1 Likelihood calculation rules

To reason about likelihoods, we use the likelihood calculation rules defined in [7]. In Appendices D.2–D.5, all the likelihoods assigned to threat scenarios, unwanted incidents, or gates are given in the form of frequencies, while conditional likelihoods assigned to leads-to relations are given in the form of probabilities. Furthermore, the frequencies are given in the form of intervals with respect to a period of one year. This period for calculating risk values has been chosen since the two risk functions (defined in Equations (1) and (2) on page 32 and in Equations (3)–(5) on pages 51 and 52) use values that are given for the period of one year.

As can be seen in Table 1 on page 31, the maximum value of the likelihood scale is the frequency interval “Certain” ([50, ∞] : 1 year). Thus, a “certain” event has 50 times per year, i.e., almost one time per week, as its lower frequency threshold, while it has ∞ times per year as its upper frequency threshold. To make use of this frequency interval in a practical setting, e.g., to calculate risk values, we need to replace ∞ with a more reasonable value. Client EPP is of the opinion that the events of relevance (i.e., those that have been identified in the risk analysis) never occur more than 100 times per year. We therefore decide that ∞ should be replaced by 100 times per year, i.e., almost two times a week. Thus, in the remainder of this report, “Certain” will equal [50, 100] : 1 year.

In the likelihood calculations, we use the rules for interval arithmetic defined in Section 5.3. In addition, we use rules for determining the minimum and maximum value of two closed intervals. For two intervals [a, b] and [c, d], where both are subsets of the positive real line R+ , we use the following rules to calculate the minimum and maximum value:

- **Minimum value:** $$\min([a, b], [c, d]) = \min(a, c)$$
- **Maximum value:** $$\max([a, b], [c, d]) = \max(b, d)$$

Notice that in addition to calculating the minimum and maximum value of two intervals, we also calculate the minimum and maximum value of two positive real numbers and single intervals in this appendix. In those cases, normal rules for minimum and maximum value calculation apply.

In the following we present rules for calculating and reasoning about frequencies in CORAS diagrams. First we present two rules from [7] for calculating and reasoning about exact frequencies. Afterwards we present three rules that apply to frequency intervals. The examples in Figure

---

1See Appendix C.3 for more information.
60 are used during the presentation to explain the rules. The rules are given on the following form:

\[
\begin{array}{cccc}
P_1 & P_2 & \ldots & P_n \\
\hline
C
\end{array}
\]

We refer to \(P_1, \ldots, P_n\) as the premises and to \(C\) as the conclusion. The interpretation is that if the premises are valid, so is the conclusion.

**Rule 1 (Leads-to)** For the scenarios/incidents \(e_1\) and \(e_2\) related by the leads-to relation, we have:

\[
\frac{e_1(f) \quad e_1 \overset{\rightarrow}{\rightarrow} e_2}{(e_1 \sqcap e_2)(f \cdot l)}
\]

The leads-to rule captures the conditional likelihood semantics embedded in the leads-to relation. The frequency of the occurrences of the scenario/incident \(e_2\) that are due to the scenario/incident \(e_1\) is equal to the frequency \(f\) of \(e_1\) multiplied with the conditional likelihood \(l\) that \(e_1\) will lead to \(e_2\) given that \(e_1\) occurs. \(e_1 \sqcap e_2\) is to be understood as the subset of the scenarios/incidents \(e_2\) that are preceded by \(e_1\).

We let \(f_1\) and \(p_1\) in Figure 60 be equal to \(3 : 1\) year and \(0.3\), respectively. Recall that we only use probabilities for conditional likelihoods. We calculate the frequency of \(e_1 \sqcap e_2\) as follows:

\[
f_{e_1 \sqcap e_2} = f_1 \cdot p_1 = 3 \cdot 0.3 = 0.9
\]

The frequency of \(e_1 \sqcap e_2\) occurring is approximately 1 time per year. Notice that \(f_{e_1 \sqcap e_2}\) is equal to \(f_2\) if \(e_1\) is the only scenario that can lead to \(e_2\).

**Rule 2 (Separate scenarios/incidents)** If the scenarios/incidents \(e_1\) and \(e_2\) are separate, we have:

\[
\frac{e_1(f_1) \quad e_2(f_2)}{(e_1 \sqcup e_2)(f_1 + f_2)}
\]

Two scenarios/incidents \(e_1\) and \(e_2\) are separate if they do not overlap in content. If this is the case, then neither of the two scenarios/incidents is an instance of the other. It also means that one scenario/incident cannot be a special case of the other. For instance, if \(e_1\) is the scenario “Virus

Figure 60: Examples used for explaining the rules for calculating and reasoning about frequencies
infects computer,” while $e_2$ is the scenario “Malware infects computer,” then $e_1$ is a special case of $e_2$ and the two scenarios overlap in content.

Let us assume that $e_3 \cap e_5$ and $e_4 \cap e_5$ in Figure 60 are separate. Moreover, we let $f_3$ and $p_2$ be equal to $3 : 1$ year and $0.9$, respectively, while we let $f_4$ and $p_3$ be equal to $4 : 1$ year and $0.6$, respectively. We then calculate the frequency of $(e_3 \cap e_5) \cup (e_4 \cap e_5)$ as follows:

$$f_{(e_3 \cap e_5) \cup (e_4 \cap e_5)} = f_{e_3 \cap e_5} + f_{e_4 \cap e_5}$$

$$= (f_3 \cdot p_2) + (f_4 \cdot p_3) = (3 \cdot 0.9) + (4 \cdot 0.6)$$

$$= 2.7 + 2.4 = 5.1$$

The frequency of $(e_3 \cap e_5) \cup (e_4 \cap e_5)$ occurring is approximately 5 times per year. Notice that $f_{(e_3 \cap e_5) \cup (e_4 \cap e_5)}$ is equal to $f_5$ if $e_3$ and $e_4$ are the only scenarios that can lead to $e_5$.

**Rule 3 (Leads-to – frequency interval)** For the scenarios/incidents $e_1$ and $e_2$ related by the leads-to relation, we have:

$$e_1([f_a, f_b]) \xrightarrow{e_1 \rightarrow [l_a, l_b]} e_2$$

$$(e_1 \cap e_2)([f_a \cdot l_a, f_b \cdot l_b])$$

The rule above is a generalization of Rule 1 to frequency intervals.

We let $f_1$ and $p_1$ in Figure 60 be equal to $[0.6, 1.9] : 1$ year (“Unlikely”) and $[0.1, 0.3]$, respectively. Moreover, we let $f_{e_1 \cap e_2}$ be the frequency of $e_1 \cap e_2$. We then have:

$$f_{e_1 \cap e_2} = f_1 \cdot p_1 = [0.6, 1.9] \cdot [0.1, 0.3] = [0.06, 0.57]$$

The frequency of $e_1 \cap e_2$ is $[0.06, 0.57] : 1$ year.

**Rule 4 (Separate scenarios/incidents – frequency interval)** If the scenarios/incidents $e_1$ and $e_2$ are separate, we have:

$$e_1([f_a, f_b]) \xrightarrow{(e_1 \cap e_2)([f_a + f_c, f_b + f_d])} e_2([f_c, f_d])$$

The rule above is a generalization of Rule 2 to frequency intervals.

Let us assume that $e_3 \cap e_5$ and $e_4 \cap e_5$ in Figure 60 are separate. Moreover, we let $f_3$ and $p_2$ be equal to $[0, 0.1] : 1$ year (“Rare”) and $0.9$, respectively, while we let $f_4$ and $p_3$ be equal to $[0.2, 0.5] : 1$ year (“Very unlikely”) and $0.6$, respectively. We then calculate the frequency of $(e_3 \cap e_5) \cup (e_4 \cap e_5)$ as follows:

$$f_{(e_3 \cap e_5) \cup (e_4 \cap e_5)} = f_{e_3 \cap e_5} + f_{e_4 \cap e_5}$$

$$= (f_3 \cdot p_2) + (f_4 \cdot p_3) = ([0, 0.1] \cdot 0.9) + ([0.2, 0.5] \cdot 0.6)$$

$$= [0, 0.09] + [0.12, 0.3] = [0.12, 0.39]$$

The frequency of $(e_3 \cap e_5) \cup (e_4 \cap e_5)$ is $[0.12, 0.39] : 1$ year.

**Rule 5 (General – frequency interval)** For two scenarios/incidents $e_1$ and $e_2$, we have:

$$e_1([f_a, f_b]) \xrightarrow{(e_1 \cap e_2)([max(f_a, f_c), f_b + f_d])} e_2([f_c, f_d])$$
The rule above is the general rule for calculating with frequency intervals. We can for instance use this rule if two scenarios/incidents are not separate.

Let's assume that $e_3 \sqcap e_5$ and $e_4 \sqcap e_5$ in Figure 60 are not separate. Moreover, we let $f_3$ and $p_2$ be equal to $[5, 9.9]$ : 1 year (“Likely”) and 0.4, respectively, while we let $f_4$ and $p_3$ be equal to $[2, 4.9]$ : 1 year (“Possible”) and 0.7, respectively. We first calculate the frequencies of $e_3 \sqcap e_5$ and $e_4 \sqcap e_5$ as follows:

\[
\begin{align*}
    f_{e_3 \sqcap e_5} &= f_3 \cdot p_2 = [5, 9.9] \cdot 0.4 = [2, 3.96] \\
    f_{e_4 \sqcap e_5} &= f_4 \cdot p_3 = [2, 4.9] \cdot 0.7 = [1.4, 3.43]
\end{align*}
\]

To calculate the minimum frequency value $f_{\text{min}}$ and the maximum frequency value $f_{\text{max}}$ of $(e_3 \sqcap e_5) \cup (e_4 \sqcap e_5)$ we do as follows:

\[
\begin{align*}
    f_{\text{min}} &= \max(\min(f_{e_3 \sqcap e_5}), \min(f_{e_4 \sqcap e_5})) \\
                 &= \max(\min([2, 3.96]), \min([1.4, 3.43])) = \max(2, 1.4) = 2 \\
    f_{\text{max}} &= \max(f_{e_3 \sqcap e_5}) + \max(f_{e_4 \sqcap e_5}) \\
                 &= \max([2, 3.96]) + \max([1.4, 3.43]) = 3.96 + 3.43 = 7.39
\end{align*}
\]

The frequency of $(e_3 \sqcap e_5) \cup (e_4 \sqcap e_5)$ is $[2, 7.39]$ : 1 year.

D.2 Sensor data service provided to Public telecom system

The risk values of the risks “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus” and “Sensor data is sent in plain text from Small hydro power plant to an outsider” are indirectly monitored based on two conditional likelihoods. The two likelihoods are monitored by the use of the composite indicators $c_1$ and $c_2$ in Figure 21 on page 43. The two conditional likelihoods are assigned to two leads-to relations. The two likelihoods can be used to calculate frequencies assigned to vertices that the two leads-to relations lead up to, including the frequencies of the two risks.

In Figures 61 and 62, we have replaced the frequencies to be calculated, based on the composite indicators, with variables. The two figures contain CORAS diagrams that are based on different CORAS diagrams in Section 5. We assume that the CORAS diagrams in the two figures are based on complete CORAS diagrams. This means that no other threat scenarios or unwanted incidents than the ones specified in Figures 61 and 62 can lead to other threat scenarios, unwanted incidents, or out-gates. Based on this assumption we can for instance state that $f_2 = f_{TS1 \sqcap TS2}$, where $f_{TS1 \sqcap TS2}$ is the frequency of $TS1 \sqcap TS2$. In other words, $TS1$ is the only threat scenario that can lead to $TS2$.

As can be seen in the figures, some of the variables have been used for a number of frequencies. The same variable is assigned to all frequencies that should be equal. For instance, the out-gate $o_{19}$ in Figure 62 has been assigned the frequency variable $f_1$ of the threat scenario “Small hydro power plant’s sensors are infected with a computer virus,” since the threat scenario leads to the out-gate with the conditional likelihood 1.0, and since $f_1 \cdot 1.0 = f_1$ (Rule 3).

To monitor the risk value of the risk “Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus,” we start by calculating the frequency $f_1$ of the threat scenario $TS3$ in Figure 62. We calculate this frequency as follows:

\[
\begin{align*}
    f_1 &= f_{TS1 \sqcap TS3} \\
         &= [0.6, 1.9] \cdot c_1 \text{ (Rule 3)} \\
         &= [0.6 \cdot c_1, 1.9 \cdot c_1]
\end{align*}
\]
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

UI1: Sensor data is sent in plain text from Small hydro power plant to an outsider

UI2: Incorrect sensor data is sent to Public telecom system due to sensors being infected with a computer virus

Confidentiality of sensor data delivered to Public telecom system

Integrity of sensor data delivered to Public telecom system

Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Figure 61: CORAS diagram based on CORAS diagrams in Figures 14, 16, and 20 on pages 33, 36, and 39, respectively
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

TS1: The outsider sends an encrypted control instructions message to Small hydro power plant that contains malicious code

[Unlikely]

Trojan horse protection not up to date

Virus protection not up to date

TS2: Spyware is introduced at the Small hydro power plant by the use of a Trojan horse

c1(b1,b2)

TS3: Small hydro power plant’s sensors are infected with a computer virus

[f1]

TS4: The spyware can send data to an outsider without being detected

[f5]

0.5

Lack of monitoring of outgoing data connections

Figure 62: CORAS diagram based on the CORAS diagram in Figure 19 on page 38

Notice that the frequency “Unlikely” is given for the period of one year, and that Rule 3 is used to calculate f1. Based on this frequency, we can calculate the frequency f5 of the unwanted incident U12 in Figure 61 occurring. We calculate the frequency as follows:

\[
f_5 = f_{TS3} \cap U_{12} = f_1 \cdot 0.6 \text{ (Rule 3)}
\]

\[
= [0.6 \cdot c_1, 1.9 \cdot c_1] \cdot 0.6 = [0.36 \cdot c_1, 1.14 \cdot c_1]
\]

We continue by calculating the Expected service level. We calculate it as follows:

\[
\text{Expected service level} = \frac{\text{Maximum service level} - (\text{Likelihood} \cdot \text{Consequence})}{\text{Maximum service level}}
\]

\[
= \frac{5000 - (f_5 \cdot 96)}{5000}
\]

\[
= \frac{5000 - ([0.36 \cdot c_1, 1.14 \cdot c_1] \cdot 96)}{5000}
\]

\[
= \frac{5000 - [34.56 \cdot c_1, 109.44 \cdot c_1]}{5000}
\]

The final step is to define how Risk Value should be calculated. We calculate it as follows:

\[
\text{if } \frac{5000 - [34.56 \cdot c_1, 109.44 \cdot c_1]}{5000} \geq \frac{5000 \cdot 0.999}{5000} \text{ then}
\]

\[
\text{Risk value} = \text{Acceptable}
\]

\[
\text{else}
\]

\[
\text{Risk value} = \text{Unacceptable}
\]

\[
\text{endif}
\]

To monitor the risk value of the risk “Sensor data is sent in plain text from Small hydro power plant to an outsider,” we start by calculating the frequency f2 of the threat scenario TS2
in Figure 62. We calculate this frequency as follows:

\[ f_2 = f_{TS1} \cap TS2 \]
\[ = [0.6, 1.9] \cdot c_2 \text{ (Rule 3)} \]
\[ = [0.6 \cdot c_2, 1.9 \cdot c_2] \]

The threat scenario TS2 leads to TS4 in the same figure. We calculate the frequency of TS4 as follows:

\[ f_3 = f_{TS2} \cap TS4 \]
\[ = f_2 \cdot 0.5 \text{ (Rule 3)} \]
\[ = [0.6 \cdot c_2, 1.9 \cdot c_2] \cdot 0.5 = [0.3 \cdot c_2, 0.95 \cdot c_2] \]

Based on this frequency, we can calculate the frequency \( f_4 \) of the unwanted incident UI1 in Figure 61 occurring. We calculate the frequency as follows:

\[ f_4 = f_{TS4} \cap UI1 \]
\[ = f_3 \cdot 0.7 \text{ (Rule 3)} \]
\[ = [0.3 \cdot c_2, 0.95 \cdot c_2] \cdot 0.7 = [0.21 \cdot c_2, 0.665 \cdot c_2] \]

We continue by calculating the \textit{Expected service level}. We calculate it as follows:

\[
\text{Expected service level} = \frac{\text{Maximum service level} - (\text{Likelihood} \cdot \text{Consequence})}{\text{Maximum service level}}
\]
\[ = \frac{5000 - (f_4 \cdot 96)}{5000} \]
\[ = \frac{5000 - ([0.21 \cdot c_2, 0.665 \cdot c_2] \cdot 96)}{5000} \]
\[ = \frac{5000 - [20.16 \cdot c_2, 63.84 \cdot c_2]}{5000} \]

The final step is to define how \textit{Risk Value} should be calculated. We calculate it as follows:

\[
\text{if } \frac{5000 - [20.16 \cdot c_2, 63.84 \cdot c_2]}{5000} \geq \frac{5000 \cdot 0.995}{5000} \text{ then } \text{Risk value} = \text{Acceptable} \\
\text{else } \text{Risk value} = \text{Unacceptable} \\
\text{endif}
\]

D.3 Control instructions service provided to Public telecom system

The risk value of the risk \textit{“No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data”} is indirectly monitored based on three conditional likelihoods. The three likelihoods are monitored by the use of the composite indicators \( c_3 \) and \( c_4 \) in Figures 34 and 35 on pages 65 and 66, respectively. The three conditional likelihoods are assigned to three leads-to relations. The three likelihoods can be used to calculate frequencies assigned to vertices that the three leads-to relations lead up to, including the frequency of the risk.
UI1: No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data

Sensor data message is not delivered to Home office computer

Sensor data message with lack of integrity is delivered to Home office computer

Figure 63: CORAS diagram based on CORAS diagrams in Figures 27, 29, and 33 on pages 58, 60, and 64, respectively
Sensor data message is not delivered to Home office computer

Figure 64: CORAS diagram based on the CORAS diagram in Figure 30 on page 61

Sensor data message with lack of integrity is delivered to Home office computer

Figure 65: CORAS diagram based on the CORAS diagram in Figure 31 on page 62
In Figures 63–65, we have replaced the frequencies to be calculated, based on the composite indicators, with variables. The three figures contain CORAS diagrams that are based on different CORAS diagrams in Appendix C.1.1. As in Appendix D.2, we assume that the CORAS diagrams in the figures are based on complete CORAS diagrams.

To monitor the risk value of the risk “No control instructions message is sent to Public telecom system due to lack of sensor data or use of invalid sensor data,” we need to define how the different frequencies, represented by the variables, should be calculated. We start by showing how this is done for the threat scenarios in Figure 65. As can be seen in Figure 65, the frequency of the threat scenario $TS_2$, i.e., $f_2$, depends on the frequency of $TS_8$, i.e., $f_8$. This frequency depends again on the frequency of $TS_7$, i.e., $f_7$, which again depends on the frequency of $TS_{10}$, i.e., “Certain.” We therefore start by calculating $f_7$. We calculate $f_7$, $f_8$, and $f_2$ as follows:

\[
\begin{align*}
  f_7 &= f_{TS_{10}} \cap nolimits \cap TS_7 \\
  &= [50, 100] \cdot c_3 \, (\text{Rule 3}) \\
  &= [50 \cdot c_3, 100 \cdot c_3] \\
  f_8 &= f_{TS_7} \cap nolimits \cap TS_8 \\
  &= f_7 \cdot 0.001 \, (\text{Rule 3}) \\
  &= [50 \cdot c_3, 100 \cdot c_3] \cdot 0.001 = [0.05 \cdot c_3, 0.1 \cdot c_3] \\
  f_2 &= f_{TS_8} \cap nolimits \cap TS_2 \\
  &= f_8 \cdot 0.5 \, (\text{Rule 3}) \\
  &= [0.05 \cdot c_3, 0.1 \cdot c_3] \cdot 0.5 = [0.025 \cdot c_3, 0.05 \cdot c_3]
\end{align*}
\]

The next step is to calculate the different frequencies of the threat scenarios in Figure 64. Here, $f_3$ depends on the frequency $f_2$ and the conditional likelihood $0.8$ given in Figure 63, while $f_4$ depends on $f_3$. Moreover, $f_5$ depends on $f_4$. We therefore start by calculating $f_3$, before we calculate $f_4$ and $f_5$. We calculate the frequencies as follows:

\[
\begin{align*}
  f_3 &= f_{TS_2} \cap nolimits \cap TS_3 \\
  &= f_2 \cdot 0.8 \, (\text{Rule 3}) \\
  &= [0.025 \cdot c_3, 0.05 \cdot c_3] \cdot 0.8 = [0.02 \cdot c_3, 0.04 \cdot c_3] \\
  f_4 &= f_{TS_3} \cap nolimits \cap TS_4 \\
  &= f_3 \cdot 0.9 \, (\text{Rule 3}) \\
  &= [0.02 \cdot c_3, 0.04 \cdot c_3] \cdot 0.9 = [0.018 \cdot c_3, 0.036 \cdot c_3] \\
  f_5 &= f_{TS_4} \cap nolimits \cap TS_5 \\
  &= f_4 \cdot c_4 \, (\text{Rule 3}) \\
  &= [0.018 \cdot c_3, 0.036 \cdot c_3] \cdot c_4 = [0.018 \cdot c_3 \cdot c_4, 0.036 \cdot c_3 \cdot c_4]
\end{align*}
\]

As can be seen in Figure 64, the frequency of the threat scenario $TS_1$ depends on the frequencies of $TS_5 \cap nolimits \cap TS_1$ and $TS_6 \cap nolimits \cap TS_1$. We use Rule 4 to calculate the frequency $f_1$, since $TS_5 \cap nolimits \cap TS_1$ and $TS_6 \cap nolimits \cap TS_1$ are separate. Before we calculate $f_1$, we calculate the frequencies of $TS_9 \cap nolimits \cap TS_6$, i.e., $f_6$, $TS_5 \cap nolimits \cap TS_1$, and $TS_6 \cap nolimits \cap TS_1$. We calculate the frequencies as
follows:

\[
\begin{align*}
  f_6 &= f_{TS6} \sqcap TS_6 \\
  &= [50, 100] \cdot c_3 \quad \text{(Rule 3)} \\
  &= [50 \cdot c_3, 100 \cdot c_3] \\
  f_{TS5} \sqcap TS_1 &= f_5 \cdot 0.9 \quad \text{(Rule 3)} \\
  &= [0.018 \cdot c_3 \cdot c_4, 0.036 \cdot c_3 \cdot c_4] \cdot 0.9 \\
  &= [0.0162 \cdot c_3 \cdot c_4, 0.0324 \cdot c_3 \cdot c_4] \\
  f_{TS6} \sqcap TS_1 &= f_6 \cdot 0.9 \quad \text{(Rule 3)} \\
  &= [50 \cdot c_3, 100 \cdot c_3] \cdot 0.9 \\
  &= [45 \cdot c_3, 90 \cdot c_3] \\
  f_1 &= f_{TS5} \sqcap TS_1 \sqcup (TS_6 \sqcap TS_1) \\
  &= [\min f_1, \max f_1] \quad \text{(Rule 4)} \\
  \min f_1 &= \min(f_{TS5} \sqcap TS_1) + \min(f_{TS6} \sqcap TS_1) \\
  &= (0.0162 \cdot c_3 \cdot c_4) + (45 \cdot c_3) \\
  \max f_1 &= \max(f_{TS5} \sqcap TS_1) + \max(f_{TS6} \sqcap TS_1) \\
  &= (0.0324 \cdot c_3 \cdot c_4) + (90 \cdot c_3)
\end{align*}
\]

As can be seen in Figure 63, \( f_1 \) is also the frequency of \( TS_1 \sqcap UI_1 \). In other words, \( f_1 \) is the frequency of the unwanted incident \( UI_1 \) occurring. We continue by calculating the Expected service level. We calculate it as follows:

\[
\text{Expected service level} = \frac{\text{Maximum service level} - (\text{Likelihood} \cdot \text{Consequence})}{\text{Max service level}} \\
= \frac{1000 - (f_1 \cdot 1)}{1000} = \frac{1000 - f_1}{1000}
\]

The final step is to define how Risk Value should be calculated. We calculate it as follows:

\[
\begin{align*}
  \text{if} \quad \frac{1000 - f_1}{1000} &\geq \frac{1000 \cdot 0.9999}{1000} \quad \text{then} \\
  \text{Risk value} &= \text{Acceptable} \\
  \text{else} \\
  \text{Risk value} &= \text{Unacceptable}
\end{align*}
\]

D.4 Electricity services provided to Distribution line 2 and Distribution line 3

The unwanted incident “Small hydro power plant is shut down due to damage to unstable power generator” is given in both Figures 42 and 46 on pages 77 and 87, respectively. It impacts the two assets “Availability of electricity delivered to Distribution line 2” and “Availability of electricity delivered to Distribution line 3” with the same consequence if it occurs. Thus, the risk values for the two risks are identical. In the following we focus on the monitoring of the risk that impacts “Availability of electricity delivered to Distribution line 2.” The monitoring of the other risk is identical.

The risk value of the risk “Small hydro power plant is shut down due to damage to unstable power generator” is indirectly monitored based on one conditional likelihood and one consequence. The conditional likelihood and the consequence are monitored by the use of the
composite indicators $c_5$, $c_6$, and $c_7$ in Figures 43 and 44 on pages 81 and 82, respectively. The conditional likelihood is assigned to a leads-to relation. It can be used to calculate frequencies assigned to vertices that the leads-to relation leads up to, including the frequency of the risk.

In Figures 66 and 67, we have replaced the frequencies to be calculated, based on the composite indicator, with variables. The two figures contain CORAS diagrams that are based on different CORAS diagrams in Appendix C.2.1. As in Appendix D.2, we assume that the CORAS diagrams in the figures are based on complete CORAS diagrams.

To monitor the risk value of the risk “Small hydro power plant is shut down due to damage to unstable power generator,” we start by calculating the frequency $f_1$ of the threat scenario $TS_2$ in Figure 67. We calculate this frequency as follows:

\[
f_1 = f_{TS_1} \cap TS_2 = [0.6, 1.9] \cdot c_5 \quad \text{(Rule 3)}
\]

\[
= [0.6 \cdot c_5, 1.9 \cdot c_5]
\]

The next step is to calculate the frequency of the unwanted incident $UI_1$ occurring. We calculate
Control instructions message that do not comply with the data confidentiality policy is sent to Small hydro power plant

Figure 67: CORAS diagram based on the CORAS diagram in Figure 41 on page 76

this frequency as follows:

\[ f_2 = f_{TS2} \cap UI_{1} \]
\[ = f_1 \cdot 0.2 \text{ (Rule 3)} \]
\[ = [0.6 \cdot c_5, 1.9 \cdot c_5] \cdot 0.2 = [0.12 \cdot c_5, 0.38 \cdot c_5] \]

We can now calculate Expected service level \( T \) and Expected service level \( E \). We calculate these as follows:

\[ \text{Expected service level}_T = \frac{\text{Maximum service level}_T - (\text{Likelihood} \cdot \text{Consequence}_T)}{\text{Maximum service level}_T} \]
\[ = \frac{8760 - (0.12 \cdot c_5 \cdot 0.38 \cdot c_5 \cdot c_6)}{8760} \]
\[ = \frac{8760 - [0.12 \cdot c_5 \cdot c_6, 0.38 \cdot c_5 \cdot c_6]}{8760} \]

\[ \text{Expected service level}_E = \frac{\text{Maximum service level}_E - (\text{Likelihood} \cdot \text{Consequence}_E)}{\text{Maximum service level}_E} \]
\[ = \frac{37000 - (0.12 \cdot c_5 \cdot c_7)}{37000} \]
\[ = \frac{37000 - [0.12 \cdot c_5 \cdot c_7, 0.38 \cdot c_5 \cdot c_7]}{37000} \]

The composite indicators \( c_6 \) (Consequence\(_T\)) and \( c_7 \) (Consequence\(_E\)) represent the consequence of the unwanted incident \( UI_{1} \) occurring. The final step is to define how Risk Value should be
calculated. We calculate Risk Value as follows:

\[
\text{if } \frac{8760 - [0.12 \cdot c_5 \cdot c_6, 0.38 \cdot c_5 \cdot c_6]}{8760} \geq \frac{8760 \cdot 0.999}{8760} \text{ and } \frac{37000 - [0.12 \cdot c_5 \cdot c_7, 0.38 \cdot c_5 \cdot c_7]}{37000} \geq \frac{36980}{37000} \text{ then }
\]

\[\text{Risk value} = \text{Acceptable}\]

else

\[\text{Risk value} = \text{Unacceptable}\]

endif

D.5 Electricity service provided to Transmission line

The risk value of the risk “Large hydro power plant is shut down due to lack of control instructions for correcting errors” is indirectly monitored based on four conditional likelihoods. The four likelihoods are monitored by the use of the composite indicators \( c_8 \) and \( c_9 \) in Figure 59 on page 99. The four conditional likelihoods are assigned to four leads-to relations. The four likelihoods can be used to calculate frequencies assigned to vertices that the four leads-to relations lead up to, including the frequency of the risk.

In Figures 68–72, we have replaced the frequencies to be calculated, based on the composite indicators, with variables. The five figures contain CORAS diagrams that are based on different CORAS diagrams in Appendix C.4.1. As in Appendix D.2, we assume that the CORAS diagrams in the figures are based on complete CORAS diagrams. Notice that we let both of the referring threat scenarios “BPS-PTS electricity service is not delivered according to requirements” and “BPS-CS electricity service is not delivered according to requirements” in Figure 68 refer to the referenced threat scenario in Figure 69. We have done this, since the referenced threat scenario is based on the (almost) identical referenced threat scenarios in Figures 50 and 51 on page 91.

To monitor the risk value of the risk “Large hydro power plant is shut down due to lack of control instructions for correcting errors,” we need to define how the different frequencies, represented by the variables, should be calculated. We start by showing how this is done for the threat scenarios in Figure 71. The frequency \( f_6 \) of the threat scenario \( TS_5 \) depends on the frequencies of \( TS_1 \sqcap \sqsupseteq TS_5 \), \( TS_2 \sqcap \sqsupseteq TS_5 \), and \( TS_3 \sqcap \sqsupseteq TS_5 \). The threat scenarios \( TS_1 \) and \( TS_2 \) are given in Figure 69, while the threat scenario \( TS_3 \) is given in Figure 70. The frequencies of \( TS_1 \sqcap \sqsupseteq TS_5 \) and \( TS_2 \sqcap \sqsupseteq TS_5 \) are equal to \( f_1 \) and \( f_2 \), respectively, while the frequency of \( TS_3 \sqcap \sqsupseteq TS_5 \) equals the frequency of \( TS_3 \), i.e., “Likely,” since \( TS_3 \) and \( TS_5 \) are connected by a path of leads-to relations where each leads-to relations has the conditional likelihood 1.0. To calculate the frequencies of \( TS_1 \sqcap \sqsupseteq TS_5 \) and \( TS_2 \sqcap \sqsupseteq TS_5 \), we do as follows:

\[
\begin{align*}
\hat{f}_{TS_1 \sqcap \sqsupseteq TS_5} &= f_1 \\
&= [0.2, 0.5] \cdot c_9 \quad \text{(Rule 3)} \\
&= [0.2 \cdot c_9, 0.5 \cdot c_9] \\
\hat{f}_{TS_2 \sqcap \sqsupseteq TS_5} &= f_2 \\
&= [0, 0.1] \cdot c_8 \quad \text{(Rule 3)} \\
&= [0, 0.1 \cdot c_8]
\end{align*}
\]

We use the general rule (Rule 5) to calculate the frequency \( f_6 \), since we do not know whether
Figure 68: CORAS diagram based on CORAS diagrams in Figures 47, 52, and 58 on pages 88, 92, and 97, respectively
BPS-PTS electricity service is not delivered according to requirements / BPS-CS electricity service is not delivered according to requirements

TS2: The Backup power system fails due to component failure [Rare]

TS1: The tank connected to the Backup power system does not contain the amount of fuel required for providing electrical power during a disruption in the electrical power grid [Very unlikely]

[Very unlikely] \( o_2 / o_4 \)

1.0

[Very unlikely] \( o_3 / o_5 \) [Rare]

Figure 69: CORAS diagram based on CORAS diagrams in Figures 50 and 51 on page 91

DL3-PTS electricity service is not delivered according to requirements

TS3: DL3-PTS electricity service is not delivered by Distribution line 3 according to the availability requirements that Distribution line 3 is required to fulfill [Likely]

[Likely] \( o_1 \)

1.0

DL3-PTS electricity service is not delivered according to requirements

Figure 70: CORAS diagrams based on CORAS diagrams in Figures 48 and 49 on page 89

DL2-CS electricity service is not delivered according to requirements

TS4: DL2-CS electricity service is not delivered by Distribution line 2 according to the availability requirements that Distribution line 2 is required to fulfill [Likely]

[Likely] \( o_6 \)

1.0
Figure 71: CORAS diagram based on the CORAS diagram in Figure 53 on page 93
LHPP-TL electricity service is not delivered according to requirements

Figure 72: CORAS diagram based on the CORAS diagram in Figure 57 on page 96

TS1 ⊑ TS5, TS2 ⊑ TS5, and TS3 ⊑ TS5 are separate. We calculate $f_6$ as follows:

$$f_6 = f_{TS1 ⊑ TS5} \cup (TS2 ⊑ TS5) \cup (TS3 ⊑ TS5)$$

$$= \min f_6, \max f_6$$ (Rule 5) where

$$\min f_6 = \max(\min(f_{TS1 ⊑ TS5}), \min(f_{TS2 ⊑ TS5}), \min(f_{TS3 ⊑ TS5})) = \max(\min([0.2 \cdot c_9, 0.5 \cdot c_9]), \min([0, 0.1 \cdot c_8]), \min([5, 9.9])) = \max(0.2 \cdot c_9, 0.5)$$

$$\max f_6 = \max(f_{TS1 ⊑ TS5}) + \max(f_{TS2 ⊑ TS5}) + \max(f_{TS3 ⊑ TS5}) = \max([0.2 \cdot c_9, 0.5 \cdot c_9]) + \max([0, 0.1 \cdot c_8]) + \max([5, 9.9]) = (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9$$

The frequency interval calculated for TS5 can also be used as a frequency for TS7 since $f_{TS1 ⊑ TS5}$ equals $f_{TS1 ⊑ TS7}$, $f_{TS2 ⊑ TS5}$ equals $f_{TS2 ⊑ TS7}$, and $f_{TS3 ⊑ TS5}$ equals $f_{TS3 ⊑ TS7}$. In addition, the frequency of TS7 needs to be calculated by the use of Rule 5, since we do not know whether $TS1 ⊑ TS7$, $TS2 ⊑ TS7$, and $TS4 ⊑ TS7$ are separate. We end with the following value for $f_8$:

$$f_8 = f_{TS1 ⊑ TS7} \cup (TS2 ⊑ TS7) \cup (TS4 ⊑ TS7)$$

$$= \min f_8, \max f_8$$ (Rule 5) where

$$\min f_8 = \max(0.2 \cdot c_9, 5) = 5$$

$$\max f_8 = (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9$$

We continue by calculating the frequency of TS10, i.e., $f_{10}$. Before we can calculate this frequency, we need to calculate the frequencies of TS5 ⊑ TS8, i.e., $f_9$, TS7 ⊑ TS6, i.e., $f_7$, etc.
We calculate these as follows:

\[
\begin{align*}
\text{Rule 3:} & \quad f_9 = f_{TS5 \cap TS8} = f_5 \cdot 0.7 \\
& = [\max(0.2 \cdot c_9, 5), (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9] \cdot 0.7 \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.7, (0.35 \cdot c_9) + (0.07 \cdot c_8) + 6.93]
\end{align*}
\]

\[
\begin{align*}
\text{Rule 3:} & \quad f_{TS8 \cap TS10} = f_9 \cdot [0.2, 0.5] \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.7, (0.35 \cdot c_9) + (0.07 \cdot c_8) + 6.93] \cdot [0.2, 0.5] \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.14, (0.175 \cdot c_9) + (0.035 \cdot c_8) + 3.465]
\end{align*}
\]

\[
\begin{align*}
\text{Rule 3:} & \quad f_7 = f_{TS7 \cap TS6} = f_8 \cdot 0.5 \\
& = [\max(0.2 \cdot c_9, 5), (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9] \cdot 0.5 \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.5, (0.25 \cdot c_9) + (0.05 \cdot c_8) + 4.95]
\end{align*}
\]

\[
\begin{align*}
\text{Rule 3:} & \quad f_{TS6 \cap TS10} = f_7 \cdot [0.2, 0.5] \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.5, (0.25 \cdot c_9) + (0.05 \cdot c_8) + 4.95] \cdot [0.2, 0.5] \\
& = [\max(0.2 \cdot c_9, 5) \cdot 0.1, (0.125 \cdot c_9) + (0.025 \cdot c_8) + 2.475]
\end{align*}
\]

We use Rule 4 to calculate the frequency of \((TS8 \cap TS10) \cup (TS6 \cap TS10)\), i.e., \(f_{10}\), since \(TS8 \cap TS10\) and \(TS6 \cap TS10\) are separate. We calculate \(f_{10}\) as follows:

\[
\begin{align*}
\text{Rule 4:} & \quad f_{10} = f_{TS8 \cap TS10} \cup (TS6 \cap TS10) \\
& = [\min_{f_{10}}, \max_{f_{10}}] \\
\text{Rule 4:} & \quad \min_{f_{10}} = \min(f_{TS8 \cap TS10}) + \min(f_{TS6 \cap TS10}) \\
& = (\max(0.2 \cdot c_9, 5) \cdot 0.14) + (\max(0.2 \cdot c_9, 5) \cdot 0.1) = \max(0.2 \cdot c_9, 5) \cdot 0.24 \\
\text{Rule 4:} & \quad \max_{f_{10}} = \max(f_{TS8 \cap TS10}) + \max(f_{TS6 \cap TS10}) \\
& = ((0.175 \cdot c_9) + (0.035 \cdot c_8) + 3.465) + ((0.125 \cdot c_9) + (0.025 \cdot c_8) + 2.475) \\
& = (3 \cdot c_9) + (0.06 \cdot c_8) + 5.94
\end{align*}
\]

We continue by calculating the frequency of \(TS11\), i.e., \(f_{11}\). This frequency is based on the frequencies of \(TS10 \cap TS11\) and \(TS9 \cap TS11\). The events \(TS10 \cap TS11\) and \(TS9 \cap TS11\) are separate. Thus, we use Rule 4 to calculate \(f_{11}\). We calculate the frequencies of \(TS10 \cap TS11, TS9 \cap TS11,\) and \((TS10 \cap TS11) \cup (TS9 \cap TS11)\), i.e., \(f_{11}\), as follows:

\[
\begin{align*}
\text{Rule 3:} & \quad f_{TS10 \cap TS11} = f_{10} \cdot 1.0 \\
& = f_{10} \\
\text{Rule 3:} & \quad f_{TS9 \cap TS11} = [0.2, 0.5] \cdot 0.5 \\
& = [0.1, 0.25] \\
\text{Rule 4:} & \quad f_{11} = f_{TS10 \cap TS11} \cup (TS9 \cap TS11) \\
& = [\min_{f_{11}}, \max_{f_{11}}] \\
\text{Rule 4:} & \quad \min_{f_{11}} = \min(f_{TS10 \cap TS11}) + \min(f_{TS9 \cap TS11}) \\
& = (\max(0.2 \cdot c_9, 5) \cdot 0.24) + 0.1 \\
\text{Rule 4:} & \quad \max_{f_{11}} = \max(f_{TS10 \cap TS11}) + \max(f_{TS9 \cap TS11}) \\
& = ((3 \cdot c_9) + (0.06 \cdot c_8) + 5.94) + 0.25 \\
& = (3 \cdot c_9) + (0.06 \cdot c_8) + 6.19
\end{align*}
\]
We continue by calculating the frequency of $TS_{14}$, i.e., $f_3$. This frequency is based on the frequencies of $TS_{12} \sqcap TS_{14}$, $TS_{11} \sqcap TS_{14}$, and $TS_{13} \sqcap TS_{14}$. These frequencies belong to events that are separate. Thus, we use Rule 4 to calculate $f_3$. In order to calculate the frequencies of $TS_{12} \sqcap TS_{14}$ and $TS_{13} \sqcap TS_{14}$, we first need to calculate the frequencies of $TS_{5} \sqcap TS_{12}$, i.e., $f_{12}$, and $TS_{7} \sqcap TS_{13}$, i.e., $f_{13}$. In the following, we first calculate $f_{12}$ and $f_{13}$, before we calculate $f_3$ based on the frequencies of $TS_{12} \sqcap TS_{14}$, $TS_{11} \sqcap TS_{14}$, and $TS_{13} \sqcap TS_{14}$.

$$f_{12} = f_{TS_{5} \sqcap TS_{12}}$$
$$f_{12} = f_6 \cdot 0.2 \text{ (Rule 3)}$$
$$f_{12} = [\max(0.2 \cdot c_9, 5), (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9] \cdot 0.2$$
$$f_{12} = [\max(0.2 \cdot c_9, 5) \cdot 0.2, (0.1 \cdot c_9) + (0.02 \cdot c_8) + 1.98]$$

$$f_{13} = f_{TS_{7} \sqcap TS_{13}}$$
$$f_{13} = f_8 \cdot 0.1 \text{ (Rule 3)}$$
$$f_{13} = [\max(0.2 \cdot c_9, 5), (0.5 \cdot c_9) + (0.1 \cdot c_8) + 9.9] \cdot 0.1$$
$$f_{13} = [\max(0.2 \cdot c_9, 5) \cdot 0.1, (0.05 \cdot c_9) + (0.01 \cdot c_8) + 0.99]$$

$$f_{TS_{12} \sqcap TS_{14}} = f_{12} \cdot 1.0 \text{ (Rule 3)}$$
$$f_{12} = f_{12}$$

$$f_{TS_{11} \sqcap TS_{14}} = f_{11} \cdot 1.0 \text{ (Rule 3)}$$
$$f_{11} = f_{11}$$

$$f_{TS_{13} \sqcap TS_{14}} = f_{13} \cdot 1.0 \text{ (Rule 3)}$$
$$f_{13} = f_{13}$$

$$f_3 = f_{(TS_{12} \sqcap TS_{14}) \sqcup (TS_{11} \sqcap TS_{14}) \sqcup (TS_{13} \sqcap TS_{14})}$$
$$f_3 = [\min_{f_3}, \max_{f_3}] \text{ (Rule 4) where}$$

$$\min_{f_3} = \min(f_{TS_{12} \sqcap TS_{14}}) + \min(f_{TS_{11} \sqcap TS_{14}}) + \min(f_{TS_{13} \sqcap TS_{14}})$$
$$\min_{f_3} = (\max(0.2 \cdot c_9, 5) \cdot 0.2) + ((\max(0.2 \cdot c_9, 5) \cdot 0.24) + 0.1) +$$
$$\max(0.2 \cdot c_9, 5) \cdot 0.1$$
$$\min_{f_3} = (\max(0.2 \cdot c_9, 5) \cdot 0.54) + 0.1$$

$$\max_{f_3} = \max(f_{TS_{12} \sqcap TS_{14}}) + \max(f_{TS_{11} \sqcap TS_{14}}) + \max(f_{TS_{13} \sqcap TS_{14}})$$
$$\max_{f_3} = ((0.1 \cdot c_9) + (0.02 \cdot c_8) + 1.98) + ((3 \cdot c_9) + (0.06 \cdot c_8) + 6.19) +$$
$$((0.05 \cdot c_9) + (0.01 \cdot c_8) + 0.99)$$
$$\max_{f_3} = (3.15 \cdot c_9) + (0.09 \cdot c_8) + 9.16$$

We have now calculated frequencies for all the threat scenarios in Figure 71. We continue by calculating the frequency $f_4$ of the threat scenario $TS_{15}$ in Figure 72. We calculate $f_4$ as follows:

$$f_4 = f_{TS_{14} \sqcap TS_{15}}$$
$$f_4 = f_3 \cdot 0.01 \text{ (Rule 3)}$$
$$f_4 = [(\max(0.2 \cdot c_9, 5) \cdot 0.54) + 0.1, (3.15 \cdot c_9) + (0.09 \cdot c_8) + 9.16] \cdot 0.01$$
$$f_4 = [(\max(0.2 \cdot c_9, 5) \cdot 0.0054) + 0.001, (0.0315 \cdot c_9) + (0.00081 \cdot c_8) + 0.0916]$$

The next step is to calculate the frequency $f_5$ of the unwanted incident $UI_{11}$ occurring. We
calculate this frequency as follows:

\[ f_5 = f_{TS15 \cap UI1} = f_4 \cdot 0.9 \text{ (Rule 3)} \]

\[ = [(\max(0.2 \cdot c_9, 5) \cdot 0.0054) + 0.001, (0.0315 \cdot c_9) + (0.0009 \cdot c_8) + 0.0916] \cdot 0.9 \]

\[ = [(\max(0.2 \cdot c_9, 5) \cdot 0.00486) + 0.0009, (0.02835 \cdot c_9) + (0.0009 \cdot c_8) + 0.08244] \]

We continue by calculating Expected service level_{T} and Expected service level_{E}. We calculate these as follows:

\[
Expected\ service\ level_{T} = \frac{\text{Maximum service level}_{T} - (\text{Likelihood} \cdot \text{Consequence}_{T})}{\text{Maximum service level}_{T}} = \frac{8760 - (f_5 \cdot 24)}{8760}
\]

\[
Expected\ service\ level_{E} = \frac{\text{Maximum service level}_{E} - (\text{Likelihood} \cdot \text{Consequence}_{E})}{\text{Maximum service level}_{E}} = \frac{365000 - (f_5 \cdot 1000)}{365000}
\]

The final step is to define how Risk Value should be calculated. We calculate Risk Value as follows:

if \[ \frac{8760 - (f_5 \cdot 24)}{8760} \geq \frac{8760 \cdot 0.999}{8760} \text{ and } \]

\[ \frac{365000 - (f_5 \cdot 1000)}{365000} \geq \frac{364800}{365000} \text{ then} \]

Risk value = Acceptable

else

Risk value = Unacceptable
Chapter 11

Paper C: An architectural pattern for enterprise level monitoring tools
An Architectural Pattern for Enterprise Level Monitoring Tools

Olav Skjelkvåle Ligaarden*, Mass Soldal Lund*, Atle Refsdal†, Fredrik Seehusen*, and Ketil Stølen*†

*Department for Networked Systems and Services, SINTEF ICT, PO Box 124 Blindern, N-0314 Oslo, Norway
†Department of Informatics, University of Oslo, PO Box 1080 Blindern, N-0316 Oslo, Norway

E-mail: {olav.ligaarden, mass.s.lund, atle.refsdal, fredrik.seehusen, ketil.stolen}@sintef.no

Abstract—Requirements from laws and regulations, as well as internal business objectives and policies, motivate enterprises to implement advanced monitoring tools. For example, a company may want a dynamic picture of its operational risk level, its performance, or the extent to which it achieves its business objectives. The widespread use of information and communication technology (ICT) supported business processes means there is great potential for enterprise level monitoring tools. In this paper we present an architectural pattern to serve as a basis for building such monitoring tools that collect relevant data from the ICT infrastructure, aggregate this data into useful information, and present this in a way that is understandable to users.

Keywords—architectural pattern; enterprise level monitoring; software engineering

I. INTRODUCTION

Requirements from laws and regulations, as well as internal business objectives and policies, motivate enterprises to implement advanced monitoring tools. With cloud computing, it becomes even more important. An enterprise may take advantage of cloud computing by outsourcing all or parts of its information and communication technology (ICT) supported processes. In that case, monitoring becomes a necessity to ensure that the outsourcing party delivers the agreed performance. When designing an enterprise level monitoring tool based on an ICT infrastructure or a cloud system, we need to take into consideration the characteristics of the infrastructure/system. Events that can be observed directly at the ICT infrastructure level are typically of a low-level nature, such as service calls or responses. The significance of single events or types of events with respect to what we want to monitor and present at the enterprise level cannot usually be understood in isolation, and enterprise information has to be aggregated from a large number of different kinds of events and data. Extracting useful information from the available data in a manual fashion may be costly and unfeasible in practice. A main purpose of an enterprise level monitoring tool is therefore to close the gap between the low-level data obtained from the ICT infrastructure and the comprehension of human operators and decision makers. Ideally, such a monitoring tool will be able to collect all relevant input from the running ICT infrastructure, process this input so that it is turned into useful information, and present this information in a way that is understandable for those who need it. The desired level of detail depends on the role of the user. For example, a security engineer may want to know how often a port in a firewall is opened, while the company manager wants a high-level assessment of the information security risk to which the company is exposed.

Typically, a main objective for wanting to monitor something is to ensure that a certain value stays within an acceptable range. For example with respect to performance, we may want to monitor the average time delay between two steps of a business process, and to take suitable action if this delay is too long. This means that the monitor should automatically give a warning if the average delay is too long.

Irrespective of the particular kind of enterprise level monitoring tool that is being built and deployed, we consider the following capabilities to be core features of such a tool:

- Collect low-level data from the ICT infrastructure.
- Aggregate the collected low-level data.
- Evaluate the aggregated data.
- Present the aggregated data and the evaluation results to different types of enterprise users.
- Present the most recent aggregated data and evaluation results.
- Configure the tool with respect to:
  - The low-level data that should be collected.
  - How the low-level data should be aggregated into information that is relevant and useful.
  - How aggregated data should be evaluated.
  - The kind of aggregated data and evaluation results that should be presented and how this should be made comprehensible to different types of enterprise users.

In this paper we present an architectural pattern that should serve as a basis for building enterprise level monitoring tools with the above features. The pattern identifies the core components and shows how these components interact. We believe that this pattern is a good starting point for building specialized monitoring tools within various kinds of domains and enterprises.

The architectural pattern has been developed with the aim of fulfilling the following characteristics:

- It should serve as a basis for building monitoring tools within a wide range of domains and enterprises.
- It should facilitate modularity and reuse.
MonitorConfig

MonitorModel

MonitorConsole

DataSource

Figure 1. The components of the Enterprise Monitor pattern and its environment

• It should be scalable (handle growth in size or complexity).

The origin of the pattern is a risk monitor that we developed in the MASTER [1] research project. We believe that the risk monitor exhibits a number of features that are not specific to the monitoring of risks, but general to a broad class of enterprise level monitoring tools. We have therefore generalized the architecture of the risk monitor into an architectural pattern for this class of tools. In Section II we present this pattern. In Section III we demonstrate the pattern by showing the risk monitor as an instance, and we exemplify the use of the risk monitor in a health care scenario. After presenting related work in Section IV we conclude in Section V.

II. ARCHITECTURAL PATTERN

In the following we present our architectural pattern using part of the pattern template presented in [2]. According to this template, a pattern is described by the use of fourteen different categories. We do not use all categories since not all of them are relevant. The two categories Context and Problem have already been covered in Section I, while the Example category is covered in Section III.

A. Name and short summary

The Enterprise Monitor architectural pattern divides an enterprise level monitoring tool into three components: MonitorModel which contains the core monitoring functionality and data; MonitorConsole which presents core monitoring data in a specific way to a group of enterprise users; and MonitorConfig which is used by enterprise analysts to set up the enterprise level monitoring tool and to configure it during run-time.

B. Solution

In Fig. 1 is a UML [3] component diagram which shows the components of the Enterprise Monitor pattern as well as the component DataSource, which is part of the pattern’s environment. The Enterprise Monitor collects low-level data from DataSource during run-time. DataSource may for instance be a sensor network, an ICT infrastructure, a cloud system, a database, and so on. The external behavior of the different components is defined by provided and required interfaces. A provided interface, represented by a full circle, describes services that a component provides to other components, while a required interface, represented by a half circle, describes services that a component requests from other components. The interfaces are again connected to ports on the component.

The component MonitorModel contains the core monitoring functionality and data. An enterprise analyst configures MonitorModel using the component MonitorConfig. This component is also used by the analyst to re-configure MonitorModel during run-time. The MonitorModel is independent of how core monitoring data is presented to different enterprise users.

An Enterprise Monitor may have several MonitorConsoles, but for simplicity, only one is shown in Fig. 1. Each MonitorConsole presents core monitoring data in a specific way to a group of enterprise users. The MonitorConfig configures how each MonitorConsole should present core monitoring data to its enterprise users. The MonitorModel must keep track of which MonitorConsoles that depend on the different core monitoring data. Each MonitorConsole therefore notifies the MonitorModel about what core monitoring data it depends on. When core monitoring data is updated, the MonitorModel notifies the MonitorConsoles. These MonitorConsoles then retrieve the data and present it to their enterprise users. This change-propagation mechanism is described in the Publisher-Subscriber1 design pattern [2].

C. Structure

In this section we specify the structure of the pattern by the use of Class-Responsibility-Collaborator (CRC) cards [4] and a UML class diagram.

In Fig. 2 CRC-cards describing the three components are shown, while in Fig. 3 a UML class diagram describing how the components are related to each other is provided. Each CRC-card describes the responsibilities of one component and specifies its collaborating components. In the UML class diagram each component is represented by a class. The different classes are related to each other by associations, where each end of the association is given a multiplicity.

The responsibilities of MonitorConfig are to create a data config and a number of presentation models, and to configure the MonitorModel and MonitorConsoles. The relation between a data config and a presentation model is important for understanding the structure of the pattern. In Fig. 4 a UML

1 Also referred to as Observer.
class diagram that shows the relation between a presentation model and a data config is provided. We distinguish between basic and composite indicators. By a basic indicator we mean a measure such as the number of times a specific event generated by the ICT infrastructure has been observed within a given time interval, the timing of an event, the load on the network at a particular point in time, or similar. The basic indicators are the low-level data. A composite indicator is an aggregation of basic indicators. The information carried by a composite indicator should be both relevant and useful for an enterprise user. The basic indicators and the aggregation functions are defined in the data config. A presentation model specifies how results from the monitoring should be presented to the enterprise user. Each presentation model is parameterized with respect to the composite indicators in the data config. The presentation models can therefore easily be updated when the composite indicators change. This parameterization is exemplified in Fig. 5. The figure shows excerpts from a data config, in the form of a file, and a presentation model. An enterprise user is presented the presentation model with the value of $f_1$ inserted. In this example we only use a textual presentation model, but other kinds of models, such as graphical presentation models, may be used as well. We can also consider letting different values trigger different presentations. For instance in the example in Fig. 5, the enterprise user may be presented with a warning if the value of $f_1$ exceeds some threshold.

The MonitorModel keeps a registry, referred to as observerList in Fig. 3, to keep track of the composite indicators that the different MonitorConsoles depend on. The core monitoring data cmd contains the basic and composite indicators used by MonitorModel, as well as other data used during the monitoring. The MonitorConsoles realize the interface Observer and therefore implements the update procedure. A MonitorConsole’s update procedure is invoked by the notify procedure if one or more of the MonitorConsole’s composite indicators have been updated. The update procedure will then retrieve the updated composite indicators from MonitorModel, and the MonitorConsole will update its display.

**D. Dynamics**

This section presents typical scenarios, modeled by the use of UML sequence diagrams, that describe the dynamic behavior of the pattern. Each entity in a sequence diagram is a component and is represented by a dashed, vertical line called a lifeline, where the box at its top specifies which entity the lifeline represents, its name as well as its type separated by a colon. Entities interact with each other through the transmission and reception of messages, which are shown as horizontal arrows from the transmitting lifeline to the receiving lifeline.

The sequence diagram in Fig. 6 describes the initial configuration of the MonitorModel and MonitorConsoles, while the sequence diagram in Fig. 7 describes how the displays of the MonitorConsoles are updated when basic indicators are updated. For simplicity, we only use one MonitorConsole.

In Fig. 6 the entity config configures the entity model by the use of a data config d. During configuration, model will set up a connection with some data source, specified by a sequence diagram Data source setup (not shown in this paper) referred to by the ref construct in the diagram.
Figure 4. The relation between presentation model and data config

Figure 5. Parametrization exemplified

Figure 6. Sequence diagram “Configure”

Figure 7. Sequence diagram “Update basic indicators”

**E. Implementation**

In this section we provide guidelines for implementing the pattern in the form of code fragments written in Java. The code fragments serve only as suggestions, but they highlight some important aspects that should be considered when implementing the pattern. In the code fragments we use $bi$ and $ci$ to refer to basic and composite indicators, respectively. In Fig. 8 some of the code that may be used to implement MonitorModel is provided. The Maps $biVals$ and $ciVals$ use basic and composite indicator names, respectively, as keys, while the values assigned to the indicators during monitoring are used as values. The indicator names should be specified in the data config used to configure the MonitorModel. The Map $ciDefs$ uses composite indicator names as keys, while the values are the definitions of the composite indicators. This Map
is used when aggregating basic indicators. The last Map observers uses Observer objects as keys, where each object refers to a MonitorConsole, while the values are Maps. Each of these Maps uses composite indicator names as keys and Booleans as values. The value is true if the MonitorConsole has not retrieved the latest updated value of the composite indicator, and false otherwise. The Map observers keeps track of which MonitorConsoles have registered with MonitorModel, and makes sure that MonitorConsoles only retrieve composite indicators that have been updated.

The function updateCoreMonitoringData is executed after the MonitorModel has retrieved updated basic indicators from some data source. It takes a Map of basic indicator names and their updated values as input. The function updates the values of the basic indicators in the Map biVals. Then it checks for each composite indicator if it needs to be updated. This is done by checking whether the definition of the composite indicator refers to any of the basic indicators in biNames. If this is the case, the value of the composite indicator in ciVals is updated. The function aggregate produces this value based on the definition d and the basic indicator values stored in biVals. At the end, the function notify is called with the names of the updated composite indicators as parameter.

For each Observer object in the Map observers the function notify sets the value in obsMap to true if the key is a member of the set updatedCis. If obsMap contains one or more true values and if obsMap did not contain any true values before notify started updating its values, then update for the Observer o is called. update will not be called if updateCalled equals true since this means that update has already been called and that Observer o has not retrieved the updated composite indicators yet.

The definition of update is shown in Fig. 9. MonitorConsole implements the update function since it implements the Observer interface. The return value of calling the getData function, shown in Fig. 8, is a Map of composite indicators names and their updated values. The Map updatedCis is used to update the display of MonitorConsole. We can see from the definition of getData, in Fig. 8, that only updated composite indicators are retrieved. All values in the Map obsMap are false when getData returns, since Observer o has retrieved all updated composite indicators.

F. Consequences

In this section we present the consequences in the form of benefits and liabilities of applying the pattern.

The application of the Enterprise Monitor pattern has the following benefits:

- Separation of MonitorModel and MonitorConsoles.
  The pattern separates user interfaces from the core

```java
public Map<String, Object> getData(Observer o) {
    //Map<String, Boolean> obsMap = observers.get(o);
    for (String name : obsMap.keySet()) {
        if (obsMap.get(name) != null) {
            updatedCis.put(name, ciVals.get(name));
            obsMap.put(name, false);
        }
        if (!updatedCis && obsMap.containsValue(true)) {
            o.update();
        }
    }
    return updatedCis;
}
```

Figure 8. Some Java code for class MonitorModel
interface Observer {
    abstract void update();
}

class MonitorConsole implements Observer {
    private MonitorModel m;
    private PresentationModel p;

    public void update() {
        Map<String, Object> updatedCis = m.getData(this);
        updateDisplay(updatedCis);
    }

    private void updateDisplay(
        Map<String, Object> updatedCis) {
        // Update the presentation model "p" based
        // on "updatedCis".
        // Display the updated model.
    }
}

Figure 9. Some Java code for interface Observer and class MonitorConsole

Synchronized MonitorConsoles. The change-propagation mechanism ensures that all MonitorConsoles are notified about updates of their data at the correct time.

Efficient updates. MonitorConsoles are only notified when their data have been updated. The MonitorConsoles retrieve only the data that have been updated and nothing more.

Re-configuration. By using a data config and presentation models it is possible to re-configure the enterprise level monitoring tool during run-time.

Creation of MonitorConsoles during run-time. It is possible to create new MonitorConsoles and register them with the MonitorModel during run-time.

The liabilities of the pattern are as follows:

• Scalability. Using the pattern to build a very large enterprise level monitoring tool may lead to high complexity and a large number of MonitorConsoles. It may then be better to build separate tools for the different monitoring tasks.
• Close coupling of MonitorModel to MonitorConfig. The MonitorModel depends on the specific language in which the data config is expressed. Change of language may most likely require changing the code of the MonitorModel.
• Re-configuration. If the MonitorModel is re-configured it may also be necessary to re-configure a number of the MonitorConsoles. Losing track of which presentation models must be changed when the data config used by the MonitorModel is changed is a potential consequence.

G. Related patterns

The Model-View-Controller [2] (MVC) architectural pattern solves a related problem. This pattern was used as inspiration when we designed our pattern. MVC divides an interactive application into a model (core functionality and data) and different user interfaces, where the user interfaces presents the same core data in different ways. If the core data is updated in the model or in a user interface, then this change should be reflected in all user interfaces. As for our pattern, MVC uses the Publisher-Subscriber design pattern [2] to implement the change-propagation mechanism. This mechanism ensures consistency between the model and user interfaces.

III. DEMONSTRATION OF ARCHITECTURAL PATTERN

The risk monitor is an implementation of the architectural pattern presented in this paper. It is a monitor specialized at aggregating basic indicators related to risk and presenting the results in risk models. In particular, the language used for the presentation models is the CORAS risk modeling language. For more information on the language and the use of indicators in risk analysis, we refer to [5] and [6], respectively. The structure of the risk monitor is shown in Fig. 10 in the form of a UML component diagram.

The Run-time Risk Monitor contains the components that play a part at run-time, while the three components Risk Model Display, CORAS Diagram Editor, and Risk Data Config Editor provide a graphical user interface to the user;
they are either used to configure the monitor or view the output of the monitor. Fig. 11 shows the correspondence between the components of the risk monitor and the components described in the pattern.

In the Risk Monitor the Indicator Harvester retrieves updated basic indicators from a data repository, referred to as Data Warehouse in the diagram. The Risk Analyzer aggregates the basic indicators into composite indicators. The Risk Monitor Controller configures the Run-time Risk Monitor based on the data config provided by the Risk Data Config Editor, while the CORAS Diagram Editor configures the Risk Model Display by the use of a risk model.

The Risk Monitor implements its components as plug-ins to Eclipse. As a result, the Risk Monitor appears to the risk analyst as a standalone Eclipse tool with the three front-end components: CORAS Diagram Editor, Risk Data Config Editor, and Risk Model Display, as well as basic functionality to communicate with the Risk Monitor Controller, available as a integrated whole. The Risk Model Display is primarily used by the enterprise user during run-time, but can also be used by the risk analyst for testing when configuring the Risk Monitor.

In the following we concentrate on the configuration of the Risk Monitor, but first we give a little introduction to risk analysis by the use of CORAS.

A prerequisite for doing risk monitoring is to do a risk analysis. The outcome of a risk analysis is usually a description of threats to the target of analysis, vulnerabilities that the threats might exploit, unwanted incidents initiated by the threats, risks (which are usually defined as the likelihood of an unwanted incident and its consequence for a specified asset), and treatment of the risks.

In the CORAS methodology for model-based risk analysis [5], threats, threat scenarios (series of events initiated by threats, leading to unwanted incidents), vulnerabilities, unwanted incidents, and risks are modeled using a number of diagrams, referred to as CORAS diagrams. In Fig. 12, a simple CORAS threat diagram is shown. The example is adapted from one of the example cases used in the MASTER [1] project and concerns a hospital that has outsourced its drug stock management. A Drug Personalized Kit contains the drugs needed in the treatment of a particular patient. A drug provider is responsible for delivering Drug Personalized Kits to the hospital within a given deadline after receiving an order from the hospital. The diagram documents a part of a risk analysis where the focus is on protecting the asset “Patient health”. The unwanted incident “Inadequate health care due to unavailable or damaged drugs” may harm this asset. Two different threat scenarios which may lead to this unwanted incident have been identified, each represented by an oval with a warning sign. Both threat scenarios are initiated by the unintentional threat “Drug provider”. The first scenario is initiated via the vulnerability “Old transportation equipment”, while the second is initiated via the vulnerability “Poor quality control”. Moreover, the threat scenarios and unwanted incident, as well as the relations between them, are annotated with quantitative or qualitative likelihood (i.e. frequency or probability) values, and the relation between the unwanted incident and the asset is annotated with a consequence value. Fig. 12 represents a snapshot view at the point in time when the risk analysis was conducted, and the assigned likelihood and consequence values are assumed to apply at that point. The goal of the Risk Monitor, however, is to obtain risk pictures that are updated as the situation changes. In the Risk Monitor we achieve this by using values that are derived from basic indicators that represent observable and measurable properties of the target. These basic indicators are aggregated and presented as part of the risk picture.

To configure the Risk Monitor, the risk analyst needs to create a data config, used by the Run-time Risk Monitor, and a presentation model, used by the Risk Model Display. The risk analyst creates the data config by the use of the Risk Data Config Editor. In the data config, the risk analysts define:

- the basic indicators that should be monitored; and
- variables (composite indicators), aggregated from basic
indicators, for likelihood values, consequence values, trigger conditions, and risk levels.

The presentation model used is a risk model annotated with variables from the data config. The risk analyst creates the presentation model by the use of the CORAS Diagram Editor. In Fig. 13 an example presentation model is provided, while a small excerpt of an example data config, which provides definitions for how to assign values to some of the variables used in the presentation model, is shown in Fig. 14.

The diagram in Fig. 13 is similar to the diagram in Fig. 12, but it differs in two ways. First, the likelihood and consequence values have been replaced by variables whose values will be dynamically updated during run-time. For variable names we follow a convention where \( b \) is a basic indicator, \( f \) is a frequency, \( p \) is a probability, \( t \) is a triggering condition, \( c \) is a consequence, and \( r \) is a risk level.

Second, treatments have been attached to the two threat scenarios. Each of these will be shown to the enterprise user only if the Boolean variables \( t_1 \) and \( t_2 \) have been assigned the value \( true \).

From Fig. 14 we can see that the variables \( f_1, f_2, f_3, p_1, \) and \( p_2 \) are the likelihood variables that occur in Fig. 13, while \( b_1, b_2, \) and \( b_3 \) are the basic indicators whose values will be obtained via the Indicator Harvester. \( t_1 \) and \( t_2 \) are the Boolean variables that also occur in Fig. 13. The variable \( r \), that is used in the definition of \( t_1 \) and \( t_2 \), will be 0 if the risk level associated with the unwanted incident is acceptable and 1 otherwise. Note that the definitions of \( r \), \( p_1, p_2, \) and \( c \) have been omitted from the excerpt to save space.

The variables in Fig. 13 will be substituted for actual values retrieved from the Run-time Risk Monitor during run-time, while treatments will appear in the diagram if the retrieved Boolean variables have the value \( true \). We do not illustrate this further, as these diagrams will be similar to Fig. 12, with the possible addition of treatments.

IV. RELATED WORK

To the best of our knowledge, there exist no other architectural patterns that address the problem of building enterprise level monitoring tools with the core features described in Section I. As already mentioned, the MVC [2] architectural pattern solves a related problem. Our pattern and MVC are related, since MVC was used as inspiration when we designed our pattern. One of the main differences between the two patterns is that in MVC all the user interfaces display the same core data, while in our pattern the MonitorConsoles can display different core monitoring data, which means that they need to be updated differently.

Design patterns specifically targeted on building software health monitoring applications are described in [7]. It focuses on design patterns for sensors collecting information about the internal state and operation of software and how this information can be combined into software health indicators describing different aspects of software health. Design patterns [8] describe relationships between classes and objects. Our pattern is at the architectural level, since it describes relationships between components. Examples of architectural patterns can be found in [2]. Another approach for creating monitoring applications is described in [9], which presents a tool framework called Mozart that uses a model driven approach to create performance monitoring applications using indicators.

Within business intelligence (BI) there are many tools that do some form of enterprise level monitoring. In the following we mention some of the most central ones. One type of tools is digital dashboards [10]. A dashboard should present monitored data in a highly visual and intuitive way so that managers can monitor their progress towards their identified goals. Other types of tools that do a sort of monitoring are data [11] and process [12] mining tools. Within business intelligence, data mining uses techniques from statistics and artificial intelligence to identify interesting patterns in often large sets of business data. Process mining is used to extract information about business process by the use of event logs.

Both data and process mining use historical data, while
in our approach we deal with real-time data as well as historical data. Another type of tools that rely on monitoring is business performance management [13] tools. These tools are used to monitor, control, and manage the implementation of business strategies.

A monitoring tool closely related to our pattern is the MASTER ESB [14]. This tool is used to monitor compliance with access and usage policies in a system. The tool monitors low-level evidence data that is aggregated into meaningful evidence on how different parties comply with the policies. This evidence is then evaluated, and actions against compliance violations may be taken.

A central aspect of our pattern is the aggregation of data into more meaningful information. Fault trees [15], Markov models [16], and Bayesian networks [17] all specify some form of aggregation, and have support for updating the output values when the input values changes. In [18], fault trees are used with influence diagrams [19], a graphical language originally designed for supporting decision making based on the factors influencing the decisions. These diagrams are connected to leaf nodes in fault trees. The influencing factors contribute to the probabilities of the leaf nodes, and these probabilities are propagated to the unwanted incident specified at the root of the tree. In this case, the fault tree specifies how the low-level data at the leaf nodes should be aggregated into something more meaningful. It is possible to monitor these influencing factors as well as the input values for the other approaches that aggregate data.

The pattern presented in this paper uses basic and composite indicators. Indicators are also used by other monitoring approaches, such as in [20] where low-level indicators are aggregated to create high-level indicators. These high-level indicators are used to monitor business performance. The low-level and high-level indicators are equivalent to our two types of indicators.

V. Conclusion

In this paper we have presented an architectural pattern called Enterprise Monitor that may serve as a basis for building enterprise level monitoring tools. The pattern identifies the core components and shows how they interact. The applicability of the pattern is demonstrated by an instantiation in the form of a risk monitor.

We believe the architectural pattern captures the core features of enterprise level monitoring tools. The MonitorModel collects relevant low-level data in the form of basic indicators from the ICT infrastructure and aggregates the basic indicators into composite indicators. The MonitorConsoles retrieve the most recent updated composite indicators from MonitorModel, evaluate them if needed, and update the displays used by their enterprise users based on the composite indicators and evaluation results. The enterprise analyst can quite easily configure the two above-mentioned components by using a data config and presentation models. The analyst also has the option of re-configuring the components during run-time. Since the architectural pattern captures all the core features and since these features are general, the pattern can be used as a starting point for building specialized monitoring tools within various kinds of domains and enterprises.

The pattern supports modularity by separating the MonitorConsoles from the MonitorModel, making it quite easy to introduce new MonitorConsoles. There are however some dependencies between the MonitorModel and the MonitorConfig. The MonitorModel depends on the specific language that the data config is written in. If this language is changed, then most likely the code of the MonitorModel must be changed as well.

The design pattern also promotes reuse of components. It is possible to make very general implementations of some of the components since configurations are used to specialize the components. For instance, it is possible to make a very general MonitorModel, which becomes specialized when configured.

Today, enterprises are starting to take advantage of the possibilities provided by cloud computing. This will result in increased need and demand for enterprise level monitoring and new challenges with respect to the capabilities of enterprise level monitoring tools. In future work we will identify and address these challenges.

Acknowledgments

The research on which this paper reports has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER and NESSoS projects, both funded from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreements FP7-216917 and FP7-256980, respectively.

References


Chapter 12

Paper D: Experiences from using a UML-based method for trust analysis in an industrial project on electronic procurement
Experiences from using a UML-based method for trust analysis in an industrial project on electronic procurement

Tormod V. Håvaldsrud · Olav S. Ligaarden · Per Myrseth · Atle Refsdal · Ketil Stølen · Jon Ølnes

Published online: 24 August 2010
© The Author(s) 2010. This article is published with open access at Springerlink.com

Abstract This paper reports on experiences from using a UML-based method for trust analysis in an industrial project. The overall aim of the trust analysis method is to provide a sound basis for making trust policy decisions. The method makes use of UML sequence diagrams extended with constructs for probabilistic choice and subjective belief, as well as the capture of policy rules. The trust analysis method is evaluated with respect to a set of criteria. The industrial project focused on the modeling and analysis of a public electronic procurement (eProcurement) system.
making use of a validation authority service for validating electronic certificates and signatures.

**Keywords** Trust management · Modeling · Electronic certificates · Electronic procurement

1 **Introduction**

Trust is often linked to the notion of subjective probability. For example, inspired by [5, 10], [11] defines trust as the subjective probability by which an actor, the trustor, expects that another entity, the trustee, performs a given transaction on which its welfare depends. Unless you are a psychologist, subjective probabilities (or beliefs) are not very interesting as long as they are studied in isolation. In computer science we are interested in trust or the more general notion of belief only as long as it has an impact on the factual (or objective) behavior of a computer system or computer-based facility. Moreover, within computer science we are often more interested in the trust of an organization than the individual trust of a human being.

In order to analyze something we need a clear understanding of this “something” (or target) to be analyzed. This target is often captured in the form of a model. Unfortunately, modeling approaches of industrial maturity targeting the computer industry (like UML [13]) do not have the expressiveness required to fully cover the aspects of relevance for a trust analysis. This motivated us to develop a method for trust analysis based on UML sequence diagrams extended with constructs for capturing

1. beliefs of agents (humans or organizations) in the form of subjective probabilities;
2. factual (or objective) probabilities of systems which may contain agents whose behavior is described in terms of subjective probabilities;
3. trust decisions in terms of policy rules with the deontic modalities obligation, prohibition and permission.

This paper reports on experiences from using this method for trust analysis (first proposed in [17]) in an industrial project focusing on the modeling and analysis of a public electronic (eProcurement) system making use of a validation authority service for validating electronic certificates and signatures. The trust analysis was conducted on behalf of Det Norske Veritas (DNV) in the autumn of 2008. DNV’s goal was to obtain a better understanding of the potential usefulness of a service they offered for supporting trust-based decisions in systems which rely on electronically signed documents. The performance of the trust analysis is evaluated with respect to a set of evaluation criteria.

The rest of the paper is organized as follows: Sect. 2 introduces our modeling approach which is UML sequence diagrams extended with constructs for probabilistic choice and belief. It also gives a brief introduction on how to specify trust policies to ensure and enforce the desirable behavior of a system. Section 3 presents the method for trust analysis, that builds on the modeling and policy specification approaches introduced in Sect. 2. Section 4 outlines the industrial case we used to test the feasibility of the trust analysis method. It also presents a set of evaluation criteria. Section 5
presents the use of the trust analysis method in the industrial case. Section 6 presents the results from the evaluation based on the criteria identified in Sect. 4. And finally, in Sect. 7 we draw the main conclusion and present related work.

2 Modeling approach

UML 2.1 sequence diagrams [13] are widely used for the modeling and specification of information systems; in particular to capture communication or interaction between system entities.

In this section we give a brief introduction to UML 2.1 sequence diagrams and the constructs for probabilistic choice and belief proposed in [18]. We also explain how sequence diagrams can be enriched to capture policy rules with deontic modalities. We use a running example: Alice purchases items on the Internet. For many of the purchases, Alice needs to send advance payment to the seller of the item. In these cases Alice runs a risk of not receiving the item after paying for it. The challenge is to model the trust considerations made by Alice, as well as their impact on the observable behavior.

2.1 Basic constructs as in the UML standard

The diagram purchase in Fig. 1 address the situation where Alice has found an item, with an acceptable price, that she might be interested in purchasing. The keyword sd (sequence diagram) in front of the diagram name marks the diagram as a sequence diagram. Each entity modeled by the diagram is represented by a dashed, vertical line called a lifeline, where the box at its top specifies which entity the lifeline represents, its name as well as its type separated by a colon. If one lifeline represents several entities with different names but of the same type, we only specify the type. Entities interact with each other through the transmission and reception of messages, which are shown as horizontal arrows from the transmitting lifeline to the receiving lifeline. For each message we distinguish between two events; a transmission event, represented by the arrow tail, and a reception event, represented by the arrow head. The transmission events occur, of course, before the corresponding receive events. The events on each single lifeline are ordered in time from top to bottom.

According to Fig. 1, Alice starts by requesting a tender from the seller. The seller sends in response a tender, signed with his electronic ID (eID) to Alice. Based on this signed tender, Alice decides whether she trusts the seller to send the item after she has sent the advance payment. Alice may behave in two alternative ways, with respect to this decision. She can either send the advance payment, or she can cancel the deal. This is represented by the outermost alt operator, which specifies alternative behavior. A dashed horizontal line separates the alternative behaviors. If Alice trusts the seller and sends the advance payment, the scenario continues in two alternative ways. This is represented by the innermost alt operator. Either the seller sends the item to Alice, or the seller does not. In the latter case Alice is forced to write off the money she paid for the item.
2.2 Construct for probabilistic choice

The diagram **purchase** in Fig. 1 specifies behaviors that may occur, but not how often or with what probability. Alice interacts with several sellers, and the scenario in Fig. 1 will therefore be repeated several times. We want to model how Alice behaves with respect to all these sellers. For this purpose we introduce the **palt** construct for probabilistic choice. By using the **palt** construct we can say something about the probability of a specific behavior.

The diagram **purchase2** in Fig. 2 is a probabilistic version of **purchase** in Fig. 1. The numbers occurring after **palt** in the upper corner of the operator frame specify the probabilities of the various alternatives. In the outermost **palt** the first number
states that the scenario of the first operand occurs with a probability of 0.7, which means that this behavior occurs in 70% of the cases, while the second number states that the scenario of the second operand occurs with a probability of 0.3, which means that this behavior occurs in 30% of the cases. Furthermore, for the innermost `palt` operator the first operand has a probability of 0.8 of occurring, while the second operand has a probability of 0.2 of occurring. The probabilities in this diagram are factual in the sense that they are meant to reflect observable probabilistic behavior of the system.

2.3 Belief construct

It is clear that Alice behaves based on how much she trusts the seller of the item, but this notion of trust is not really reflected in the diagrams we have seen so far. They capture that she makes a choice, but not whether this choice is based on trust or something else. We want to model explicitly to what degree she needs to trust a seller before she sends advance payment.

Trust is the belief of a trustor that a trustee will perform a specific transaction on which the welfare of the trustor depends. Often the trustor will only expect the trustee to perform the transaction if another event has already occurred. In our example this event would be the sending of advance payment from Alice to the seller. Here, Alice is the trustor, while the seller is the trustee. Alice believes that there is a certain probability that the seller will send the item.

To model trust considerations we use so-called subjective sequence diagrams. Subjective sequence diagrams captures the subjective belief of an actor. Syntactically, subjective sequence diagrams differ from the ordinary sequence diagrams in two respects. Firstly, `ssd` (subjective sequence diagram) is used instead of `sd` to mark the diagram. Secondly, we annotate exactly one lifeline head with the keyword `subj`. This identifies the annotated entity as the subject, meaning that the diagram is used to capture this entity’s subjective belief. According to the subjective sequence diagram `trust` in Fig. 3 Alice believes that the probability of receiving the item after sending the payment to the seller Bob is 0.8, and that the probability of not receiving the item is 0.2.

Semantically, a subjective sequence diagram aims to capture the belief of some entity like a person, an organization, or even a computer to the extent a computer may be said to believe. An ordinary sequence diagram, on the other hand, aims to capture...
the factual reality, i.e. how things really are, independent of subjective beliefs, and such diagrams are in the following often referred to as objective sequence diagrams.

2.4 Combining objective and subjective diagrams

In the previous section we showed how we can model Alice’s trust in a seller using subjective sequence diagrams. In this section we explain how subjective diagrams relate to the objective ones. Alice will only send payment if her trust in the seller is sufficiently high, i.e. if it reaches a certain threshold. The threshold says how much trust Alice needs to have in the seller in order to send him advance payment. In the diagram purchase3 in Fig. 4, the threshold is represented as guards. A guard is a Boolean expression within square brackets. It constrains the choice of operand. An operand can only be chosen if its guard evaluates to true. We can see that the two guards refer to the variable trust.p. Here, trust refers to the subjective diagram trust in Fig. 4. Unlike the diagram in Fig. 3, which captures the trust with respect to one specific seller, this diagram uses the variable p for the probability of the palt operands. This variable can be referred to in the objective diagram, since it is an out parameter of the subjective diagram. The trust.p expression in the objective diagram refers to this output value. The guards in the objective diagram specify that Alice sends advance payment if she believes that the probability of receiving the item is greater than or equal to 0.5. This will happen in 70% of the cases, since the operand where this guard holds has the probability of 0.7.

2.5 Policy specification

A policy is a set of rules that determines choices in the behavior of a system [19], and is used in policy based management. Each rule determines a system choice of
behavior, where a given trust level is a decisive factor for each choice. Enforcement of the given rules aims to ensure the optimal balance of the risks and opportunities that are imposed by trust based decisions within the system.

To formalize the policy the trust analysis method, proposed in [17], uses Deontic STAIRS [21], which is a language for expressing policies, and based on UML sequence diagrams. Deontic STAIRS has the expressiveness to specify constraints in the form of obligations, prohibitions, and permissions, corresponding to the expressiveness of standard deontic logic [12]. Such constraints are normative rules that describe the desired system behavior. This reflects a key feature of policies, namely that they “define choices in behavior in terms of the conditions under which predefined operations or actions can be invoked rather than changing the functionality of the actual operations themselves” [20]. Furthermore, Deontic STAIRS supports the specification of triggers that define the circumstances under which the various rules apply. In particular, the policy triggers can specify the required trust levels for a particular choice of behavior to be constrained.

Figure 5 shows an example of a policy rule in Deontic STAIRS for the scenario described in this section. The keyword rule in the upper left corner indicates that the diagram specifies a policy rule, while obligedToPay is the name of the rule. The diagram consists of two parts, a trigger and an interaction that is the operand of a deontic modality.

The first operator with keyword trigger specifies the circumstances under which the rule applies and consists of an interaction and a condition. The former refers to a scenario such that when it occurs, the rule applies. In this case the scenario is the reception by Alice of a signed tender. The condition of the trigger limits the applicability of the rule to a set of system states. In this case it refers to the states in which the relevant trust level is 0.5 or higher.

The second operator with keyword obligation shows the modality of the rule, while its operand specifies the behavior that is constrained by the rule. In this case, the relevant behavior is that Alice sends payment to the seller. According to obligedToPay, she is obliged to do so, given that the trigger is fulfilled. On the other hand, if the keyword had been prohibition then Alice would have been prohibited from sending payment, while if the keyword had been permission then Alice could choose whether or not to send payment.
3 The trust analysis method

In this section we give a brief overview of the trust analysis method that was introduced in [17]. For further details we refer to [17] and of course Sect. 5 of this paper which describes how the method was used in the industrial project on which this paper reports.

Figure 6 shows an overview of the method. There are three major steps. The first step is to model the target, the second step is to analyze the target and the third step is to capture policies to optimize the behavior of the target based on the knowledge acquired in the first two steps.

Step 1. Modeling of target. In order to analyze a system, we first need to understand the system under analysis, including the behavior of its users. A major goal of the first step and the resulting models is to provide such an understanding. However, as most systems are highly complex, it is neither feasible nor desirable to take every detail into account. Therefore the target should be modeled at a level of abstraction suitable for the analysis to come. Thus, the models should only capture the aspects of the system that enhances our understanding of the decisions that are taken on the basis of trust and the considerations that lie behind these decisions, as well as the resulting system behavior and outcomes that are relevant.

As explained in Sect. 2, the modeling approach is based on UML sequence diagrams. The reason is that trust is mainly of relevance in the context of interactions between different entities, and sequence diagrams are well suited for modeling interactions. Moreover, UML sequence diagrams are fairly easy to understand at an intuitive level. This is important, as the models developed in the first step should serve as a point of focus for discussions and as an aid in communication between the analysts and participants throughout the analysis. The extensions of UML sequence diagrams provided by subjective STAIRS [18] ensures that the trust considerations behind decisions can be captured in the models, as well as the resulting system behavior.
Step 2. Analysis of target. After a suitable model of the target has been established, the next step is to conduct the actual analysis. This involves investigating the current system behavior and the way in which trust-based decisions are being made, as well as potential alternative behaviors. The aim is to obtain a good understanding of the risks and opportunities involved. The analysis is divided into four sub-steps.

Step 2.1. Identify critical decision points. In this sub-step critical decision points that will be further investigated are identified. This will typically be points where actors in the system make trust-based decisions. But it may also be points where one could benefit from introducing new trust-based decisions. For example, if time is a critical factor, it may be more important to make a quick decision than to make the optimal decision. In such cases, it may be better to allow actors to make decisions based on trust than to insist on more time-consuming decision procedures.

Step 2.2. Evaluate well-foundedness of trust. Trust involves a subjective estimate of the potential behavior of another entity. The second sub-step of Step 2 consists of evaluating to what degree the subjective estimates reflect reality. In the industrial project on which this paper reports this step was not relevant since our task was not to evaluate an existing trust solution, but rather to develop a policy from scratch.

Step 2.3. Estimate impact of alternative behavior. In this sub-step the impact of various alternative behaviors that the system may potentially perform is investigated with respect to risks and opportunities. The goal is to get an understanding not only of the current “as-is” system behavior, which may not be optimal, but also of potential alternatives. Typically, this involves asking “what if” questions about the system, and capturing the answers in models. For example: what would be the overall effect on the system behavior if a certain actor was more (or less) willing to engage in interactions with other entities? What happens if a different policy is applied when making a certain decision?

Step 2.4. Evaluate and compare alternative behavior. In the final sub-step, the different alternative behaviors that were identified and investigated in the previous step are evaluated and compared. The purpose is to identify behaviors that should be sought or avoided.

Step 3. Capturing a policy to optimize target. The final step of the method proposed in [17] consists of using the obtained knowledge about preferred behavior to form policies to ensure and enforce the desirable behavior.

4 The industrial project on electronic procurement

We now present the industrial project in which the trust analysis method outlined in Sect. 3 was applied. The trust analysis method was used to model and analyze a public eProcurement system, which makes use of a Validation Authority (VA) service for validating electronic certificates and signatures. We first present the public eProcurement system, before describing how this system can make use of the VA service. Then we present criteria for evaluating the trust analysis method.
4.1 Public electronic procurement

Public eProcurement is used by public authorities within the EU to award public work contracts, public supply contracts, and public service contracts to economic operators [16]. We consider only the open procedure for individual contracts as specified in [3]. In the open procedure, any interested economic operator may submit a tender. The procedure consists of three phases: eNotification, eTendering, and eAwarding. In the eNotification phase a procurement officer¹ creates a call for tenders. This call specifies the requirements of the contracting authority for the goods/services/works to be procured. In the eTendering phase, interested economic operators will create tenders containing legal, financial, and technical information. Before submitting the tender electronically, one or more persons representing the economic operator need to sign the tender with their electronic IDs (eIDs), issued by Certificate Authorities (CAs). When received by the system, the system will examine whether the tender is compliant with the requirements defined in the call, including examining whether the digital signatures in the tender are valid. The eAwarding phase begins after the deadline for submission has expired. In this phase the contract is awarded based on an evaluation of the received tenders.

4.2 The validation authority service

For the eProcurement system it is important to be able to accept electronically signed tenders from electronic operators from all over Europe, regardless of the eID used by the operator. Due to the potential large number of CAs, the technical validation of eIDs and digital signatures has some challenges with respect to scaling [14], but the real problem is the assessment of the risk implied by accepting a digital signature. Here, one particular concern is that an economic operator can refute the validity of the offer stated in the submitted tender, if awarded the contract. The eProcurement system can ensure that this risk is acceptable by making an assessment of the signature quality and accepting only those of a certain quality. The higher the quality is, the harder it would be for an economic operator to refute the validity of a submitted tender. The quality of a signature [15] can be decided from the quality of the eID, which is derived from the certificate policy of the CA, and the cryptography used. A certificate policy may be written in a foreign language and may refer to a foreign legislation, so with a large number of CAs, the contracting authorities will have a hard time determining the quality of digital signatures. Thus, it will be hard if not impossible for the contracting authorities to have agreements with all the CAs on which it may want to rely, which again limits the number of economic operators that can submit tenders. A solution to this, as proposed in [15], is to use a VA as the single trust anchor, as shown in Fig. 7. In the figure we can see that the VA supports a number of CAs. For each CA that it supports, the VA is able to assess the quality of the eIDs issued by this CA and the signatures produced with those eIDs. A relying party, in this case the eProcurement system, can then validate and assess the quality of the

¹Representative for the contracting authorities.
signature\textsuperscript{2} in a signed tender by issuing a request to the VA. If the eID used to create the signature has been issued by a supported CA, the VA will use the CA to validate the eID, while the quality of the signature is computed by the VA by applying the formula

\[
\text{Signature Quality} = \text{eID Quality} + \text{Hash Quality} + \text{Public Key Crypto Key Length Quality},
\]

which will assign a Signature Quality value from 0 to 20 according to criteria further specified in [15]. This value is then compared to the minimum required quality level requested by the eProcurement system. If the signature is valid, meaning that the technical validation of the eID and the signature was successful, and the signature has sufficient quality, the VA will give the tender a trusted verdict. Otherwise, the VA will give the tender a not trusted verdict.\textsuperscript{3} By trusting the VA and its assessments, the eProcurement system is able to trust any CA that the VA handles. Hence, the eProcurement system can support a large number of CAs and it gets a one-stop shopping service for verification of digital signatures and eIDs and quality assessment of digital signatures.

\textsuperscript{2}It is possible to sign a tender with more than one signature. The VA is able to make an overall quality assessment of all these signatures.

\textsuperscript{3}The VA can also give an inconclusive verdict. This will happen in the cases when the VA cannot validate the eID and/or signature and/or cannot assess the quality of the signature.
4.3 Evaluation criteria

A major objective with applying the trust analysis method in the industrial project was to get an idea of how well the method performs in a practical setting. To do so we need a set of evaluation criteria and they are characterized and motivated in the following. The criteria are general in the sense that they do not target the eProcurement system or VA service specifically; they are equally valid for other kinds of trust-related infrastructures on which the trust analysis method may be applied.

When evaluating a method for trust analysis there are of course many concerns. To make sure that we covered the most important concerns we started by identifying groups of stakeholders of relevance for the method. What is important for one group may of course be less so for another group. We identified three main groups of stakeholders, and the evaluation criteria are based on the point of view for each of these groups. First, the customers are those who pay for the analysis. Typically, this will be managers and decision makers. They do not necessarily take part in the analysis process themselves, but will use the results of the analysis as a basis for making policy decisions. Second, the analysts are those who will conduct the analysis (process) and document results. They know the analysis method, but cannot be assumed to know the particular target system at the start of the analysis. Third, the participants are people such as decision makers, system users, developers, or engineers with whom the analysts interact during the analysis process. We now present evaluation criteria classified according to the stakeholder group for which they are most relevant.

For the customer of a trust analysis, the overall goal is to make the right trust policy decisions. This requires a good understanding of the outcome of potential alternative trust policies. Hence:

**EC1**: The trust analysis should provide the customers with a good basis for making trust policy decisions. This means that sufficient information about the impact of the potential alternatives must be provided.

Clearly, the cost of the analysis needs to be justified with respect to the benefit for the customer. Hence:

**EC2**: The trust analysis should be cost effective.

The task of the analyst is to conduct the trust analysis and to document the findings within the allotted time and cost frame. This means that the trust analysis method should be sufficiently simple to be carried out within a reasonable time frame. However, as this is implied by the requirement expressed in **EC2**, we do not include this as a separate criterion. On the other hand, the analyst would like to document the findings, and in particular all assumptions and constraints on which their validity depends, to cover him/herself as much as possible. Hence:

**EC3**: The modeling approach should be sufficiently expressive to capture the information, assumptions, and constraints of relevance.

The participant is supposed to communicate her or his knowledge in such a way that the analysis will result in correct models of the target, and the models should serve as a means of communication between the analysts and participants. It is therefore important that the models are comprehensible for the participants when properly
Experiences from using a UML-based method for trust analysis

assisted by an analyst. Otherwise, it will be hard for them to identify shortcomings and errors. Hence:

EC4: The models should be comprehensible for the participants of the analysis.

5 Trust analysis in the industrial project

In this section we present how the trust analysis method as described in Sect. 3 was applied in the industrial case outlined in Sect. 4.

5.1 Step 1. Modeling the target

The trust analysis focused on the scenarios where the eProcurement system makes decisions based on trust, i.e. where it is decided whether a received tender should be trusted to be authentic or not. On the one hand, it was an objective to minimize the risk that non-authentic tenders were accepted for further evaluation in the eAwarding phase, as contracts should not be awarded based on non-authentic tenders. On the other hand, it was also an objective to avoid authentic tenders being rejected without further evaluation.

There was some discussion on whether non-authentic tenders actually represent a real problem. Although this may not be the case today, it was agreed that this may easily become a problem in the future, as the use of eProcurement increases. It is easy to imagine cases where economic operators submit false tenders in a competitor’s name. The motivation for this could be, for example, to ensure that minimum requirements on the number of received tenders to enable eAwarding are fulfilled, or to bind the competitor to unfavorable obligations, or to make the operator’s own tender appear more attractive compared to a false costly tender.

The decision on whether to trust the authenticity of a tender is made in the eTendering phase, while the selection of the best tender based on price, quality, and so on from the tenders judged to be authentic is made in the eAwarding phase. Thus, for the purpose of the trust analysis we are only interested in the behavior related to submission of tenders in the eTendering phase. The task in Step 1 is therefore to model this behavior. Figure 8 shows the resulting overview diagram.

First the eProcurement system needs to find the minimum quality level the signatures have to comply with to be accepted to the eAwarding phase. This particular process is described in more detail in the diagram chooseQualityPolicy in Fig. 9.4 Choosing the quality policy and communicating the choice to the VA is typically done even before the eNotification phase, since the economic operators must be informed about the requirements for the submission. Note that the eNotification phase is not captured in the models, as it is of little relevance for the trust analysis.

After the quality policy has been set an economic operator may submit a tender \( t \) to the eProcurement system. This is represented by the message submitTender(\( t \)) in the diagram. The eProcurement system will validate the signatures of this tender by

\( \text{ref} \) The construct is a reference to another diagram. Its meaning is the same as we would get by inserting the contents of the referred diagram at the place of the reference.
using the VA service validateSignatures(t). The VA will then use the required minimum quality level to decide whether the signature should be trusted or not.

The first operand of the outermost alt operator describes the case where the tender is reported trusted by the VA, and therefore is accepted for further evaluation by the eProcurement system. The second operand describes the case where the tender is reported as notTrusted or as inconclusive, and therefore rejected by the eProcurement system.

We now explain how the choice of quality policy level performed by the eProcurement system is captured in the models. Intuitively, the process of choosing one of the 21 quality policy levels can be described as follows: the eProcurement system uses a threshold value that specifies the least amount of trust that is needed to accept the risk of accepting a non-authentic tender. In order to balance the risk of accepting a non-authentic tender against the desire not to reject authentic tenders, the eProcurement system chooses the lowest quality policy level needed to ensure that its trust exceeds the threshold.

Figure 9 describes the process of choosing a suitable quality policy based on trust. The diagram chooseQualityPolicy shows that there are 21 alternative quality policies from which the eProcurement system may choose. After choosing the quality policy in terms of an assignment, the eProcurement system communicates to the VA service the chosen policy, as shown by the setQualityPolicy(qp) message at the bottom of the diagram.

The trust level which the threshold is compared to, is captured by expressions of the form trust(j).p, where j is one of the 21 quality policy levels; this value is bound to the input parameter qp of the subjective diagram trust in Fig. 9. So,

---

5Remember that the quality policy scale is from 0 to 20.
Experiences from using a UML-based method for trust analysis

Fig. 9 chooseQualityPolicy and trust describe how the quality policy is found and set

for example, trust(7).p yields the return value of the subjective sequence diagram trust, assuming quality policy level 7 is used. As shown by the lifeline head, the eProcurement system is the trustor. Therefore, the expression trust(j).p represents the trust of the eProcurement system that a tender that receives a trusted verdict from the VA service is indeed authentic, given that quality policy level j is used. Note that the diagrams in Fig. 9 do not refer to one specific VA. Hence, trust(j).p may yield different values for different VAs for the same tender.

5.2 Step 2. Analyzing the target

We now explain how the sub-steps of Step 2 were performed.

Step 2.1. Identify critical decision points. The only critical decision point that was identified was the point where the signature quality policy is chosen by the eProcurement system. The reason for this was that the analysis was performed from the point of view of the eProcurement system, with the purpose of setting the right trust threshold. This decision point is represented by the assignment in the diagram chooseQualityPolicy in Fig. 9.

Step 2.2. Evaluate well-foundedness of trust. As explained in Sect. 3, this step was not relevant in the project on which this paper reports. Our task was not to evaluate one particular trust solution, but rather come up with a policy for how to choose quality policy level.

Step 2.3. Estimate impact of alternative behavior. As the decision point under analysis was the choice of quality policy level, the task of this sub-step was to estimate the impact of choosing different levels with respect to how many of the authentic or non-authentic tenders would receive the different verdicts from the VA service. This would serve as a basis for the evaluation and comparison in the next sub-step. To do so, we obviously needed to collect data. No historical data were available, and we
had to base ourselves on expert judgments. The experts were all employees of DNV including the two co-authors from DNV. The data they provided is documented in Figs. 10, 11, and Table 1. Figure 10 is a refinement of Fig. 8. It is unchanged above the \( palt \) operator. The first operand of the \( palt \) operator represents the cases where tenders are authentic, while the second alternative represents the cases where they are non-authentic. We consider a tender to be authentic if it actually comes from the company (EO) in whose name it has been submitted. This applies even if the employees who have signed the tender electronically is not strictly speaking authorized by the company to do so, as long as the company acknowledges the tender and intends to honor its commitment. To emphasize the fact that the EO is the only entity who actually knows whether the tender is authentic, we have used guards on the EO lifeline in the model to represent whether a tender is authentic or not; the two cases are represented by the guards \( t\text{.}authentic==\text{true} \) and \( t\text{.}authentic==\text{false} \), respectively. The probability 0.95 assigned to the first \( palt \) operand in Fig. 10 captures that 95% of received tenders are authentic. The remaining 5% are non-authentic and the second \( palt \) operand is therefore assigned the probability 0.05. The two operands of the \( palt \) operator in Fig. 10 refer to the same parameterized diagram (i.e. tenderValidation in Fig. 11), as the behavior of the system for the two cases only differ with respect to the probabilities for the alternatives. The tenderValidation diagram has three input parameters; namely the quality policy \( (qp) \), the probability \( (x) \) of being judged as trusted by the VA with respect to the selected quality policy, and similarly the probability \( (y) \) of being judged as not trusted as opposed to inconclusive in the other case. The first operand of the outermost \( palt \) operator in tenderValidation gives the probability for the tender being reported trusted by the VA, and therefore is accepted for further evaluation by the eProcurement system. The second operand shows the case where the tender is reported as notTrusted or as inconclusive, and therefore rejected by the system; this will occur with probability \( 1 - x \). Within this alternative, the notTrusted verdict from the VA will occur with probability \( y \), while the inconclusive verdict will occur with probability \( 1 - y \).
The actual values of $x$ and $y$ depend on whether the tender is authentic or not. Therefore the references to `tenderValidation` in the operands of the `palt` in Fig. 10 bind $x$ and $y$ to different entities. In the first operand representing the cases where tenders are authentic, $x$ is bound to $t_1$ and $y$ is bound to $r_1$. In the second operand representing the cases where tenders are non-authentic, $x$ is bound to $t_2$ and $y$ is bound to $r_2$. The intuitive meaning of $t_1$, $t_2$, $r_1$, and $r_2$ for a given quality policy $qp$ can be summarized as follows: $t_1$ denotes the probability of assigning a trusted verdict to an authentic tender; $r_1$ denotes the conditional probability of assigning a not trusted verdict (as opposed to inconclusive) for an authentic tender given that a trusted verdict is not assigned; $t_2$ denotes the probability of assigning a trusted verdict to a non-authentic tender; $r_2$ denotes the conditional probability of assigning a not trusted verdict (as opposed to inconclusive) for a non-authentic tender given that a trusted verdict is not assigned.

Table 1 shows how the representatives from DNV estimate that the probabilities will vary according to the quality policy. Note that even though the VA service offers a quality scale from 0 to 20, it was deemed sufficient to analyze only five different quality policy levels for the purpose of this analysis. Based on a consideration of criteria for assigning quality levels, the following steps on the scale were selected for analysis: 0, 5, 7, 10, and 20. The first line in the table provides the values of the parameters $t_1$, $r_1$, $t_2$, and $r_2$, in the diagram in Fig. 10, if the quality policy 0 ($qp = 0$) is chosen. The second line provides the values in the case where the quality policy 5 is chosen and so on.

Given the data captured by Table 1, we calculated the probabilities for the possible combinations of authenticity and verdicts, which amounted to a simple multiplication of the probabilities assigned in the objective diagrams in Figs. 10 and 11. For example, the probability that a given tender is authentic and receives a trusted verdict is obtained by $0.95 \times t_1$, while the probability that it is non-authentic and receives an inconclusive verdict is obtained by $0.05 \times (1 - t_2) \times (1 - r_2)$. Table 2 shows the result from these calculations (when inserting the values from Table 1 for the variables $t_1$, $t_2$, $r_1$, and $r_2$).
Table 1  The probabilities corresponding to the quality policy

<table>
<thead>
<tr>
<th>Quality policy</th>
<th>t1</th>
<th>r1</th>
<th>t2</th>
<th>r2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.65</td>
<td>0.05</td>
<td>0.05</td>
<td>0.10</td>
</tr>
<tr>
<td>5</td>
<td>0.80</td>
<td>0.15</td>
<td>0.01</td>
<td>0.20</td>
</tr>
<tr>
<td>7</td>
<td>0.95</td>
<td>0.40</td>
<td>0.005</td>
<td>0.50</td>
</tr>
<tr>
<td>10</td>
<td>0.75</td>
<td>0.70</td>
<td>0.002</td>
<td>0.80</td>
</tr>
<tr>
<td>20</td>
<td>0.80</td>
<td>0.80</td>
<td>0.001</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Table 2  Probabilities for authentic and non-authentic tenders

<table>
<thead>
<tr>
<th>Quality policy</th>
<th>Authentic</th>
<th>Non-authentic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trusted</td>
<td>Not trusted</td>
</tr>
<tr>
<td>0</td>
<td>0.61750</td>
<td>0.01663</td>
</tr>
<tr>
<td>5</td>
<td>0.76000</td>
<td>0.02850</td>
</tr>
<tr>
<td>7</td>
<td>0.90250</td>
<td>0.01900</td>
</tr>
<tr>
<td>10</td>
<td>0.71250</td>
<td>0.16625</td>
</tr>
<tr>
<td>20</td>
<td>0.76000</td>
<td>0.15200</td>
</tr>
</tbody>
</table>

Figure 12 shows the left-hand part of Table 2 as a trend-graph, i.e. it shows the probability that a tender is authentic and receives each of the three possible verdicts, depending on the chosen quality level. On the left-hand side of the graph, we see that the probability of getting a trusted verdict is relatively low (but increasing), while the probability of getting an inconclusive verdict is correspondingly high (but decreasing). According to the domain experts providing the data, the reason for this is that when a request for tenders is announced with low certificate and signature requirements, relatively many of the received tenders will use certificates from CAs that are not supported by a VA; there are many CAs offering low quality eIDs, and a VA service is primarily aimed at supporting CAs with higher quality eIDs. After quality policy level 7, we see a decrease in the probability of receiving a trusted verdict. According to the same experts, this is due to the fact that when higher quality policy levels are used, more of the received tenders will use certificates from CAs that are supported by the VA, but of insufficient quality.

Figure 13 shows the right-hand part of Table 2 as a trend-graph, i.e. it shows the probability that a tender is non-authentic and receives each of the three possible verdicts. Here we are operating with very small scales, due to the small amount (5%) of non-authentic tenders, and the probability for getting the trusted verdict is almost non-existing for all quality policy levels. Not surprisingly, the probability for a non-authentic tender getting the not trusted verdict increases with increasing quality policy level. Furthermore, the probability of an inconclusive verdict decreases with increasing quality level, as more of the received tenders will use certificates from CAs that are supported by VA when high quality policies are used.

---

Recall that 95% of tenders are assumed to be authentic in all cases.
Step 2.4. Evaluate and compare alternative behavior. In order to decide which of the quality policies that will give the optimal behavior of the system, we looked at the probabilities for desirable and undesirable outcomes (opportunities and risks) for the
different quality levels, with the goal of finding the optimal balance. For each quality level, the desirable outcomes are as follows:

- A tender that has been accepted by the system, due to a trusted verdict from the VA, is authentic.
- A tender that has been rejected by the system, due to a not trusted or an inconclusive verdict from the VA, is non-authentic.

We seek to maximize the probabilities of these outcomes, since these outcomes represent opportunities. The probabilities are given as follows:

- \( P(\text{aut}|\text{acc}) \) — The conditional probability of a tender being authentic, given that it is accepted by the system.
- \( P(\text{not aut}|\text{not acc}) \) — The conditional probability of a tender being non-authentic, given that it is rejected by the system.

On the other hand, for each quality level, the undesirable outcomes are as follows:

- A tender that has been accepted by the system, due to a trusted verdict from the VA, is non-authentic.
- A tender that has been rejected by the system, due to a not trusted or an inconclusive verdict from the VA, is authentic.

We seek to minimize the probabilities of these outcomes, since these outcomes represent risks. The probabilities are given as follows:

- \( P(\text{not aut}|\text{acc}) \) — The conditional probability of tender being non-authentic, given that it is accepted by the system.
- \( P(\text{aut}|\text{not acc}) \) — The conditional probability of a tender being authentic, given that it is rejected by the system.

From the diagrams in Figs. 10 and 11 we get the following values directly: the probability \( P(\text{aut}) = a \) for a tender being authentic is 0.95, irrespective of the quality policy, while the probability for a tender being non-authentic is \( P(\text{not aut}) = 1 - a \).

We also have the probabilities \( P(\text{acc}|\text{aut}) = t_1 \) and \( P(\text{not acc}|\text{aut}) = 1 - t_1 \) for a tender being accepted and not accepted by the system, given that it is authentic, while \( P(\text{acc}|\text{not aut}) = t_2 \) and \( P(\text{not acc}|\text{not aut}) = 1 - t_2 \) give the probabilities for a tender being accepted and not accepted, given that it is non-authentic. Values for \( t_1 \) and \( t_2 \), depending on the chosen quality level, are taken from Table 1. The probabilities for a tender being accepted and not accepted are obtained as follows:

\[
P(\text{acc}) = P(\text{aut}) \times P(\text{acc}|\text{aut}) + P(\text{not aut}) \times P(\text{acc}|\text{not aut})
= a \times t_1 + (1 - a) \times t_2
\]

\[
P(\text{not acc}) = 1 - P(\text{acc})
\]

The conditional probabilities, mentioned above, were calculated for the different quality levels by applying Bayes’ theorem as follows:

\[
P(\text{aut}|\text{acc}) = \frac{P(\text{acc}|\text{aut}) \times P(\text{aut})}{P(\text{acc})} = \frac{t_1 \times a}{a \times t_1 + (1 - a) \times t_2}
\]
Table 3  Table showing the probabilities related to opportunity (column 2 and 5) and risk (column 3 and 4) for the different quality policies

| Quality policy | \(P(\text{aut}|\text{acc})\) | \(P(\text{aut}|\text{not acc})\) | \(P(\text{not aut}|\text{acc})\) | \(P(\text{not aut}|\text{not acc})\) |
|----------------|----------------|----------------|----------------|----------------|
| 0              | 0.995968       | 0.875000       | 0.004032       | 0.125000       |
| 5              | 0.999343       | 0.793319       | 0.000657       | 0.206681       |
| 7              | 0.999723       | 0.488432       | 0.000277       | 0.511568       |
| 10             | 0.999860       | 0.826374       | 0.000140       | 0.173626       |
| 20             | 0.999934       | 0.791832       | 0.000066       | 0.208168       |

\[
P(\text{aut}|\text{not acc}) = \frac{P(\text{not acc}|\text{aut}) \times P(\text{aut})}{P(\text{not acc})} = \frac{(1 - t1) \times a}{1 - (a \times t1 + (1 - a) \times t2)} \quad (4)
\]

\[
P(\text{not aut}|\text{acc}) = 1 - P(\text{aut}|\text{acc}) \quad (5)
\]

\[
P(\text{not aut}|\text{not acc}) = 1 - P(\text{aut}|\text{not acc}) \quad (6)
\]

The results of the calculations are shown in Table 3. For \(P(\text{aut}|\text{acc})\), which we want to maximize, there is little difference between the values of \(P(\text{aut}|\text{acc})\) for the different quality policies. On the other hand, for \(P(\text{not aut}|\text{not acc})\), which we also want to maximize, we see that for level 7 we have a much higher value than for the others.

5.3 Step 3. Capturing a policy to optimize target

The numbers in Table 3 provide useful input, assuming of course that the expert judgments are sound. However, they do not take the nature of the call for tender into consideration, which of course is an essential factor when formulating a policy. After all, the significance of the numbers in Table 3 depends heavily on what is to be procured. If the cost of goods to be procured is low (e.g. pencils for the administration), we would probably worry only about \(P(\text{aut}|\text{acc})\) and based on that choose quality policy 0. This is partly because the difference in \(P(\text{aut}|\text{acc})\) for the quality policy levels does not matter much when the cost of goods to be procured is low, and partly because a higher quality level might frighten off potential submitters of tenders.

On the other hand, if the cost of the goods to be procured is very high (e.g. new fighter planes in the extreme case) the procurer would probably want as much legal coverage as possible and use quality policy level 20, since this gives the best value for \(P(\text{aut}|\text{acc})\). Moreover, if the goods to be procured are so costly that it is important to avoid disqualifying authentic tenders as well as obtaining a high level of trust in certificates, quality policy level 7 seems to be the best option.

Based on these considerations we ended up with the policy rules specified in Figs. 14–16. We make the assumption that the eProcurement system trusts the VA and its assessments. This is important since an eProcurement system cannot make use of the VA service if it is not trustable. The trigger of each rule contains a condition which limits the applicability of the rule to a set of system states. For rule \texttt{qp0} in Fig. 14 the condition is that the cost of the goods to be procured is low, while for rule \texttt{qp20} in Fig. 15 it is that the cost is very high. For rule \texttt{qp7} in Fig. 16 the condition
is that the cost is high, that disqualifying authentic tenders should be avoided, and a high level of trust in certificates is required. Depending on which one of these three conditions that is satisfied, the eProcurement system must use either quality policy level 0, 7, or 20.
6 Evaluation of the trust analysis method

In this section we evaluate the performance of the trust analysis method in the industrial project with respect to the criteria presented in Sect. 4.3.

**EC1**: The trust analysis should provide the customers with a good basis for making trust policy decisions. This means that sufficient information about the impact of the potential alternatives must be provided.

The project gave strong indications that the trust analysis method is feasible in practice. We went through the various steps of the method (with exception of Step 2.2) in close interaction with the industrial representatives of the customer and delivered a result in the form of a policy that we believe gives an institution making use of eProcurement system with a VA service useful input on selecting the right quality policy level. Based on models developed in the modeling step of the method we collected expert judgments and documented them in the models.

Of course an industrial case like this can never give solid repeatable evidence of anything. There are too many factors influencing what happens. In the case of our project it may be argued that we should have used historical data rather than expert judgments, but such data were not available. It may also for example be argued that we should have had involvement of representatives of an eProcurement institution, and having the inventors of the trust analysis method in the analyst team is of course also rather extraordinary.

**EC2**: The trust analysis should be cost effective.

The trust analysis was carried out in a series of five meetings, each of which took about 1.5 hours. Typically, four analysts and two to four participants/representatives of the customer took part in the meetings. In addition, the analysts spent time between the meetings developing models and preparing the next meeting. Table 4 shows an estimate of the total amount of time spent on the trust analysis. Note that time spent on writing a final report is not included in the numbers—this depends heavily on the type of report the customer wants. There are some issues that must be taken into consideration when evaluating these numbers. Firstly, this was the first time the trust analysis method was applied to a real industrial case. Hence, even though the analysis team included authors of the paper [17] proposing the trust analysis method, none of the analysts had any experience with applying the method in a realistic setting. It can reasonably be assumed that the process will be more effective as the analysts gain experience with applying the trust analysis method. Furthermore, the reason for having as many as four analysts was a desire to learn as much as possible from this first application of the method. Normally, we believe two analysts would be enough.

Based on the experience gained, we believe that it should be possible to carry out this kind of analysis with within a time frame of ca. 80 man-hours spent by analysts

<table>
<thead>
<tr>
<th>Table 4</th>
<th>The number of hours used on the trust analysis, not including writing of a final report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Meetings</td>
</tr>
<tr>
<td>Analysts</td>
<td>27</td>
</tr>
<tr>
<td>Participants</td>
<td>20</td>
</tr>
</tbody>
</table>


(not including writing a final report) and ca. 20 man-hours spent by participants. Whether this counts as being cost effective has to be evaluated in light of the values at stake in the target of analysis.

**EC3:** The modeling approach should be sufficiently expressive to capture the information, assumptions, and constraints of relevance.

The participants provided a lot of information about the target during the analysis process. There were no instances where we were not able to capture the relevant information in the models. The diagrams in Figs. 8–11 contain all the information that was finally used (and deemed relevant) for finding the best trust policy. These diagrams have been abstracted from more detailed models of the target. We also formalized the policy we recommended in the end.

**EC4:** The models should be comprehensible for the participants of the trust analysis.

During the meetings, models were presented and explained by an analyst in order to validate the correctness of the models. There were many instances where the participants pointed out parts of a model that did not correctly represent the target, provided additional information, or asked relevant questions about some detail in a model. This indicates that the models were in general comprehensible for the participants, and our experience is that the models served well as an aid in establishing a common understanding of the target between the participants and analysts. The fact that all the participants in this analysis had a strong technical background may have contributed to making the models easier for them to understand than would be the case for a more diverse group. Note also that we do not know whether the models would have been well understood by the participants without any guidance or explanation, as all the models were presented by an analyst. In particular with respect to subjective diagrams and their relation to the objective diagrams, we believe it is necessary to have an analyst explain the diagrams in order for them to be understood by the participants if they have no prior experience with the notation, other than some background in UML.

There was one aspect of the models that proved hard to understand for the participants. This occurred when the operands of \( \text{palt} \) operators contained more \( \text{palt} \) operators. In Fig. 10 the \( \text{palt} \) operator contains references to the diagram in Fig. 11, which again contains a \( \text{palt} \) operator with another \( \text{palt} \) operator inside one of its operands. This nesting of operators made it hard for the participants to understand exactly what each of the alternatives represented. In order to explain, one of the analysts drew a tree-structure where the root represented the outermost \( \text{palt} \) operator and each branch represented a \( \text{palt} \) operand. Based on this experience, we believe that the presentation style of UML interaction overview diagrams are better suited than sequence diagrams to present cases with nested alternatives. Interaction overview diagrams have the same kind of semantics as sequence diagrams and are often used in combination with sequence diagrams, but nested alternatives are represented syntactically by a branching point (the operator) with branches (the operands), rather than boxes inside boxes.
7 Conclusion and related work

The paper has presented experiences from using a UML-based method for trust analysis in an industrial project focusing on the modeling and analysis of a public eProcurement system making use of a validation authority service for validating electronic certificates and signatures. The contributions of the paper include:

1. a detailed account of how the method proposed in [17] scales in an industrial context; in particular, we have illustrated the specific constructs for capturing
   - beliefs of agents (humans or organization) in the form of subjective probabilities;
   - factual (or objective) probabilities of systems which may contain agents whose behavior is described in the form of subjective probabilities;
   - trust decisions in the form of policy rules;
2. an evaluation of the feasibility of the method in an industrial context; in particular, it is claimed that
   - the project gave strong indications that the trust analysis method is feasible in practice;
   - this kind of trust analysis can be carried within the frame of 100 man-hours (not including writing of a final report);
   - there were no instances were the analysts were not able to capture the relevant information in the models;
   - the models to a large extent were comprehensible for the industrial participant with some experience in UML but no background in the specific extensions used by the method.

The method for trust analysis makes use of models that capture the subjective trust considerations of actors, as well as their resulting behavior. We are not aware of other approaches that combine these elements in this way. However, the issues of uncertainty, belief, and trust have received much attention in the literature. We now present a small selection of the proposed approaches.

Giorgini et al. [6] presents a formal framework for modeling and analyzing trust and security requirements. Here, the focus is on modeling organizations, which may include computer systems as well as human actors. The approach is based on a separation of functional dependencies, trust, and delegation relationships. Trust and security requirements can be captured without going into details about how these will be realized, and the formal framework supports automatic verification of the requirements.

An interesting approach to modeling and reasoning about subjective belief and uncertainty is subjective logic [7, 8], which is a probabilistic logic that captures uncertainty about probability values explicitly. The logic operates on subjective belief about the world. Different actors have different subjective beliefs, and these beliefs are associated with uncertainty. The approach makes it possible, for example, to calculate to what degree an actor believes that a system will work based on the actor’s beliefs about the subsystems, or to calculate the consensus opinion of a group of actors. Subjective logic deals strictly with the actors’ beliefs and reasoning, and does not address the question of how their beliefs affect their behavior.
The belief calculus of subjective logic can be applied in risk analysis to capture the uncertainty associated with such analysis, as shown in [9]. This is achieved by using subjective beliefs about threats and vulnerabilities as input parameters to the analysis. Through application of the belief calculus, the computed risk assessments provides information about the uncertainty associated with the result of the analysis.

With respect to situations in which the outcome of a choice of one actor depends on the subsequent choice of another actor, the field of game theory [4] is highly relevant. Game theory provides strategies for making rational choices with respect to desirable and undesirable outcomes from the point of view of the different players/actors. These potential outcomes are described by a payoff structure in terms of the loss and gain to which the various players are exposed; a rational player will seek the outcome with the best payoff for herself. Not surprisingly, game theory can also be applied to analyze trust, as shown by Bacharach and Gambetta [1]. They explain how the trustor’s choice to trust or not, and the trustee’s subsequent choice to deceive or not, can be modeled in terms of this rational choice theory.

A formal model for trust in dynamic networks based on domain theory is proposed by Carbone et al. in [2]. Here, trust is propagated through delegation in a “web of trust”, where the trust of one actor is affected by the trust of other actors. An important contribution of the approach is the distinction between a trust ordering and an information ordering. The former represents degrees of trust, while the latter represents degrees of precision of information from which trust is formed. An interval construction is introduced to capture uncertainty, and a simple trust policy language is proposed based on the formal model.

Acknowledgements We would like to thank Aida Omerovic (SINTEF ICT) and Anette Andresen (DNV) for participating in discussions and commenting on the work in the industrial project.

Open Access This article is distributed under the terms of the Creative Commons Attribution Noncommercial License which permits any noncommercial use, distribution, and reproduction in any medium, provided the original author(s) and source are credited.

References

Experiences from using a UML-based method for trust analysis


Chapter 13

Paper E: Experiences from using indicators to validate expert judgments in security risk analysis
Report

Experiences from Using Indicators to Validate Expert Judgments in Security Risk Analysis

Author(s)
Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stølen
Report

Experiences from Using Indicators to Validate Expert Judgments in Security Risk Analysis

VERSION
Final version

DATE
2012-01-09

AUTHOR(S)
Olav Skjelkvåle Ligaarden, Atle Refsdal, and Ketil Stalen

CLIENT(S)
Research Council of Norway

PROJECT NO.
908245

CLIENT'S REF.
180052/510

NUMBER OF PAGES/APPENDICES:
16/4

ABSTRACT
Expert judgments are often used to estimate likelihood values in a security risk analysis. These judgments are subjective and their correctness relies on the competence, training, and experience of the experts. Thus, there is a need to validate the correctness of the estimates obtained from expert judgments. In this paper we report on experiences from a security risk analysis where indicators were used to validate likelihood estimates obtained from expert judgments. The experiences build on data collected during the analysis and on semi-structured interviews with the client experts who participated in the analysis.

PREPARED BY
Olav Skjelkvåle Ligaarden

CHECKED BY
Bjørnar Solhaug

APPROVED BY
Bjørn Skjellaug

REPORT NO.
SINTEF A21560

ISBN
978-82-14-04998-5

CLASSIFICATION
Unrestricted

SIGNATURE

CLASSIFICATION THIS PAGE
Unrestricted
# Table of Contents

I Introduction ........................................ 1

II Security risk analysis case .................. 1

III Data collected from case ................. 2

IV Discussion ........................................ 2
   IV-A Step 1 ........................................ 2
   IV-B Step 2 and 3 ................................ 3
   IV-C Step 4 ........................................ 3
   IV-D Step 5 ........................................ 3
   IV-E Step 6 ........................................ 4

V Conclusion ......................................... 4

References ........................................... 4

Appendix A: Security risk analysis case .... 5

Appendix B: Data collected from case ...... 6
   B-A Data for Step 1 ............................... 6
   B-B Data for Step 2 ............................... 6
   B-C Data for Step 3 ............................... 8
   B-D Data for Step 4 ............................... 8
   B-E Data for Step 5 ............................... 8
   B-F Data for Step 6 ............................... 8

Appendix C: Thematic analysis ............... 8
   C-A Procedure for collecting data ........... 8
   C-B Procedure for analyzing data .......... 10
   C-C Results from the thematic analysis ... 11

Appendix D: Threats to validity ............. 11
Experiences from Using Indicators to Validate Expert Judgments in Security Risk Analysis

Olav Skjelkvåle Ligaarden∗†, Atle Refsdal∗, and Ketil Stølen∗†

∗ Department for Networked Systems and Services, SINTEF ICT, PO Box 124 Blindern, N-0314 Oslo, Norway
E-mail: {olav.ligaarden, atle.refsdal, ketil.stolen}@sintef.no
† Department of Informatics, University of Oslo, PO Box 1080 Blindern, N-0316 Oslo, Norway

Abstract—Expert judgments are often used to estimate likelihood values in a security risk analysis. These judgments are subjective and their correctness relies on the competence, training, and experience of the experts. Thus, there is a need to validate the correctness of the estimates obtained from expert judgments. In this paper we report on experiences from a security risk analysis where indicators were used to validate likelihood estimates obtained from expert judgments. The experiences build on data collected during the analysis and on semi-structured interviews with the client experts who participated in the analysis.

Keywords—security risk analysis; expert judgment; indicator

I. INTRODUCTION

Much research report on procedures for eliciting expert judgment in risk analysis, decision support, and in general [1], [2], [3], [4]. There is also research that address the quality of expert judgments [5]. In this paper, however, we focus on the validation of the estimates obtained from expert judgments.

One way to validate likelihood estimates based on expert judgments is to use indicators calculated from historical data. Since we base ourselves on historical data, it is in most cases not possible to define indicators from which likelihood values can be inferred directly. For instance, in the case of the unwanted incident “eavesdropper reading a sensitive e-mail”, an obvious indicator would be the number of times this has occurred in the past. However, as it is normally not feasible to calculate this from historical data, we will have to make do with other indicators that are less to the point. One potential indicator for this unwanted incident could for example be “the number of encrypted sensitive e-mails sent”. Together with knowledge about the total number of sensitive e-mails being sent during a given period of time, this provides relevant input for validating the likelihood estimate.

In this paper we report on experiences from using indicators to validate expert judgments in a security risk analysis conducted in 2010. We build on data collected during the analysis and on semi-structured interviews with the client experts that participated in the analysis.

The rest of the paper is structured as follows: in Section II we present the security risk analysis from 2010. In Section III we present the most relevant data collected during the analysis. In Section IV we discuss this data in relation to input from the semi-structured interviews, while in Section V we conclude.

A more detailed description of the security risk analysis case is given in Appendix A. Appendix B presents all the data collected during the analysis. In Appendix C we present the thematic analysis of the semi-structured interviews, while in Appendix D threats to the validity of our conclusions are discussed.

II. SECURITY RISK ANALYSIS CASE

Our empirical study was integrated in a commercial security risk analysis based on CORAS [6] conducted in 2010. As the client of this analysis requires full confidentiality we can not report on the system assessed, the risk models obtained, the personnel from the client involved, or the name of the client.

Fig. 1 depicts the fragment of the security risk analysis of relevance for this paper as a process of six steps. Step 1 is the likelihood estimation based on expert judgments. Indicators for validation of the likelihood estimates were identified in Step 2. The analysis team proposed a number of indicators, and these indicators were revised during a meeting with the client experts in Step 3. During this meeting some indicators were rejected, some were subject to minor modifications,
and some new indicators were identified. In Step 4 the analysis team formulated validation criteria for the likelihood estimates in terms of indicators. Each criterion specifies the expected values of the indicators related to the likelihood estimate in question. Here, each criterion makes a prediction about the value of a set of indicators under the assumption that the likelihood estimate in question is correct. Indicator values were obtained by the client experts in Step 5. In Step 6 the validation criteria were evaluated and some of the initial likelihood estimates were adjusted.

In total, an estimated number of 400 hours were spent on the security risk analysis (not including writing the final report) by the analysis team. Three domain experts (E1, E2, and E3) of the client participated in the analysis. The client experts held degrees equivalent to Master of Science and four to ten years of experience in information security and risk analysis.

### III. DATA COLLECTED FROM CASE

For each step of the estimation and validation process, depicted in Fig. 1, we collected data, documented in Table I.

In **Step 1**, we came up with 28 likelihood estimates based on expert judgments.

In **Step 2**, the analysis team proposed at least one indicator for each of the 28 likelihood estimates. In total, 68 indicators were proposed.

In **Step 3**, the indicators proposed in Step 2 were revised in a meeting with the client experts. Some of the proposed indicators were rejected during this meeting, because their values were not obtainable within the client’s organization. After Step 3, there were 25 out of 28 likelihood estimates with at least one indicator. In total, 57 indicators remained after Step 3 had been conducted.

In **Step 4**, 19 indicators were used by the analysis team to formulate validation criteria for 15 likelihood estimates. For 10 likelihood estimates, validation criteria were not formulated. One of these 10 likelihood estimates was not assigned a criterion because the validation of the estimate was given a low priority by the client experts. For the remaining nine likelihood estimates, the analysis team was not able to come up with good validation criteria, although the indicators were considered to provide relevant information for validating the likelihood estimates.

In **Step 5**, the client experts obtained values for 13 out of the 19 indicators used to formulate the 15 validation criteria. This resulted in that only 10 out of the 15 validation criteria could be evaluated after Step 5.

In **Step 6**, we evaluated 10 validation criteria based on the values obtained by the client experts. The validation criteria were fulfilled for four likelihood estimates, while for two likelihood estimates we could not say whether the criteria were fulfilled or not, because the values of the indicators referred to in the criteria were too uncertain. The criteria were not fulfilled for the remaining four likelihood estimates. For two of these estimates, the client experts decided to adjust the likelihood estimates.

### IV. DISCUSSION

With each of the three client experts, we conducted a semi-structured interview, focusing on likelihood estimation based on expert judgments and the use of indicators to validate these. The transcribed interviews were analyzed by the use of a simplified version of thematic analysis [7]. In this section we discuss the data in Table I in relation to the results from the thematic analysis.

#### A. Step 1

Experts E1 and E2 were quite confident that their likelihood estimates were correct, while expert E3 did not want to give a clear yes or no answer to this. Even though they believed the estimates to be correct, expert E1 pointed out that validation in terms of indicators still has a purpose: “... I think there were one or two such cases where we had to adjust the estimates because of their indicator values. So I think the quality was good anyway but it is still an extra
quality adjustment when you get acknowledgments or only minor adjustments of the estimates.”

B. Step 2 and 3

It was challenging to identify relevant indicators for which values could actually be obtained within the available time and resources for the analysis. Expert E1 supports this: “It was a challenge in the least because it is terribly difficult to find good indicators of information security, and there were a number of examples where it was actually not possible to find indicators. Even though we had proposals we discovered later that they were not usable. But there were also areas where we came up with indicators that could be used.”

During the revision meeting in Step 3, many indicators were rejected because their values were not obtainable within the client’s organization. This resulted in that three likelihood estimates were left without indicators. One might argue that we should have used more time to identify indicators in Step 2, and also that we should have involved the client experts in this step. With respect to the former argument, according to our records we spent about 50 hours to identify indicators in Step 2, which is quite a lot when considering that about 400 hours were spent on the whole security risk analysis. With respect to the latter argument, all three client experts were of the opinion that the analysis team should come up with the initial indicator proposals. Expert E1 even expressed: “... I also think that when it comes to indicators, it can be a strength that they are proposed by someone else who does not have built-in limitations with respect to ideas.”

On the other hand, we could perhaps have obtained information from the client experts on the kinds of data, in the form of logs and so on, that are available within their company, prior to identifying indicators in Step 2. This would most likely have resulted in fewer indicator proposals being rejected due to their values not being obtainable. On the other hand, proposing relevant indicators where their values are not obtainable at the time of analysis may also prompt the client organization to implement more measurements, as expressed by expert E2: “It turned out that some of the measurements that had not been carried out should perhaps have been carried out, and that is the experience obtained from what we found here.”

C. Step 4

The analysis team was not able to formulate validation criteria for nine out of 25 likelihood estimates. We do not have the opinions of the client experts on this matter. They were not asked to comment on the formulation of validation criteria in the interviews, since this task was conducted solely by the analysis team.

The indicators of the nine estimates were considered relevant for validating the estimates, but we could not figure out how to link them to the estimates. Common for these indicators is that they are only indirectly linked to the estimates of which they were seen as relevant. An example of such an indicator is “the number of code lines used to produce the web server application” which is indirectly linked with the likelihood estimate of the unwanted incident “hacker takes over the web server by exploiting weaknesses in its code”. In many cases it is reasonable to believe that the number of weaknesses will increase with the number of code lines. However, it is not easy to predict how the value of this indicator affects the likelihood estimate since the estimate depends on a lot of other factors as well. On the other hand, the indicator “the number of times the web server was taken over by hackers during the past five years due to weaknesses in its code” is directly linked with the likelihood estimate of the incident. It is not surprising that it is easier to formulate validation criteria based on this kind of more direct indicators than by the use of the more indirect ones.

Eight out of the 10 validation criteria evaluated in Step 6 used an indicator that is directly linked to the likelihood estimate. In relation to this it must be pointed out that we would have had seven validation criteria using solely indirect indicators if we had managed to obtain all the indicator values in Step 5 needed for evaluating the 15 validation criteria.

D. Step 5

For five of the validation criteria the client experts did not manage to obtain the indicator values necessary for evaluating the criteria. One reason may be that obtaining all the indicator values required too much effort. The client experts tried to obtain values for 49 out of the 57 indicators remaining after Step 3. Out of the 19 indicators that ended up being used in validation criteria, they managed to obtain values for 13. They may have succeeded for a higher proportion if we had only requested the values for the 19 indicators being used in validation criteria. The reason for requesting all indicator values was that the validation criteria were formulated after the value collection process had been initiated, and before we received the indicator values from the client experts. Thus, we did not know at the time when the value collection process was initiated which indicators we would use to formulate the validation criteria. It would have been better to first identify the indicators needed in the validation criteria, and then ask the client experts to obtain values for those.

Another reason for failing to obtain six of the necessary values may have been that the client experts postponed the task a little too long. This is very likely since we know that many of the indicator values where obtained just before the given deadline. But it can also be the case that the values were not as easily available as first expected. Expert E2 supports the latter: “... for me the process went pretty smoothly. I got answers if there had been done measurements, but I also got feedback like “we have no...”
All three experts interviewed believe that indicator values of high quality were obtained. It is however a bit uncertain whether this was actually the case. We know, for instance, that some of the values obtained were just new expert judgments by other experts. Expert E2 told us that he obtained indicator values by asking other people working at the company: “The hardest part is to find the right person who has the right competence. It was pretty easy to find the answers for those indicators where there were numbers if we found the right person. In our case there were actually two-three persons who answered all.” It is however a bit uncertain how many of the obtained indicator values that were just new expert judgments.

E. Step 6

Two likelihood estimates were changed by the client experts as a result of their validation criteria being falsified. When changing the likelihood estimate of an unwanted incident, its risk level will often change as well, since the risk level depends on the likelihood value and the consequence value of the unwanted incident. A change in the risk level will often have consequences for the type of treatments that are implemented. In our case, however, the risk levels associated with the two unwanted incidents did not change when their likelihood estimates were updated.

In the case of a validation criterion being falsified we can not straightforwardly conclude whether likelihood estimates should be changed or kept as they are. For instance, although we manage to obtain correct indicator values, it may be that the validation criterion does not capture what we believe it does. In the risk analysis we had two cases where the client experts decided not to adjust the likelihood estimates of two unwanted incidents, even though their validation criteria were falsified. In the case of the first incident, the client experts kept the likelihood estimate because the value of the indicator used in the criterion did not represent a typical value. In the case of the second incident, its likelihood estimate was, according to its validation criterion, equal to zero since some technology required for the incident to occur was not in use at the time of analysis. As a consequence of this the incident should have been removed from the threat model. The client experts wanted the threat model to reflect the situation were the technology was in place, and the threat model was therefore not changed. Also, it was no harm in keeping the incident, since it did not result in any unacceptable risks needing treatments, due to a low likelihood estimate.

V. Conclusion

In this paper we have presented experiences from using indicators to validate likelihood estimates based on expert judgments in a security risk analysis conducted in 2010. The use of indicators brought forward new information resulting in two out of 28 likelihood estimates being changed.

We also identified some challenges that need to be addressed in order to get the most out of indicator-based validation. First, it is challenging to identify indicators for which it is feasible to obtain values within the available time and resources for the analysis. For a number of the indicators identified, their values were not obtainable within the client’s organization. By having some knowledge on the kinds of historical data that are available within the organization and whose responsible for the different kinds of data, it should be easier to both identify indicators and obtaining their values. Unfortunately, it may be difficult to obtain this knowledge since data is often spread across the organization and since few, if any, have a complete overview of the data available. Second, it is challenging to formulate validation criteria for likelihood estimates in terms of indicators. It is especially difficult to predict how indicator values affect a likelihood estimate when the indicators are only indirectly related to the estimate in question. This will typically be a problem when formulating validation criteria for likelihood estimates of incidents that that are not easily observable. Third, the indicator values obtained from an organization may vary when it comes to correctness. In order to get the most out of the validation, the uncertainty of the values should be taken into account. Moreover, one should strive to reduce uncertainty by using several independent indicators to validate the same estimate.

Acknowledgments

The research on which this paper reports has been carried out within the DIGIT project (180052/S10), funded by the Research Council of Norway, and the MASTER and NESSoS projects, both funded from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreements FP7-216917 and FP7-256980, respectively.

References


The analysis was conducted at the following three levels: Level 1: Identify vulnerabilities with respect to the requirements. Select the vulnerabilities that require further analysis. Level 2: Document possible high-level threat scenarios that may arise if the selected vulnerabilities from Level 1 are exploited. Select the threat scenarios that require further analysis. Level 3: Conduct a detailed analysis of the selected threat scenarios from Level 2.

As seen in Fig. 2, the first four steps of the CORAS method were conducted as part of the Level 1 and 2 analyses, while the four last were conducted during the Level 3 analysis. As part of the Level 3 analysis we validated likelihood estimates, obtained from expert judgments, by the use of indicators. In the following we focus our attention on the steps of the CORAS method relevant for the Level 3 analysis and their relation to the steps of the estimation and validation process. For more details on the different steps of the CORAS method, see [6].

The relation between the Level 3 analysis and the estimation and validation process is depicted in Fig. 2. In the following we refer to Step X of the CORAS method and Step Y of the estimation and validation process as Step CMX and Step EVPY, respectively.

In Step CM5 we used the selected threat scenarios from the Level 2 analysis as a starting point for identifying and documenting unwanted incidents as well as threats, vulnerabilities, and the threat scenarios leading up to the unwanted incidents, by the use of CORAS threat diagrams. In Step CM6 the client experts estimated likelihoods for the unwanted incidents identified in Step CM5, and their consequences with respect to different assets. All the estimates were based on expert judgments. Thus, Step EVP1 was conducted as part of this step.

After conducting Step EVP1, the analysis team identified indicators in Step EVP2 for validating likelihood estimates assigned to the different unwanted incidents. The indicators identified by the analysis team were revised during a meeting with the client experts in Step EVP3. During this meeting some indicators were rejected, some were subject to minor modifications, and some new indicators were identified.

In Step EVP4 the analysis team formulated validation criteria for validating likelihood estimates, based on expert judgments, in terms of the identified indicators. As previously explained, each criterion specifies the expected values of the indicators related to the likelihood estimate in question. Here, each criterion makes a prediction about the value of a set of indicators under the assumption that the likelihood estimate in question is correct.

Indicator values were obtained by the client experts in Step EVP5. In Step CM7 we estimated risk levels for the different risks. As part of this step we conducted Step EVP6 where we validated likelihood estimates, based on expert judgments, by the use of the validation criteria. Some of the likelihood estimates were adjusted as a result of their validation criteria not being fulfilled. In Step CM8 we identified treatment options for the risks classified as unacceptable in Step CM7.

Table II shows estimates for the number of hours that were spent on each level of the analysis, as well as estimates for the number of hours that were spent on the steps of the Level 3 analysis and the estimation and validation process. The estimates are based on the analysis team's own notes from the analysis. For the client experts we only have numbers for meetings. Thus, we do not know how much time they spent between meetings, but we know that it was considerably less than the time spent by the analysis team. In total, an estimated number of 400 hours were spent on the full analysis (not including writing the final report) by the analysis team. About 60% of these hours were spent on the Level 3 analysis. The table shows that a large amount of the hours spent on the Level 3 analysis were spent in relation to indicators.

As already explained, three domain experts participated on behalf of the client. Table III shows the education of the client experts, as well as their experience within information security and risk analysis.
### Table II

<table>
<thead>
<tr>
<th>Level</th>
<th>Analysis team</th>
<th>Client experts (meetings only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>75</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>75</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>250</td>
<td>65</td>
</tr>
<tr>
<td>Total</td>
<td>400</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step(s)</th>
<th>Analysis team</th>
<th>Client experts (meetings only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM5</td>
<td>35</td>
<td>15</td>
</tr>
<tr>
<td>CM6 / EVP1</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>EVP2</td>
<td>50</td>
<td>-</td>
</tr>
<tr>
<td>EVP3</td>
<td>20</td>
<td>15</td>
</tr>
<tr>
<td>EVP4</td>
<td>45</td>
<td>-</td>
</tr>
<tr>
<td>EVP5</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CM7 / EVP6</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>CM8</td>
<td>30</td>
<td>10</td>
</tr>
<tr>
<td>Total</td>
<td>250</td>
<td>65</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>Expert</th>
<th>Education and experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>E1</td>
<td>Degree equivalent to Master of Science. Ten years of experience in information security. Ten years of experience in risk analysis.</td>
</tr>
<tr>
<td>E2</td>
<td>Degree equivalent to Master of Science. Ten years of experience in information security. Four years of experience in risk analysis.</td>
</tr>
<tr>
<td>E3</td>
<td>Degree equivalent to Master of Science. Ten years of experience in information security. Eight years of experience in risk analysis.</td>
</tr>
</tbody>
</table>

### APPENDIX B.

**DATA COLLECTED FROM CASE**

Table I only presents some of the data collected for the different steps of the estimation and validation process. In this appendix we present all the data collected. We provide explanations for all the different data collected, even the data explained in Section III. The data is presented in Table IV.

Some of the rows in Table IV refer to rows in Table I to make it easier for the reader to understand which of the data items that are found in both tables.

**A. Data for Step 1**

The risk analysis resulted in 28 (ID 1.1 in Table IV) likelihood estimates based on expert judgments, where each estimate is associated with an unwanted incident.

**B. Data for Step 2**

In Step 2 the analysis team identified one or more indicators for each of the 28 (2.3) likelihood estimates. In total, 81 (2.1) indicators were proposed by the analysis team, of which 68 (2.2) were unique\(^2\). Even though it has not

\(^2\)A number of the indicators were used for more than one likelihood estimate.
<table>
<thead>
<tr>
<th>Step 1: Expert judgments</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td></td>
<td>Number of likelihood estimates based on expert judgments after Step 1.</td>
<td>28</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2: Identification of indicators</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td></td>
<td>Total number of indicators after Step 2.</td>
<td>81</td>
<td>-</td>
</tr>
<tr>
<td>2.2</td>
<td></td>
<td>Number of unique indicators after Step 2.</td>
<td>68</td>
<td>2</td>
</tr>
<tr>
<td>2.3</td>
<td></td>
<td>Number of likelihood estimates with at least one indicator after Step 2.</td>
<td>28</td>
<td>3</td>
</tr>
<tr>
<td>2.4</td>
<td></td>
<td>Number of likelihood estimates after Step 2 with indicators that are directly linked to the estimates.</td>
<td>19</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: Indicator revision</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td></td>
<td>Total number of indicators after Step 3.</td>
<td>68</td>
<td>4</td>
</tr>
<tr>
<td>3.2</td>
<td></td>
<td>Number of unique indicators after Step 3.</td>
<td>57</td>
<td>5</td>
</tr>
<tr>
<td>3.3</td>
<td></td>
<td>Number of likelihood estimates with at least one indicator after Step 3.</td>
<td>25</td>
<td>5</td>
</tr>
<tr>
<td>3.4</td>
<td></td>
<td>Number of likelihood estimates with zero indicators after Step 3.</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>3.5</td>
<td></td>
<td>Total number of new indicators added during Step 3.</td>
<td>3</td>
<td>-</td>
</tr>
<tr>
<td>3.6</td>
<td></td>
<td>Number of unique new indicators added during Step 3.</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>3.7</td>
<td></td>
<td>Total number of indicators rejected during Step 3.</td>
<td>16</td>
<td>-</td>
</tr>
<tr>
<td>3.8</td>
<td></td>
<td>Number of unique indicators rejected during Step 3.</td>
<td>13</td>
<td>-</td>
</tr>
<tr>
<td>3.9</td>
<td></td>
<td>Number of likelihood estimates after Step 3 with indicators that are directly linked to the estimates.</td>
<td>11</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 4: Identification of validation criteria</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td></td>
<td>Total number of indicators used to formulate the validation criteria after Step 4.</td>
<td>20</td>
<td>-</td>
</tr>
<tr>
<td>4.2</td>
<td></td>
<td>Number of unique indicators used to formulate the validation criteria after Step 4.</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>Number of likelihood estimates with a validation criterion after Step 4.</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>4.4</td>
<td></td>
<td>Number of likelihood estimates with a validation criterion after Step 4, where indicators that are directly linked to the estimates are used in the criteria.</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>4.5</td>
<td></td>
<td>Number of likelihood estimates without a validation criterion after Step 4, due to that the validation of the likelihood estimate was given a low priority.</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>4.6</td>
<td></td>
<td>Number of likelihood estimates without a validation criterion after Step 4, due to the analysis team not being able to formulate validation criteria based on the indicators associated with the likelihood estimates.</td>
<td>9</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 5: Calculation of indicator values</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td></td>
<td>Number of unique indicators used to formulate validation criteria for which the client experts obtained values after Step 5.</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>5.2</td>
<td></td>
<td>Number of likelihood estimates for which validation criteria could be evaluated after Step 5.</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>5.3</td>
<td></td>
<td>Number of likelihood estimates for which validation criteria could not be evaluated after Step 5.</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>5.4</td>
<td></td>
<td>Number of likelihood estimates for which validation criteria could be evaluated after Step 5, where indicators that are directly linked to the estimates are used in the criteria.</td>
<td>8</td>
<td>-</td>
</tr>
<tr>
<td>5.5</td>
<td></td>
<td>Number of unique indicators for which the client experts tried to obtain values after Step 5.</td>
<td>49</td>
<td>-</td>
</tr>
<tr>
<td>5.6</td>
<td></td>
<td>Number of unique indicators for which the client experts managed to obtain values after Step 5.</td>
<td>37</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 6: Revision of expert judgments</th>
<th>ID</th>
<th>Definition</th>
<th>Value</th>
<th>Row no. Table I</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td></td>
<td>Number of likelihood estimates with a fulfilled validation criterion after Step 6.</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>6.2</td>
<td></td>
<td>Number of likelihood estimates with a not fulfilled validation criterion after Step 6.</td>
<td>4</td>
<td>11</td>
</tr>
<tr>
<td>6.3</td>
<td></td>
<td>Number of likelihood estimates with a validation criterion where it was undecided whether the criterion is fulfilled or not after Step 6.</td>
<td>2</td>
<td>12</td>
</tr>
<tr>
<td>6.4</td>
<td></td>
<td>Number of likelihood estimates with a not fulfilled validation criterion for which the likelihood estimates were adjusted after Step 6.</td>
<td>2</td>
<td>13</td>
</tr>
</tbody>
</table>
been stated explicitly, we only focus on unique indicators in 
Table 1. In Table IV we distinguish between the number of 
unique indicators and the total number of indicators.

For 19 (2.4) likelihood estimates, the analysis team pro-
posed indicators that are directly linked to the estimates. 
The distinction between indirectly and directly linked indicators 
is explained in Section IV-C. As explained in Section IV-C, 
indicators that are directly linked to a likelihood estimate 
have often an advantage over the indirect ones when it comes 
to validating the likelihood estimate.

C. Data for Step 3

The indicator proposals were revised in Step 3 during a 
meeting with the client experts. After the revision had been 
conducted, 25 (3.3) out of the 28 likelihood estimates were 
associated with one or more indicators. In total this left us 
with 68 (3.1) indicators, of which 57 (3.2) were unique. 
During the revision meeting, three (3.5) new indicators, of 
which two (3.6) were unique, were added. 16 (3.7) of the 
proposed indicators were rejected during the same meeting, 
mostly due to their values not being obtainable within the 
client’s organization. Of these indicators, 13 (3.8) were 
unique. The result of rejecting these indicators was that three 
(3.4) likelihood estimates were left without indicators.

As we can see from the data of Step 2 and 3 in Table IV, 
almost 20% of the proposed indicators were rejected. Also, 
a number of indicators that are directly linked to the estimates 
of which they were seen as relevant, were rejected in Step 3. 
The result of this was that only 11 (3.9) out of 19 (2.5) 
likelihood estimates were left with indicators that are directly 
linked to the estimates after Step 3 had been conducted.

D. Data for Step 4

Validation criteria were formulated by the analysis team, 
in Step 4 of the estimation and validation process, for 15 
(4.3) out of the 25 likelihood estimates with indicators, of 
which eight (4.4) of the validation criteria, or more than 
50%, use indicators that are directly linked to the estimates. 
For the remaining ten likelihood estimates, validation criteria 
were not formulated. One (4.5) of these estimates was not 
assigned a criterion due to that the validation was given 
a low priority by the client experts. For the other nine 
(4.6) likelihood estimates, the analysis team was not able 
to formulate validation criteria based on their indicators, 
although the indicators were considered to provide relevant 
information in terms of validating the estimates. To formu-
late the validation criteria the analysis team used 20 (4.1) 
indicators, of which 19 (4.2) are unique. Thus, only 33% of 
the unique indicators left after the revision meeting in Step 3 
was used for formulating validation criteria.

E. Data for Step 5

In Step 5, values were obtained for a number of the 
indicators remaining after the revision meeting in Step 3. 
Note that the analysis team did not restrict the client ex-
erts’ collection of indicator values to the 19 (4.2) unique 
indicators used in the validation criteria. The validation 
criteria were formulated after the collection process had been 
initiated, and before we received the indicator values from 
the client experts. Thus, we did not know at the time the 
collection process was initiated which indicators we would 
actually be able to use in the validation criteria.

The client experts tried to obtain values for 49 (5.5) 
out of the 57 (3.2) unique indicators that remained after 
Step 3 had been conducted. Of these 49 unique indicators, 
the experts managed to obtain values for 37 (5.6). Thus, 
the experts managed to obtain 75% of the values. 13 (5.1) of 
the obtained values belong to indicators used in the validation 
criteria. This means that six of the values needed in the 
evaluation of the validation criteria were not obtained. The 
result was that five (5.3) of the likelihood estimates with 
validation criteria could not be evaluated. This meant that 
we could only validate ten (5.2) of the likelihood estimates, 
of which eight (5.4) use indicators in their validation criteria 
that are directly linked to the estimates.

F. Data for Step 6

Ten likelihood estimates were validated in Step 6. For 
four (6.1) of the estimates the criteria were fulfilled, for two 
(6.3) we could not say whether the criteria were fulfilled or 
ot because the indicator values used in the criteria were 
too uncertain, while for the remaining four (6.2) the criteria 
were not fulfilled. For two (6.4) out of these four likelihood 
estimates, the client experts decided to adjust the estimates, 
while the two other estimates remained unchanged.

APPENDIX C.
THEMATIC ANALYSIS

A. Procedure for collecting data

We conducted a semi-structured interview with each of 
the three client experts that participated in the security risk 
analysis. In these interviews the experts were asked open-
ended questions related to estimation of likelihood values 
based on expert judgments and the use of indicators to 
validate these. Each question had a number of prompts 
(follow-up questions), that were asked if the client experts 
did not answer them as part of the open-ended question. All 
the interviews were recorded and conducted in Norwegian. 
The Norwegian Social Science Data Services\(^4\) has been 
notified about the interviews, as a result of personal data 
being collected, recorded, and stored.

\(^3\)The likelihood estimate was associated with an unwanted incident. For 
the specific incident to occur, some technology needed to be used, which 
was not yet implemented at the time of analysis. Thus, the client considered 
the likelihood estimate for this incident to be less important.

\(^4\)See \url{http://www.nsd.uib.no/nsd/english/index.html} and \url{http://www.nsd. 
uib.no/nsd/english/pvo.html} for more information.
The two analysts had different roles in the interviews. One analyst had the role as the interviewer, while the other acted as an observer. Almost all of the interaction was between the interviewer and the interviewee. The main tasks of the observer were to administer the recordings of the interviews, to take additional notes if necessary, and to make sure that the necessary information was collected. The observer only interacted with the interviewee in cases where he felt that additional answers were needed.

Each interview was introduced by the interviewer explaining the purpose of the interview and the terms under which the interview would be conducted. The interviewees were informed that the purpose of the interview was to collect empirical data to be used in an evaluation report on the security risk analysis they had participated in. They were also told about our intention to publish this report, that they would appear anonymous in this report, and that the report would not contain any confidential information. The interviewees were also informed that they could withdraw from the interview at any time, without giving any form of explanation, and that the interview would be recorded and that a non-confidential transcript of each recorded interview would be stored as background material for the evaluation report for a period of ten years. Before starting each interview, the interviewer asked the interviewee whether he/she accepted the terms, including that the interview would be recorded.

An interview guideline, given below, was used for conducting the interviews. Only the interviewer and the observer had access to the guideline. In the guideline, $Q$ stands for open-ended question, while $P$ stands for prompt. Also, the guideline states the topic addressed by each open-ended question, and whether a handout was provided to the interviewees for the purpose of triggering their memory. The interviewees were only allowed to have a short look at the handouts, to ensure that they did not become too focused on specific details.

- **Q1:** [Topic: General question about estimation.] [Handout: Printout of all the threat diagrams.]
  
  As you know, a risk level depends on the frequency and consequence of an unwanted incident. A part of the risk analysis focused therefore on estimating the frequency and consequence of the identified incidents. How did you experience that part of the process - that is, the frequency and consequence estimation?

- **P1:** To what extent do you think we generally came up with correct estimates?
- **P2:** Can you say something about possible sources to uncertainty with respect to the correctness of the estimates?
- **P3:** Was there estimates that were particularly difficult or easy to do? If so, which ones and why?
- **P4:** Was there any estimates where you had more confidence in their correctness than others? If so, which ones and why?
- **P5:** What do you think about the approach used for estimation?

- **Q2:** [Topic: About identification of indicators.] [Handout: Overview of unwanted incidents and their final indicators.]
  
  For some of the incidents we identified indicators to support the estimation of their frequencies. How did you experience this sub-process - that is, the identification of relevant indicators?

- **P1:** What worked well/badly?
- **P2:** How difficult was it to identify relevant indicators?
- **P3:** Was there any incidents of which identification of indicators were particularly easy or difficult? If so, why?
- **P4:** Do you think in general that there exists categories of unwanted incidents of which indicator identification is particularly easy or difficult?
- **P5:** We (the analysis team) came up with proposals for indicators that were presented and discussed at a meeting. What difference do you think it would have had if we had asked you to think through whether these were good indicators between two meetings, and possibly discussed them with colleagues?

- **Q3:** [Topic: About obtaining indicator values.] After having identified relevant indicators, the next step was to obtain the indicator values. Can you tell us how you experienced this part of the process?

- **P1:** To what extent was it difficult to obtain indicator values?
- **P2:** Was there any indicators for which it was particularly easy or difficult to obtain values?
- **P3:** To what extent do you trust that the obtained indicator values were correct?
- **P4:** What kind of approach do you think we should use for identifying indicators that it is possible to obtain values for?

- **Q4:** [Topic: From indicator values to frequency estimates.] Indicator values were used to support the frequency
The estimation of frequency and consequence values for unwanted incidents - and thus risk level - is only a part of the whole risk analysis. How important do you think it is that you arrive at good/correct estimates?

- **P1:** Can you give some indication of the amount of resources you think is right to use on this part of the analysis in relation to the rest?

- **P2:** What do you think is reasonable to expect from the documentation in an analysis report regarding the foundation of the estimates?

### B. Procedure for analyzing data

The recorded interviews were transcribed not long after the interviews had been conducted. Based on these transcripts, non-confidential transcripts were created by removing all the text mentioning the system assessed, the risk models obtained, the personnel from the client involved, and the name of the client.

The data set (the three non-confidential transcripts) have been analyzed by the use of a simplified version of thematic analysis [7], which is a method for identifying, analyzing, and reporting patterns (themes) within data. In [8], theme is defined as follows: “A theme captures something important about the data in relation to the research question, and represents some level of patterned response or meaning within the data set.” The data was analyzed with the following overall research question in mind: “To what extent may indicators be used to validate likelihood values obtained from expert judgments in a security risk analysis?”

Thematic analysis offers a lot of flexibility as a method, i.e. that it can be performed in many different ways. What separates one thematic analysis from another are, among other things: whether the themes focus on the whole data set or only parts of it; whether themes are identified by an inductive or theoretical approach; whether themes are identified at the semantic or latent level; and the research epistemology used, i.e. what you can say about your data and how you theorize meaning.

In our thematic analysis we do not try to give a rich thematic description of the entire data set. We rather focus on giving a detailed account of a group of themes within the data, which are of relevance to the overall research question. We can therefore say that our themes are identified in a theoretical way. Furthermore, we identify themes at the semantic level. This means that themes are identified within the explicit or semantic meanings of the data. We do not try to go beyond what the interviewee said in the interview. In the case of research epistemology we have followed an essentialist/realist approach. The research epistemology states what we can say about the data and it informs how we theorize meaning. In the essentialist/realist perspective, meaning and experience inhere with individuals.

Inspired by the process for thematic analysis outlined in [8], our thematic analysis was conducted as follows:

### Q5: [Topic: Indicator-based estimates versus estimates based solely on expert judgments.]

Some of the frequency estimates were based on a combination of indicators and expert judgments, while others were based solely on expert judgments. To what extent do you think this difference is important for the quality of the estimates?

[Make sure to get an explanation.]

- **P1:** Does the difference have any relevance for the correctness of the estimates? Why/why not?

- **P2:** Does the difference have any impact on yours or others’ confidence in the correctness of the estimates? Why/why not?

### Q6: [Topic: Possible benefit of indicator values as documentation.]

In the analysis report, indicator values can be included as documentation for the foundation for frequency (and possibly consequence) estimates. To what extent do you think this will affect the quality of the report? In what way could it be affected?

- **P1:** Imagine that you are going to read a report from an analysis you have not participated in. Do you think it will affect your confidence in the correctness of the estimates if the report documents the indicators that were used for the various estimates, and their values?

- **P2:** Imagine that you are participating in an analysis and contribute to making the estimates. To what extent do you think it is important to document the foundation for the estimates in terms of indicator values?

### Q7: [Topic: About the importance of good estimates.]

The estimation of frequency and consequence values...
1) We familiarized ourselves with the data by transcribing the recording interviews. From these transcripts, we created non-confidential transcripts.

2) We performed an initial coding of data extracts found in the different data items (non-confidential transcripts). The coding was performed for identifying interesting features of the data. For a number of the data extracts, more than one code was assigned. The coding also resulted in coded data extracts of varying sizes. Some of the data extracts refer to only a part of a sentence, while others refer to a number of sentences.

3) We identified themes based on the initial coding of the data, and assign the different data extracts to these themes. Some data extracts were assigned to more than one theme, while others were not assigned to any theme at all.

C. Results from the thematic analysis

The thematic analysis resulted in 5 themes. These are shown in Table V. Each theme represents a topic that the interviewees talked about. The data extracts assigned to the different themes have been used for discussing the data collected from the case.

The interview data on which the results are based comprise 89 data extracts. These data extracts differ much in size and for some extracts, more than one theme is associated. Their distribution over the different themes is shown in Table V.

APPENDIX D.
THREATS TO VALIDITY

In this appendix we present threats to the validity of our conclusions. The main threat to the validity of the conclusions is, of course, the fact that the investigation was carried out in a single risk analysis case. We should therefore be careful when generalizing the results. It is possible that a different case would have yielded completely different results. However, we do not believe this is the case, for a number of reasons. First, based on experiences from our previously conducted risk analyses we believe that the analysis is representative with respect to target and scope. Second, based on the same experiences we believe that the client experts are representative with respect to experience and background, as well as are their behavior and roles in the analysis.

Based on the above considerations, in the following we present what we believe are the most significant threats to the validity of our conclusions:

- This was the first time the analysis team used indicators in an industrial setting. On the other hand, the analysis team has some experience with indicators from an academic setting. It is therefore possible that the analysis team has more knowledge about indicators than most other analysts that are first and foremost skilled in the area of risk analysis.
- In a risk analysis we most often use experts with different areas of expertise. It is therefore not unthinkable that experts having indicators and security metrics as their field of expertise may participate in risk analyses involving the use of indicators. Based on this we cannot say that the client experts are representative when it comes to understanding the relation between indicator values and likelihood estimates, and when it comes to knowledge about indicators.
- In our particular case the client experts knew quite often who to ask when they needed data. This does not need to be the case in other organizations. It may also be the case that the client organization addressed in this paper is different from other organizations when it comes to logging of risk and security related information. Based on this we cannot say that the client organization is representative with respect to the amount of risk and security related information logged and with respect to having an overview of this data, i.e. knowing what kind of data that is logged and knowing whose in charge for the different data.
- This was the first time the analysis team documented the experiences of the participating experts by the use of semi-structured interviews. Since the analysts are not skilled in interviewing, it is possible that a more professional interviewer would have asked different questions and could perhaps have extracted more interesting information from the interviewees, where as this information could have lead to a different understanding and interpretation of the data collected during the analysis. However, we do not believe that this would have resulted in big differences, since the data collected during the analysis can mainly speak for itself.

### Table V

<table>
<thead>
<tr>
<th>Theme – Description</th>
<th>No. of data extracts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theme 1: Estimation – Reflections about estimation of likelihood and consequence.</td>
<td>18</td>
</tr>
<tr>
<td>Theme 2: Indicator identification – Reflections about the identification of indicators.</td>
<td>24</td>
</tr>
<tr>
<td>Theme 3: Obtaining indicator values – Reflections about obtaining indicator values.</td>
<td>11</td>
</tr>
<tr>
<td>Theme 4: Validation – Reflections about validation of likelihood estimates by the use of indicators.</td>
<td>14</td>
</tr>
<tr>
<td>Theme 5: Quality – Reflections about the quality of expert judgments, likelihood estimates, and indicators.</td>
<td>24</td>
</tr>
</tbody>
</table>