Utrecht University

MASTER THESIS

Reasoning on Architecture Design

Author:
ing. P.S.H. de Jong

Supervisor:
dr. ir. J.M.E.M. van der Werf

A thesis submitted in fulfilment of the requirements for the degree of Master of Science

in

Information Science, Business Informatics

Department of Information and Computing Sciences

Summer
2017

Additional information

UNIVERSITY:	Utrecht University
EDUCATION PROGRAMME:	Master's in Business Informatics
DEPARTMENT:	Information and Computing Sciences
FACULTY:	Faculty of Science
ADDRESS:	Princetonplein 5, 3584CC Utrecht
HOST:	Sogeti Netherlands B.V.
BUSINESS LINE:	Information Management
ORGANIZATIONAL PORTFOLIO:	Technology Transformation
ADDRESS:	Lange Dreef 17, 4131 NJ Vianen

AUTHOR:	ing. P.S.H. de Jong
STUDENT NUMBER:	5657652
E-MAIL:	p.s.h.dejong@students.uu.nl
MOBILE PHONE:	+31629409111
FIRST SUPERVISOR:	dr. ir. J.M.E.M. van der Werf
E-MAIL:	j.m.e.m.vanderwerf@uu.nl
DEPARTMENT:	Information and Computing Sciences
SECOND SUPERVISOR:	dr. F.J. Bex
E-MAIL:	f.j.bex@uu.nl
DEPARTMENT:	Information and Computing Sciences
EXTERNAL SUPERVISOR:	dr. ir. M. van Steenbergen
E-MAIL:	marlies.van.steenbergen@sogeti.com
DEPARTMENT:	Information Management / Architecture

ABSTRACT

As systems are growing ever more in terms of complexity, architecture exists to manage this. As this complexity increases, the need for effective architecture becomes vital. Effective architecture, among other things, is reached with proper architecture design and -description. An architecture can be seen as the culmination of design decisions, which have great impact on the resulting architecture.

Without proper explicit rationale however, architecture degradation occurs. Architectures become increasingly brittle over time if documentation of design decisions are not kept. Architects leave organizations, switch roles or switch projects. The knowledge that is produced during architecture design is lessened if left implicit and time elapses. This results is a net loss of organizational knowledge and the corresponding system to be increasingly fragile. Architecture changes, maintenance, communication and architecture reuse all become more and more difficult and resource intensive to perform. Also, naturalistic decisions in the design phase may cause flaws that surface during implementation. Proper design decisions and architectural description may decrease the risk of these system flaws.

Design reasoning stimulates effective architecture documentation through rationale capture. Architects are found to produce, on average, better architectures by explicitly reasoning about their design decisions. Currently, budget constraints or limited industry support constrain the adoption of explicit reasoning and rationale capture. These premises are, somehow, seen as a separate process from architecture design. This distinction is problematic since the reasoning and rationale process is viewed as an intrusive factor without short term benefits. This reluctance will impede the development of design reasoning if no new steps towards awareness and adoption are made.

The key goal of this thesis is to effectively embed design reasoning principles into architecture design and measure its effects. During this thesis the Rationale Capture Cycle (RCC) is designed, a reasoning structure model that guides design reasoning and stimulates rationale capture. The RCC is designed with the current limitations and challenges in mind, and tries to overcome them. The model is assembled through a method engineering process and is validated with practicing architects from Sogeti Netherlands B.V.

Two experiments are designed. The first experiment attempts to embed the RCC into architecture design activities to validate the model and its effect. A case is designed where 10 practicing architects are to design an entire architecture. The main goal for the case is to provide architects with a scenario that mirrors an average project, yet be challenging enough for experienced architects. The second experiment manipulates the extent in which design rationale is present during a similar case to measure its effects. The theory is that having access to rationale allows for easier architecture activities. This way, both rationale itself and an instrument that stimulates rationale are tested.

The architects are split into two research groups, one of those uses the RCC. The effect of the RCC is then observed through 3 different measures. First, the relative design quality of the architecture that they produce is measured. In order to measure this, points are given to each architecture that is produced by the participant. All participants are given 3 random and anonymous solutions by their peers. Each participant has to evaluate these solutions on the relative quality of the architecture model and its accompanying documentation. Second, all rationale documentation that is produced is coded and analysed. A coding scheme is designed to find and distinguish rationale types that are captured by the participant. Third, the participant is asked to fill in a survey to provide insight into their design experience. These three measures are compared between the research groups and the differences analysed.

The data does not undeniably confirm or deny the inclusion of rationale has major effects during the second experiment session. The data was, however, able to demonstrate rationale capture can be stimulated by using a rationale structure model to support reasoning. The RCC was found to be very influential. A positive effect was observed through expert evaluations, the quality of rationale documentation and the design experience of the architect when the Rationale Capture Cycle is utilized. Various different tests and results cohesively point towards the same result as the degree of agreement between the different tests are high. Therefore, reasoning can be stimulated by using a reasoning structure model during architecture design. This way, design reasoning principles can be embedded into the architecture process and its beneficial effects measured and demonstrated.

ACKNOWLEDGEMENTS

Many different parties supported the execution of this research project and are therefore responsible for its results and outcomes. Without them, this thesis would not exist as it does today.

First and foremost, Jan Martijn van der Werf deserves acknowledgement for the constant supervision and support. You brought interesting ideas and new takes on the topic which shaped and influenced the thesis greatly. You kept the thesis project rigorous, yet maintained a light and fun attitude. Thank you for your enthusiastic guidance and effort these last 8 months!

Thanks to Floris Bex for your supervision and input as second supervisor, and Matthieu Brinkhuis for lending your time and mind during intricate and complex problems.

A special thanks to Marlies van Steenbergen, whom fulfilled the role of stakeholder, yet maintained very involved throughout the project regarding the matter itself. Your enthusiastic involvement in both the creative process and scientific method contributed greatly to the success.

Finally, many practicing architects from Sogeti and other organizations contributed greatly, both in terms of advice and input. Thank you for valuable time and effort.

CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

Complex systems occur throughout every aspect of society. These systems are intricate in nature and are moulded by their boundaries, environment and purpose. Welfare and retirement entitlements are examples of complex socio-economic systems and form the bedrock of many countries' government principle. Other examples include the Earth's global climate, a 20-story building, an organism, an ecosystem and the human brain. The intricacy of these systems stem from the large amounts of related elements. These elements can simply interact, or be mutually dependent. Together, these parts form a complex whole whose purpose is expressed in its functioning.

Today's information landscape is ever so dependent of highly complex systems. Enterprises globally rely on intricate systems to fulfil certain purposes related to that enterprise's objective. Software systems, for example, support nearly all information-based processes, bearing the responsibility of their execution. The definition of these complex systems do not solely contain software systems, however. In the context of an enterprise, a complex system may comprise of many socio-technical systems as well, including people, information, processes and technologies.

Architecture refers to the standard practice for managing this complexity. The practice can be defined as the fundamental concepts or properties of a system in its environment, embodied in its elements, relationships, and in the principles of its design and evolution (ISO / IEC / IEEE, 2011). In other words, architecture entails a holistic perspective of a system in its context. In other words, architecture is the practice of design, description, structure, relation and cohesion. Architecture can exist in many different shapes and forms. Architecture can be used to describe a building and its elements, or software and its information needs. Enterprise Architecture, for example, is a specific domain of architecture and provides guidelines on how to approach managing intricate and evolving systems in order to align them with an enterprise's strategy and objectives (Lankhorst, 2009). The practitioners of architecture, architects, are tasked with performing analysis so that systems can successfully contribute to an organization's execution of strategy (Rozanski & Woods, 2011). Architects do so by designing architectural models. Architectural models provide a high-level abstraction of a system and it allows architects to visualize an architecture. These models are designed in order to easily present and communicate an architectural design, so that it may be discussed with stakeholders and demonstrate the designs viability.

An architectural design of a system is comprised of the decisions an architect makes during the design phase. Deciding not to implement the newest hardware due to security risks, even though it has network stability benefits, is a good example of a trade-off choice with a risk factor. Architects constantly have to reason and provide argumentation in order to make these intricate decisions. These problems add to the complicated nature of architectural design, especially when no standard solutions or best practices exist.

These choices have a great impact on the resulting system, making the design stage of paramount importance. Architectures require continuous maintenance, through changes, updates, improvements or integration. In many cases, architectures are worked on by architects who are not the original designer. Architects switch roles, leave companies, delegate, retire or simply cooperate with other architects. In order for new architects to effectively carry out their responsibilities, a comprehensive knowledge of that specific architecture is required. This includes which specific architectural design decisions were made, what alternatives were considered and what justification exists for that decision. An important characteristic of architecture is the dimension of time. Architectures are not built once and forgotten. Architectures have to be maintained and are required to facilitate change (Lankhorst, 2009). Architectures are of temporary nature and might change based on evolving drivers for business. These drivers might be new technological opportunities (cloud, big data, Internet of Things (IoT)), hardware upgrades, software updates or corporate restructuring.

This dynamic nature of architecture drives the need for good documentation. Without documentation, new architects are not aware of what decisions have been made, what changes drove these decisions and what alternatives have been considered and discarded. This architecture-specific information results in valuable knowledge of the system. If this knowledge is left implicit however, organizational knowledge regarding the architecture decreases over time. This knowledge is completely lost when the original architect no longer maintains the architecture due to changing personnel. This causes the system to be increasingly fragile as time elapses (Perry & Wolf, 1992). The previously mentioned reasoning and justification for design decisions is called *design rationale* (Lee, 1997). To help an architect produce design rationale *design reasoning* can be employed, which produces the rationale. Design reasoning depends on logical and rational thinking to support arguments and come to a design decision that satisfies the system's requirements and justifies the choice. It helps architects document and capture the reasons for design decisions, resulting in valuable knowledge of an architecture and thus, valuable knowledge of the corresponding system. Tang, Tran, Han & van Vliet (2008) suggest that architects produce, on average, better designs by explicitly reasoning about their design decisions. Their research also suggests that by providing them with a systematic approach, design reasoning principles are applied more effectively.

Architecture design problems can be seen as wicked problems. Rittel & Webber (1973) define wicked problems as unique issues that have no standard solution and are nearly impossible to completely and totally 'fix'. These problems are not thoroughly understood until after their solutions are found. This is due to their unique nature, where these problems often have no given solutions that can be assessed as right or wrong. In order to solve these intricate problems, architects have to make careful considerations and decisions during the architecture design process. In reality, architects often rely heavily on intuition and experience to solve these wicked problems. This subjectivity in the design process heavily influences the architecture design itself, and thus, the resulting system.

In this thesis, insight is gained into how architects can consistently employ design reasoning. This is beneficial because inexperienced architects may consistently provide better quality designs since the experience gap is lessened. This also makes it easier for architects to understand a design if they are not the original designer (Tang, Babar, Gorton & Han, 2006). A survey done by Tang et al. (2006) finds that architects strongly agree on the idea that they cannot understand a design without its rationale, especially if they are not the original designer. Also, insufficient attention to architecture design can cause flaws in the resulting system (Tang, Jin & Han, 2007), making them more difficult and resource intensive to maintain. Utilizing rational and logical decision making, and capturing rationale or justifications, may decrease the risk of these system flaws and benefit architecture design in different ways.

1.1 Thesis Outline

Chapter 1 will give an introduction for the research project that provides necessary context for the subsequent chapters. Also, the thesis' outline is presented.

Chapter 2 describes the problem statement by distinguishing three main problem areas and summarizing them into a main issue statement.

Chapter 3 defines the key goals and objectives of this research project, logically formulated from the problem statement. It also discusses the research questions that shape the project and how to measure these questions. The chapter closes with defining the project constraints.

Chapter 4 outlines the research design. The chapter will provide an overview of how the research will be set up and how the different scientific methods relate.

Chapter 5 discusses the relevant literature and analyses related scientific work. An overview of relevant material will be summarized to answer the research questions addressed by theory.

Chapter 6 elaborates on the process of designing and creating the artefact. It also outlines the various pilot phases and feedback stages.

Chapter 7 details how the empirical experiments were designed and how the process took place. It also outlines the various hypotheses, metrics and instruments for data gathering.

Chapter 8 provides the results of the various performed tests and constructs views to demonstrate interesting data and notable results.

Chapter 0 aggregates and analyses the results from chapter 9 by identifying interesting patterns, significance testing and meaning implication.

Chapter 10 concludes the research by logically inferencing from the observed data. The chapter also discusses experienced limitations and future research.

Chapter 11 provides a list of referenced work that was utilized to produce the thesis.

The appendices represent all other material produced and analysed that did not fit into the thesis itself. The relevant chapters contain summarized answers and refer to the appendices for elaborated material. This includes theory, literature and analysis work.

> *''Systems design grows more complex every day, often consisting of problems that are yet to be fully understood. Effective architecture is required to combat this growing complexity and effective architecture demands thorough documentation.''*

2 PROBLEM STATEMENT

2.1 Naturalistic Decision Making

Although the industry recognizes there is a growing need for documenting design rationale (Bass et al., 2003; Bosch, 2004), architects do not always use design reasoning for the decisions they make during this process. In other words, architects often do not provide reasons or justifications for intricate design decisions (Tang & van Vliet, 2009). On the contrary, decisions are made ad-hoc and are based on the architect's own instinct and experience. This is also known as *naturalistic decision making* (Zsambok & Klein, 2014); people using their individual experience and expertise as their main deciding factor. This approach to decision making is subject to bias, since the quality of design decisions are heavily dependent on individual expertise. This can cause flawed decision-making, especially when the architect is not familiar with the system domain (Tang et al., 2008). The absence of design reasoning can, especially if inexperienced, cause the architect to make uninformed decisions. These decisions can cause errors in the architecture, which results in design flaws in the system. When these flaws surface in a later stage, changes to the system have to be made. At this stage, changes are more difficult to implement considering the impact they may have on interdependent systems. Therefore, this process also becomes more costly. The difficulty of this process is amplified when the original architect is also no longer involved.

Researchers have suggested a theory as to why this bias in reasoning and decision making occurs. According to Kahneman & Frederick (2002) and Evans (2003) there are two distinct levels of processing information, namely through system 1 and system 2. System 1 allows for unconscious and quick cognitive processes and system 2 supports conscious and slow cognitive processes. The former is suitable for simple and quick decision making whilst system 2 is suited for intricate and deliberate reasoning, see the following table.

System 1	System 2
Fast	Slow
Unconscious	Conscious
Automatic	Effortful
Everyday	Intricate
Biased	Rational

Table 1. Dual-system cognition (Kahneman & Frederick, 2002)

When designing an architecture, architects use both systems to make their design decisions. System 1 is utilized for routine decisions that do not require a lot of thought. System 2 is used when an architect faces an intricate challenge and requires elaborate thought. The full devotion to a single system is problematic and is counter effective. Solely utilizing system 1 limits your ability to think intelligently and judge rationally, albeit quick and efficient. Solely using system 2 is far too inefficient, forcing you to consciously reason on simple and quick decisions. The challenge lies in quickly determining when to use which system when facing a design problem and to use system 2 for rational decision making as much as possible without being a hassle when facing simple issues. Although intuitive processes (system 1) are handy in everyday decision making, rational judgment should be the leading factor in strategic decision making. Khatri & Ng (2000) found that intuitive processes, whilst used too often, are negatively related to organizational performance in a long-term, stable environment. This decrease in organizational performance might stem from the first system's tendency to be biased. This phenomenon is not new as complete objectivity is unobtainable.

People always view and process information through our subjective reality and judge it as such. This flawed perception is called cognitive bias (Kahneman & Tversky, 1973), and basically refers to the systematic deviation from rational judgment. This deviation can be caused by people, environments, situations or other contextual factors. Also human emotion, or the brain's limited processing power can be the culprit.

Let us consider an example, people who bike to work every day have the preconceived notion their bike tire going flat all the time. In reality, this does not occur often at all when you consider the time spent on the bike. The same can be said for the infamous Bermuda Triangle, where supposedly air and water vehicles mysteriously disappear. In reality, this area does not have a higher count of transport vehicles lost or human deaths when you consider the size of the region and the amount of traffic that goes through it. This is known as availability bias (Tversky & Kahneman, 1975), which basically entails that people overestimate the probability of events associated with memorable or dramatic occurrences. One can imagine the occurrence of a bike tire going flat as emotionally charged, so we tend to overestimate the likelihood of it happening.

Another example of cognitive bias, that may play an important role in architecture design, is confirmation bias. Confirmation bias is our tendency to focus on information that confirms our existing preconception of a situation or concept (Oswald & Grosjean, 2004). An example may be a network engineer concentrating his efforts on designing a centralized database approach because this solution seemed fitting when first presented with the problem. Alternative options, while just as (or even more so) viable, receive less attention or are unjustly disregarded. There are many forms of cognitive bias that may affect decision making (Dietrich, 2010; Milkman, 2008; Tversky & Kahneman, 1975) and could potentially obstruct rational and logical decisions in the architecture design process. Some examples that may have the greatest influence on architecture design are listed below.

Choice-supportive bias: the tendency to feel more positive about one's decisions (Mather & Johnson, 2000). People may disregard flaws in their choices because they tend to look at themselves in a positive manner.

Conservatism: the difficulty of revising your belief when presented with new information (Edwards, 1968). People dislike being proved wrong and therefore tend to favour prior evidence than new contradicting information that has emerged.

Anchoring: the tendency to too heavily depend or value the first piece of information (Epley & Gilovich, 2006). In architecture design, this can prove obstructive since alternative options always need to be considered in later stages.

Outcome bias: the tendency to judge a decision based on its outcome, rather than how the decision was made at the time (Baron & Hershey, 1988). An architect may decree the previous approach to implement a Model-View-Controller (MVC) architecture as the correct decision because the system performed adequately last time. This is dangerous considering the decision itself may have been flawed, but the problem had not yet surfaced in the system.

Recency illusion: the tendency to weigh the most recent information more heavily than older information (Tversky & Kahneman, 1975). This bias stems from the limits of our brain, where we cannot judge and value each piece of information equally throughout our memory.

This list is by no means exhaustive. In fact, the list of cognitive biases that may affect decisional behaviour is far larger. The examples above are listed in order to illustrate the potential flaws architects make in the naturalistic design process so that resulting errors in the system can be minimized.

2.2 Industry Support

According to a survey by Tang et al. (2006) practitioners view design rationale as important, however barriers to the consistent use and documentation of design rationale still exist. Tang mentions that system designers do employ logical reasoning, but there is lack of methodology- and tool support. The survey found that out of 81 respondents 34 declared that are no standards to utilize, and 24 declared that no suitable tool exists. Cumulative, that means 58 respondents of the survey (73.6%) do not provide design rationale due to the lack of industry support. The same study also suggests that architects do acknowledge the need for documenting design rationale, therefore it is of importance that the industry adopts a systematic and structured method for the design reasoning process. The lack of methodology and tool support is in the industry is significant for architecture design is a complex process. Design decisions have to weigh trade-offs or compromises between stakeholders requirements whilst dealing with project resources. Or they have to take technological risks into account whilst still satisfying monetary goals.

Architecture frameworks help combat this complexity of architecture design. Frameworks offer guidance, support and handles for practitioners to apply and use. This is necessary due to the growing complexity of systems and, thus, their architectures. There are few industry standards in EA that acknowledge design reasoning. Among others, the Zachman Framework (1978) or The Open Group Architecture Framework (2011) are notable examples. In this study, current EA framework standards are analysed with regard to their relation to design reasoning. This is important since it allows for a better understanding of the current industry climate with respect to design reasoning. It also highlights the problem of design reasoning support for EA practitioners. This analysis is performed in the literature review, where architecture as a whole is elaborated on.

2.3 Awareness and Adoption

According to the previously mentioned survey by Tang et al. (2006), 49 respondents (60.5%) declared the lack of resources to be a reason why documenting design rationale is not worthwhile. They mention that they either do not have the time or the budget to devote to documenting design rationale. This outlines another problem in the mentality towards documenting design decisions in the current industry. A mentality where the benefits of consistently documenting design rationale do not match up against the resources required for doing so or that they do not fit in the current project constraints. Another contributing factor might be that there is simply not enough awareness and knowledge of what exactly the benefits of documenting design rationale might be and to what extent they contribute.

This issue is recognized among industry practitioners, especially when project boundaries like deadlines and budgetary constraints are present (Conklin & Burgess-Yakemovic, 1991). Architects are also reluctant to devote resources to documenting decisions they opted not to take. Currently, the capture of design rationale is seen as a separate process from the design of the architecture and the eventual construction of the artefact (Fischer, Lemke, McCall & Morch, 1995). This distinction is problematic, since the capture process is viewed as an intrusive factor without any short term benefits to the architect. This resistance will impede the design process in terms of documenting rationale and can have a long term negative impact on the quality of architecture documentation as a whole.

2.4 Problem Space

The problem points made in the previous paragraphs are related to the following problem space, as shown in the IEEE architecture standard. The elements marked in bold outline the relevant problem space, as scoped by the thesis.

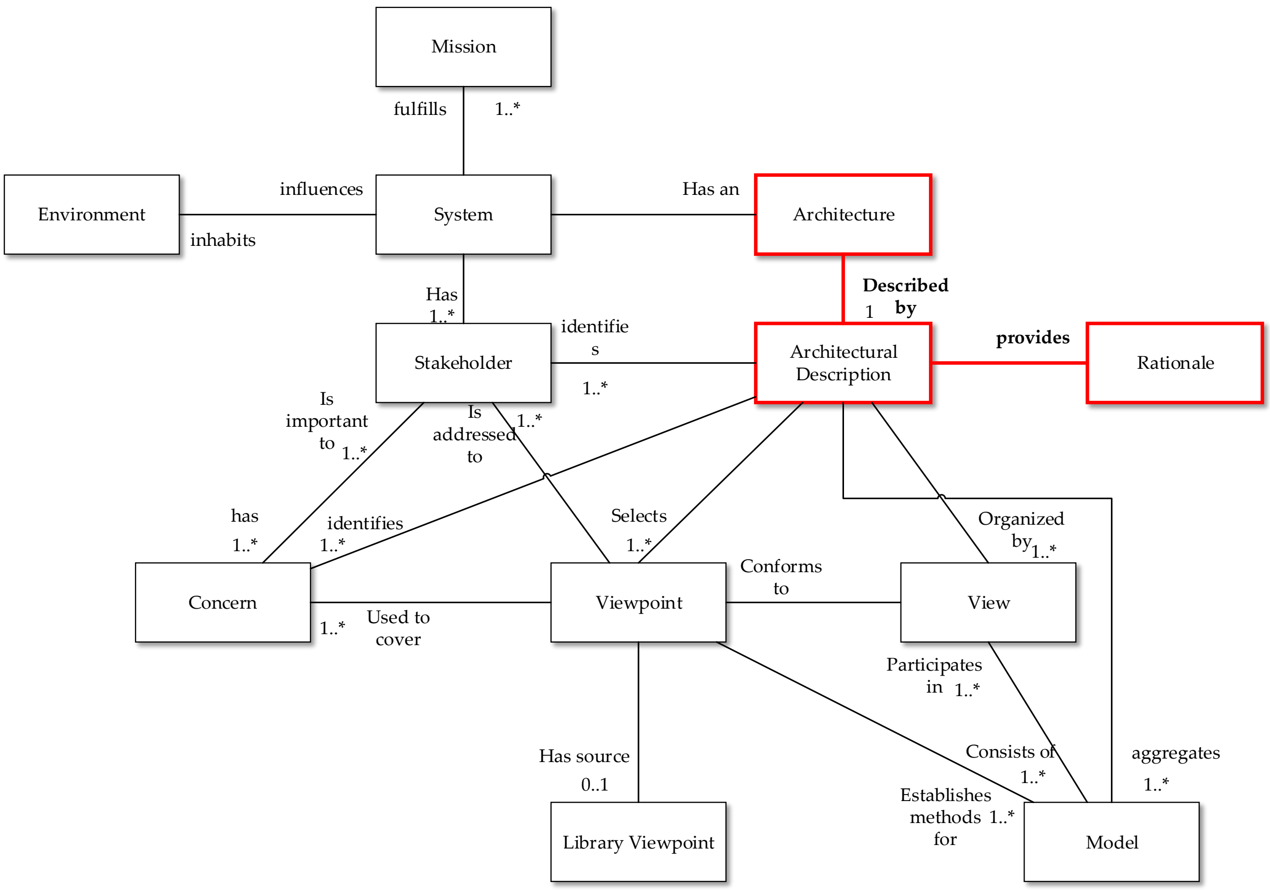


Figure 1. Problem Space, IEEE STD 1471 (Hilliard, 2007)

In order for the architecture to be effectively communicated, an architectural description (AD) describes the contents of the architecture. If the AD is lacking or omitted entirely, the architecture loses its effectiveness due to missing context as the model will be open to interpretation. An architecture in a vacuum lacks critical information and context of the architecture. According to the IEEE standard, the AD is used to express the system and its evolution and includes the communication, evaluation and verification of said architecture. This thesis focuses on the problem of ineffectively describing an architecture, which is caused by the lack of design rationale provision during architecture design.

In short, architects are aware of the benefits of providing design rationale and even utilize it sometimes. However, clear barriers exist. For one, even though a comprehensive survey by Tang et al. (2006) found that indeed architects employ design reasoning, it is unclear to what extent they do so or if they do it correctly. Providing design rationale is open to interpretation, as there is not enough awareness or consensus in the industry on what this term in fact means. A small and implicit reason for why the current architecture implementation was the correct approach might be regarded as sufficient even though a lot of important information and knowledge is omitted and lost. A full architecture design document where every decision is justified, including discarded decisions and their reasoning is as of yet not the common conception the industry has when design rationale is uttered.

Even though the industry recognizes the need for documenting design rationale it is unclear to what extent this is true. This is evident when the current architecture frameworks are analysed and shown to be clearly lacking in this regard. Most frameworks fail to support design reasoning as a process, and important details are omitted. Clear instructions on how to approach documenting design rationale are not present. Some frameworks even fail to mention or acknowledge design reasoning at all. Research found that a significant portion of practitioners lack the resources within project constraints to always provide rationale. Even though this research does not necessarily hold true for every practitioner, it does seem design reasoning as an entity is seen as separate from the architecture design process. Some industry researchers suggest that there is not enough awareness concerning the benefits of design reasoning and how it stacks up against the required time and effort and that a shift in mentality towards design reasoning should take place.

''The extent in which conscious reasoning occurs during architecture design is lacking, potentially resulting in biased design decisions and a less effective architecture.
The architectural knowledge that is produced during the reasoning process also remains mostly implicit, causing architectural degradation and a loss in organizational knowledge as time elapses.''

3 OBJECTIVES AND AIMS

3.1 Overall Objective

The ultimate objective is to provide a means by which architects and designers systematically employ design reasoning. This 'means' can take the form of a model, conceptual framework, guideline document, step-wise approach or a combination. The goal of this utilization of design reasoning is to consistently provide better architecture designs regardless of personal experience. In other words, arming architects with a method that stimulates the utilization of design reasoning is the end game. This approach should also provide instructions on how design rationale itself should be documented, since this element is often omitted in current industry standards.

We also want to better understand how such a method influences the design process and grasp what affect this may have on resulting system architectures. The research tries to create awareness of the barriers design reasoning currently faces, what benefits it potentially could have for architects and how it should be seen as an interwoven process during the architecture design phase. The method should be introduced as an elaborate approach on how design rationale should be documented, whilst concurrently being a simple and comprehensive approach that intuitively stimulates design reasoning. The key benefits mentioned should also be validated and measured during this thesis.

3.2 Research Questions

The overall objective mentioned above translates into the following research question:

MRQ: ''*What is an effective approach to embed design reasoning in the architecture design process?*''

To reach an answer to the main research question multiple sub questions are formulated:

SQ1: ''*What does architecture design entail?*''

The first question explores systems architecture and determines its main types and definitions. The question's main aim is to form a knowledge base concerning architecture in order to grasp the relevance and need of design reasoning in the architecture design process. This is necessary for the next research question.

SQ2: ''*How is design reasoning related to the architecture design process?*''

The second sub question segues into design reasoning and explores its definition with regards to the systems architecture design process. Also, the main benefits of design reasoning with regards to architecture design are defined and current industry standards are analysed.

SQ3: ''*What is the result of design reasoning in the architecture design process?*''

The third sub question looks at the result of design reasoning, the design rationale, and how it's defined. In order to create a design reasoning approach, an idea on how this result should be documented is necessary to incorporate into the approach. This sub question also analyses the current industry standards for the capture of design rationale.

SQ4: ''*How can design reasoning principles be embedded into an approach?*''

The fourth sub question combines the knowledge gained from the previous research questions into an approach or artefact. This sub question finds an answer how to embed design reasoning and –rationale theory and knowledge into an approach and deals with the design and creation thereof.

SQ5: ''*How can the design reasoning artefact's effectiveness be measured and validated?*''

The fifth sub question, which ultimately answers the main research question, finds an answer how to consistently employ design reasoning through utilizing a design reasoning approach and demonstrating its effect. Here, an experimental architecture process (which definition is explained in SQ1) is exposed to a design reasoning approach (which definition is explained in SQ2, SQ3 and SQ4) in order to demonstrate an intuitive and comprehensive manner in which design reasoning can be effectively utilized in the architecture design process (MRQ). Its effectiveness is essentially measured and validated.

Before the methods that answer these questions can be elaborated on and justified, a clear idea of what exactly is being attempted to measure and research is necessary in order to determine whether or not the research goals have been met. To that end, the concepts that are ambiguous mentioned in the research questions have to be conceptualized and operationalized.

3.3 Conceptualization and Operationalization

Conceptualization refers to the process by which ambiguous terms (concepts) are translated into quantifiable, precise constructs (Bhattacherjee, 2012). In order to concretely define these concepts a formal definition is needed and key measures to quantify fulfilment of the research questions have to be identified. Hence, table on the following page is designed, see Table 2. Conceptualization of constructs.

Construct		**Definition**	**Measure(s)**	**Method(s)**
***'effectiveness'* (MRQ)**	*'usefulness'*	*''The quality or fact of being useful with regard to the architecture design process.''*	Survey questions (ch. 8.1.5)	Qualitative analysis
	'intuitiveness'	*''Using or based on what one feels to be true even without conscious reasoning.''*	Survey questions (ch. 8.1.5)	
	'intrusiveness'	*''Causing disruption or annoyance through being unwelcome or uninvited.''*	Time (ch. 8.1.2)	Experiment observation.
			Survey questions at the (ch. 8.1.5)	Qualitative analysis
	'comprehensiveness'	*''Complete and including everything that is necessary.''*	Frequency of occurring design rationale elements (ch. 8.1.4)	Experiment observation
	'design quality'	*''The relative quality of an architecture design and its documentation.''*	Ranking by architects during experiment phase (ch. 8.1.3.1)	Experiment observation

Table 2. Conceptualization of constructs

The MRQ outlines the main goal of the research: the introduction and demonstration of an approach that *effectively* embeds design reasoning in the architecture design process. By effective an approach is meant that does not intrude on the standard architecture design process and encourages the whole spectrum of design reasoning principles. These two criteria can be defined as **intuitiveness** and **comprehensiveness.** Also, the artefact should be useful and not intrude on the regular progression of architecture design. These 4 concepts are explained below and together comprise the *effectiveness* of the artefact. The last measure is the relative quality of an architectural design and its documentation.

By **usefulness** the extent in which the artefact will substantively be of added value during the architecture design process. In other words, does the artefact generate value in some way? In order to gauge this measure, architects will be asked to fill in a survey to evaluate the artefact in this regard.

By **intuitiveness** we mean the ability to understand how to use design reasoning immediately without much deliberation and effort. This is supported by the Cambridge English dictionary definition of *intuition*. The goal is to arm architects with a nonintrusive method, which means the use of the design reasoning artefact should not distract from the architecture design process. This intuition is qualitatively measured by a small survey where participants can indicate the ease of use and intuitiveness of the artefact or approach. This small survey is to be completed by the participants at the end of the qualitative experiment.

By **intrusiveness** we mean the extent in which utilization of the artefact disrupts the regular design process. If the architects completing the case spend significantly longer whilst using the approach some comments can be made regarding the intrusiveness of the approach. This also partially says something about the intuitiveness of the approach.

Another element is the relative **comprehensiveness** of the artefact in terms of rationale elements. Does the artefact demonstrate a significant increase in all design rationale elements when decomposed?

By **design quality** we mean the relative quality of resulting architectures in terms of comprehensiveness and usefulness as determined by architects themselves. The Cambridge English dictionary interprets comprehensive as complete and including everything relevant. The goal here is to provide a means by which the full concept of design reasoning in order to produce better quality designs with accompanying rationale.

3.4 Scope

This study will focus on developing an approach on how to systematically employ design reasoning and demonstrate that a simple method can significantly improve the use of design reasoning in the architecture design process. The concrete architectural domain is intentionally left implicit, as the theories and ideas of design reasoning can be applied across multiple fields of architecture (Plataniotis, de Kinderen & Proper, 2014). Even though the concepts of design reasoning and design rationale mostly stem from the field of software architecture, this study aims on the process of architecture design itself and does not limit itself to a specific field.

The research field of design rationale can be roughly split into two major categories (Regli, Hu, Atwood & Sun, 2000), namely the aspect of representation methods, languages and notations and the field of tool support. The field of design rationale lends itself for tool support considering one of the major drawbacks of design rationale capture is the resources required to do so. The project will focus on the development of a representation method and notation and not on tool support. The reasons for this is the lack of background in software development, which would cause the project to be more focused around tool analysis, debugging and testing. Another major aspect of design rationale research is the issue of representation, which will be the main focus as it matches the background of the researcher.

The study will not focus on quantitative aspects of research like survey research. Any quantitative research, data or information that supports the research is based on existing knowledge in the industry. A theoretical base of knowledge is needed on the current utilization of design reasoning in order to proceed with providing a solution and means. Conducting surveys in order to provide evidence to these statements are deemed outside the scope of this project. This knowledge is inferred from peers whom have done extensive research on the subject and forms the base on which this project continues.

The project will not concentrate on the resulting impact on the quality of the architectures themselves. Research on how to measure impact on architectures and gauge to what extent the quality of the architecture designs improve is hard to grasp and difficult to measure. Research on how to measure the quality of architectures and how to compare an architecture to another requires a complete different project. Therefore, this research will first solely devote efforts to finding a simple, systematic approach to design reasoning utilization and not attempt to prove resulting architecture designs are significantly improved. This question is important, however, and may be considered as a subsequent research on the basis of the findings of this study. The improved quality of architectures due to the utilization of design reasoning in the architecture design process are based on theoretical assumptions and rational judgement.

This study will also not devote efforts on any financial aspects of the consequences of design reasoning, nor will it address the issue of whether the resources needed to utilize design reasoning are financially viable. The suggested pros and cons of design reasoning have no mathematical calculation, but are based on a theoretical framework of peers and other research. This is deemed out of the scope of this project, considering the concerning domain lies far out of proximity as it requires financial expertise to be applied. Also, this consideration has to be made on a case per case basis as each organization might have different factors that influence the decision.

4 RESEARCH DESIGN

4.1 Overview

This chapter contains a heavily summarized version of the research design and theory. An elaboration of the research design with research theory can be found in appendix 5. This chapter will cover the essentials by way of summary and visualization. The overall design of the study is based on guidelines and principles set out by multiple research oriented books, namely: *Researching Information Systems and Computing* by Oates (2005), *Social Science Research: principles, methods and practices* by Anol Bhattacherjee (2012), *Experimentation in Software Engineering* by Wohlin et al. (2012) and *Case Study Research in Software Engineering* by Runeson et al., 2012). The research strategies that are to be applied are as follows:

- **Literature review:** literature study is needed in order to form a theoretical base where the resulting experiment is built upon. The quality and effectiveness of the resulting solution of this research is dependent on a solid knowledge base.
- **Qualitative interviews:** interviews with professionals in the industry is needed in order to gain expert input for the provision of feedback and validation of the design reasoning artefact. Also, Sogeti professionals will be interviewed with the aim of gaining new insights and contextual information of the topic. These interviews can be classified as inductive interviews as they occur relatively early during the research phase and are used to determine the practical perspective of actually using design reasoning and –rationale in the architecture design process.
- **Artefact design & creation:** This strategy concentrates on the actual development of the proposed approach as an IT artefact. Method Engineering is used to combine useful fragments into a best practice artefact.
- **Control- and test group experimentation:** the concluding experimental design provides an answer to the main research question and demonstrates the effectiveness of the proposed design reasoning artefact.

4.2 Design

In order to validate the concept of the research layout the study is designed using the well-known Design Science Research Framework (Hevner, A. R., March, S. T., Park, J., & Ram, S., 2004). The framework can be seen in Figure 2. Applied Design Science Research Framework (Hevner et al., 2004). The framework is applied to the project and the research specifics can be seen in the figure itself.

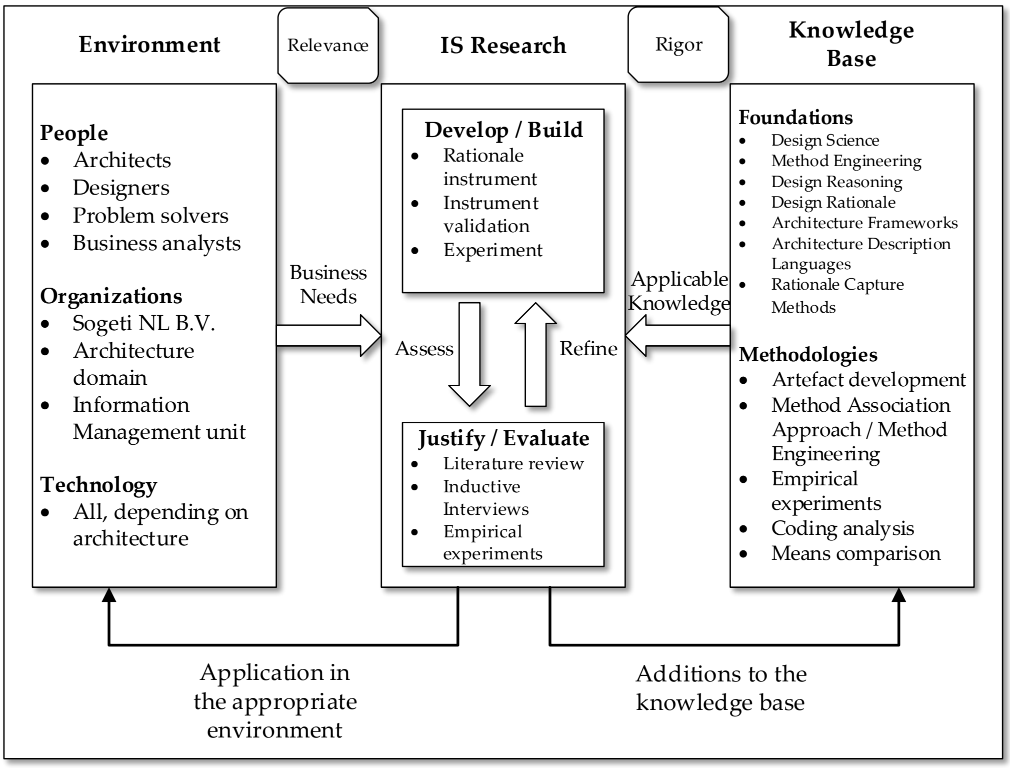


Figure 2. Applied Design Science Research Framework (Hevner et al., 2004)

The Design Science Research Framework by Hevner et al. (2004) provides 7 concrete guidelines to adhere to:

- **Design as an artefact:** the research should produce an artefact, which can take the shape of a method, model or approach.
- **Problem relevance:** the objectives of the research should be relevant to current business operations.
- **Design evaluation:** the proposed solution should be rigorously demonstrates through valid evaluation methods.
- **Research contributions:** the proposed solution should provide concrete contributions to the domain area.
- **Research rigor:** the research should rely on rigorous methods to develop and validate the proposed solution.
- **Design as a search process:** the research towards a solution for the problem domain should be seen as a search, utilizing available means, to reach an end.
- **Communication of research:** the research process and proposed solution should be effectively presented to any audience.

Applying the framework produces a conceptual framework for the research process seen in Figure 3. Research Process Conceptual Framework A.

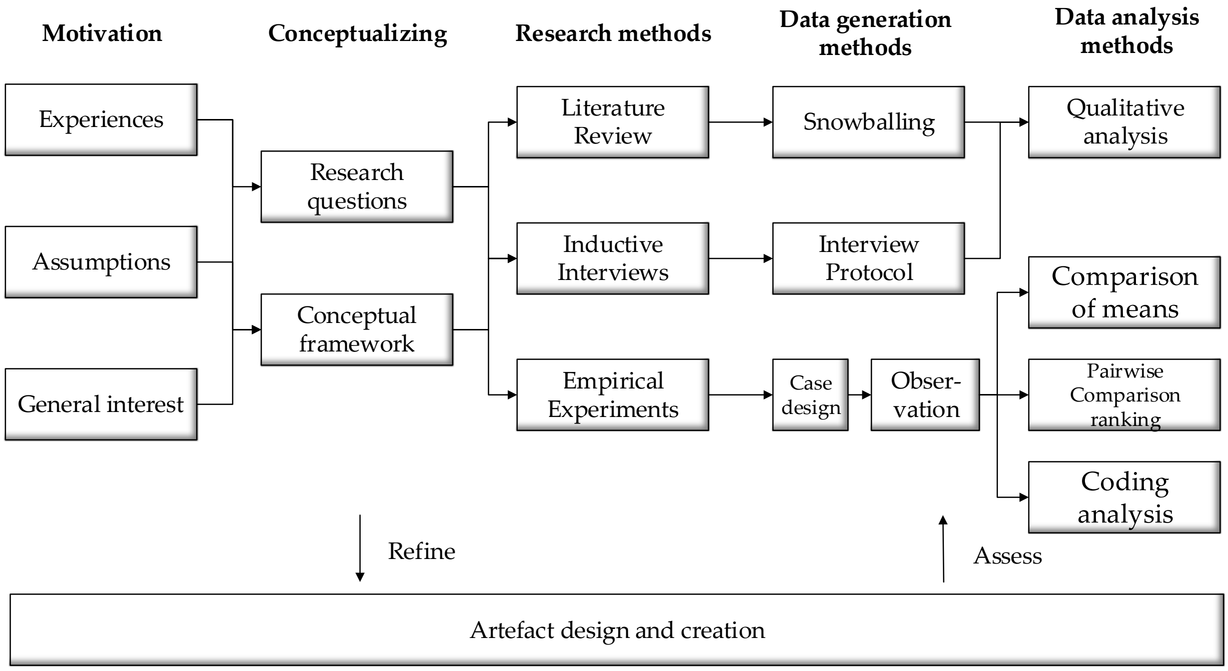


Figure 3. Research Process Conceptual Framework A

First, the research project takes shape out of experience, held assumptions and general interest. Concrete research questions are formed based on previous information and a conceptual framework is designed. In order to answer these research questions specific research methods are chosen that are best suited to answer these questions. These methods demand methods for data generation and are to be analysed through qualitative data analysis methods. The specific approaches to data analysis are elaborated on later in this chapter. The proposed artefact is designed and created throughout the research process, being refined- or assessed by new knowledge gained from the research process. When we specify the research questions and apply them to the conceptual framework it produces Figure 4. Research Process Conceptual Framework B.

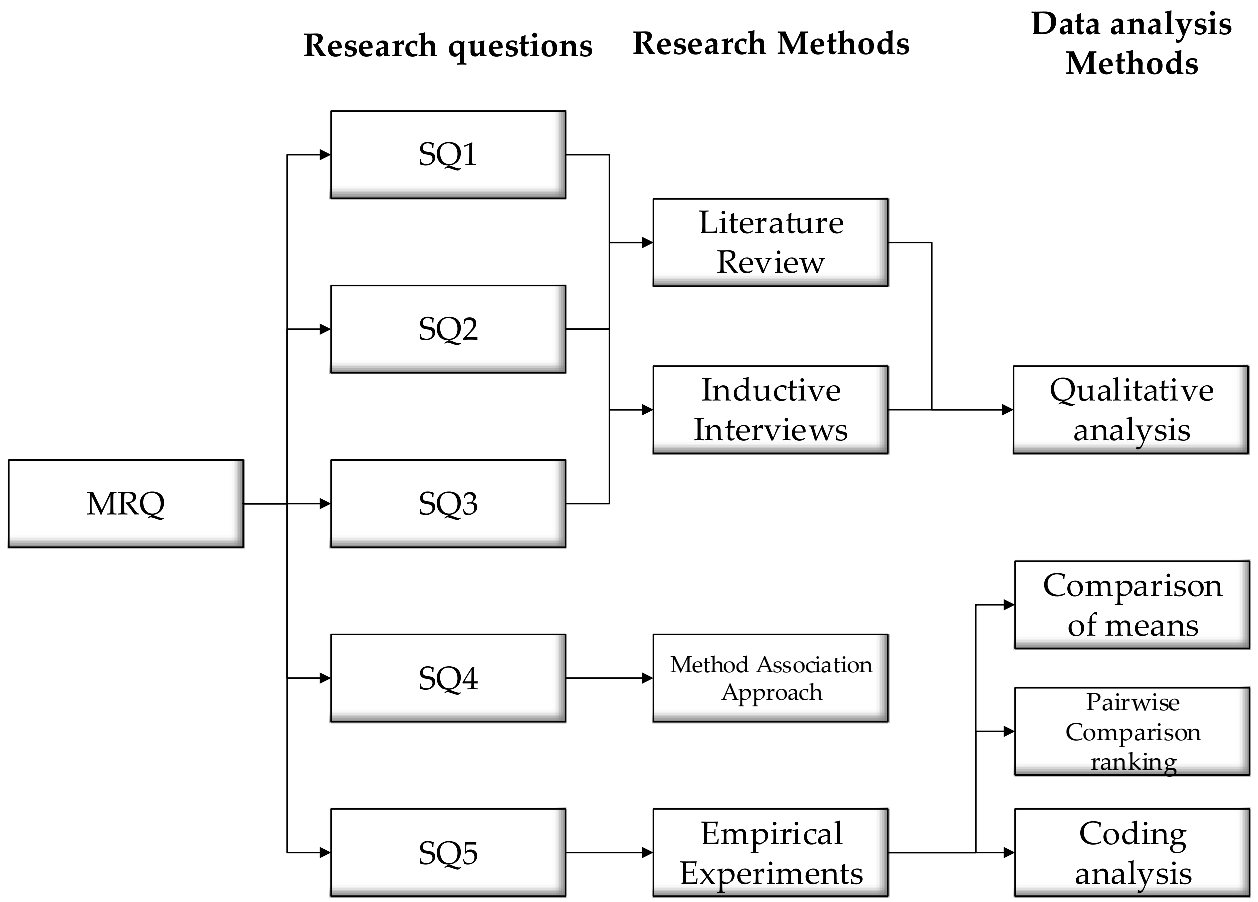


Figure 4. Research Process Conceptual Framework B

SQ1, SQ2 and SQ3 will all partially be answered by literature study. Inductive interviews will complete the answer to these questions for added context and Sogeti specific knowledge. Both the literature study and interviews will provide the research with input for the development of the artefact. The actual design and creation of the artefact will be tackled by SQ4, which deals with the Method Association Approach (MAA). SQ5 will be answered by observing empirical experiments and performing various tests and analyses. SQ5 will also be the only research question that touches a light quantitative aspect. However, due to the low sample size complete quantitative analysis is not the main focus. The interview chapter is moved to appendix 1 in order to condense the thesis, as is most of the literature review (appendix 3) and the Method Association Approach (appendix 4). The following chapters are heavily summarized. The complete chapters can be found in the appendices. From chapter 8 (Results) onward, the chapters are no longer summarized.

> *''Using the design science framework as a foundation, conceptual models can be made to effectively perform the research. This way, the research goals and –questions are made clear, also in the manner in which they are answered.''*

5 LITERATURE REVIEW

5.1 Overview

As mentioned in the research design, snowballing will be the main method for literature review. This means relevant literature is found by following or backtracking from references. The main goal of this chapter is to answer RQ1, RQ2 and RQ3. These questions are partly supported by answers from the qualitative interviews, however. To fully take advantage of the knowledge of Sogeti professionals, the results from these interviews will provide inductive information and context. More on this in can be found in appendix 1.

The main aim the literature study during this project is to gather knowledge relevant to the topic that supports the research project in gaining new knowledge. Similarly, literature review assembles knowledge that supports the claim this thesis' goals are worthwhile and realisable. Also, the gathered knowledge should support the thesis in that it does not merely repeat the work of peers and that gained knowledge was previously unknown on uncertain. It should also point to clear gaps in the existing knowledge base and clearly demonstrate how the knowledge found by the thesis fills that gap.

Relevant material will be retrieved from online databases such as Google, Google Scholar, Gartner, ACM Digital Library, DC library etcetera, by way of simple keyword queries. The utilization of multiple databases allows for a broader coverage, however it may result in duplicate studies which will have to be manually removed (Wohlin, Runeson & Höst, 2012).

The search approach is done by way of *snowballing*. The snowballing procedure means to follow references between papers to find other relevant papers (Runeson & Skoglund, 2009). Both backward and forward ad hoc snowballing will be employed. Backward snowballing follows the references in a specific paper and forward snowballing refers to viewing the papers that have cited the specific paper.

First, context to the concept of architecture is sought. It seeks out the main types of architecture design and how these distinctions are made. Its main goal is to grasp what architecture entails, so that the relationship with design reasoning has further meaning. Secondly, the concept of design reasoning will be explored. Its definition will be sought, including main benefits and application in the industry. The product of design reasoning, design rationale, will be explored in the last paragraph. Its definition is sought, including its rules and guidelines and an analysis of industry standards.

It is important to note the literature review chapter contains summarized elements for improved readability. The full chapters can be found in, and are referenced to, the appendices.

5.2 Architecture

5.2.1 Overview

Architecture refers to the fundamental elements of a system and how they relate to comprise a perspective of a system in its context (ISO / IEC / IEEE FDIS 42010:2011). The main goal is manage the system's growing complexity and allows for support in the maintenance thereof (Lankhorst, 2009). It is often used to present slices of a system to demonstrate that the stakeholder's wishes and needs are addressed. The design is done through various views to prevent an overwhelming image of the system. The views are designed through the use of various notations like UML and ArchiMate which support the architecture design of various types of architecture, among which are enterprise-, software- and hardware architecture (Bass et al., 2003). Since the field of architecture is quite complex, the industry offers frameworks that guide the process of architecture.

In this thesis, architecture is seen as the culmination of design decisions (Poort, 2014). Defining and viewing architecture as a stream of design decisions made by the architect helps focus on the rationale behind decisions. Rationale is essential to any architectural description as it explains the reasoning behind why the architecture is as it is. Instead of solely viewing architecture as a blueprint of a system, this principle helps architects to view architectures as a complete whole and process. It also helps to communicate any changes and potential implications to the architecture to peers and stakeholders. It allows the recording and capture of rationale for the architecture, which alternatives were considered and why the final decision has been made. Also, it allows tracing back of principles and decisions as to why decisions were made at the time, providing a basis for reconsideration if necessary. The concept of rationale and its importance will be elaborated on further in this thesis, as it forms the main principle on which the research is based.

The study aims to generalize the results and the approach to all forms of architecture, even though their applications and domains can be different. The research aims for the design process itself, regardless of application domain. Even though there are differences between the design processes, the essence remains similar. Every design process still features design decisions that have to be made when problems and issues arise.

An important distinction has to be made, however, between solution architecture and enterprise architecture. Solution architecture contains all forms of architecture that are designed around a solution i.e. addressing an issue or problem (Greefhorst & Proper, 2013). These forms of architectures are usually made for a single project as it concerns a specific deliverable or issue. Enterprise Architectures, however, are not necessarily made to solve a certain problem. Enterprise Architectures can also be made at the start of a project to guide the development process and to identify the borders and context in which change is meant to occur. Enterprise Architectures can have a guiding role, instead of solution architecture which focuses on fixing an issue. Enterprise Architectures often do not deal with a specific set of system requirements for a new system, for instance. However, even in Enterprise Architecture, design issues can still arise when creating the architecture due to the availability of alternative options. These options have to be evaluated, compared and weighed. Considering the approach is concerned with documenting design decisions themselves, and not the specific applications of architecture, the scope remains relevant for Enterprise Architecture as well. For a more elaborate literature analysis on the concept of architecture, refer to chapter 3.1, Architecture in appendix 3, Literature Review. In this appendix, various definitions of architecture are analysed and compared. Also, a more in depth analysis with regards to the architecture process, architecture design and different types of architecture is present.

5.2.2 Architecture Frameworks

We have previously established that the key goal of architecture is to manage the growing complexities of systems. Architecture Frameworks provide architects with guidance on how to develop architectures (Lankhorst, 2009). Considering the process of architecture design is complex and intricate, frameworks offer an overview of elements that an architecture should contain and instructions on how to create these elements. For simplicity, the analysis of the architecture frameworks analysed in the appendix are summarized in Table 3. Reasoning / rationale support in Architecture Frameworks. This table features each architecture framework and marks them with regards to what extent it supports design reasoning or –rationale. The rows represent the individual architecture frameworks whilst the columns represent the extent to which design reasoning or -rationale is supported. This information is found by analysing the frameworks themselves, accompanying documentation and relevant framework specification documents from the organizations, departments and businesses themselves.

Architecture Framework	Guidance / Description	Example / Syntax	Mention	No support
Zachman				X
TOGAF			X	
IEEE			X	
4+1			X	
MDA				X
ARIS				X
FEAF				X
DoDAF			X	

Table 3. Reasoning / rationale support in Architecture Frameworks

Full support means the framework completely guides and instructs the architect to fully employ design reasoning and elaborates on how to do so. *Guidance / description* represents frameworks that provide at least some detail on how design reasoning can be utilized or describe how design rationale can be captured. *Mention* is ticked when the framework, documentation and specification at least mentions design reasoning or –rationale *should* be used. *No support* is for frameworks that in no way, shape or form support or even mention design reasoning can or should be used.

Only half of the standards in industry architecture frameworks only mention the use of design reasoning or the capture of design rationale. The other half do not mention design reasoning as a concept at all, nor does it mention rationale for making design decisions should be kept and documented. No architecture frameworks guide the architect in terms of reasoning on their design decisions, nor do they provide instructions on how rationale should be documented and kept. Therefore, we can say the support for design reasoning and –rationale in current architecture frameworks is very limited.

For a full description and analysis of each framework, including a modelled example, please refer to paragraph 3.3 in appendix 3.

5.2.3 Architecture Description Languages (ADLs)

As mentioned previously, architectures are large and intricate designs that, in order to create them, require frameworks to support the development process. As frameworks offer high over guidance on creating architectures, frameworks themselves often do not provide a language or notation in which to make designs explicit. Those languages and notations are Architecture Description Languages (ADLs) (ISO / IEC / IEEE, 2011) and have predefined syntax and semantics that are suitable for the representation of architectures. ADLs offer a clear medium for architects to create and communicate architecture designs.

Similar to the previous paragraph, where a table summarizes the findings of the analysis of architecture frameworks, another table for the analysis of ADL's is made (see Table 4. Reasoning / rationale support in ADLs). Again, this information is found by analysing the frameworks themselves, accompanying documentation and the relevant framework specification document. The same elements for the rows and columns are used.

ADL	Guidance / Description	Example / Syntax	Mention	No support
ArchiMate				X
BPMN				X
Petri nets				X
UML				X
AADL				X
SysML			X	
SysADL		X		
GRASP		X		

Table 4. Reasoning / rationale support in ADLs

No ADL explicitly guides the use of design rationale in their models. Only SysML even mentions that the capture of design rationale should be done in their accompanying specification sheet and SysADL and GRASP both offer designers an explicit structure and syntax to provide rationale. However, SysADL and GRASP only offer an example in terms of structure. Both ADLs do not guide how rationale should be captured nor does it fully explain to what extent rationale should be made explicit. All other ADL's do not mention design reasoning should be used, nor do they indicate in any way, shape or form that the capture of design rationale should be done. The ADL's also offer no clear support for the capture of rationale in their syntax. Therefore, we can say the support for design reasoning and –rationale in current architecture description languages is very limited.

Another finding during literature study and the interview phase is the difference between architecture and design. Even though they may seem similar, the difference is subtle. Even though, in terms of software architecture, the two are closely intertwined they are not necessarily equal. Architecture can refer to a practice that guides development or organization whereas design specifies a certain end. Architecture can be concerned with placing limits and handles to future development and IT organization based on organizational needs whereas design is more specific. Design is a means to an end and its goal is to specify the elements of that end. In architecture design, both entities occur. When reasoning however, this distinction holds no water as decision making occurs on all levels of architecture. The precise outcome of reasoning may be different, as architecture produces guidance documents and models whereas design produces the specification of a product. In both cases, however, design decisions exist. And these design decisions require reasoning to be justified and substantiated.

For a full analysis of each ADL, including a modelled example, see chapter 3.4 in appendix 3, Literature Review.

5.3 Design Reasoning

Design reasoning refers to the process of reasoning and coming to a design decision (Tang et al., 2006). So if an architecture design is faced with problems due to stakeholder requirements or regulatory influences, design reasoning helps to tackle this problem in a logical fashion. The process of reasoning can be summarized into a few key principles, among which are reasoning and inferencing, problem structuring and assumption-, constraint-, option-, trade-off- and risk analysis (Tang, 2011).

Even though the principles are theoretically sound, the industry has not adopted them completely. Overall, industry practitioners agree that design reasoning principles are important, even though the extent in which it is used is limited (Tang et al., 2006). The detail in which design rationale is explicitly documented is, also, unknown.

If architecture is established to be a practice, an architectural design can be the product. Both of these terms contain design decisions. Ideally, this decision making process comes about logically and rationally. To achieve this, the process should stretch the expression of ideas, the evaluation of alternatives, deliberation, consideration, arguing, pondering, debate and presentation. These elements together form design reasoning, i.e. a rational and explicit process where logical reasoning is utilized in order to come to a design decision.

Design reasoning can be utilized in many different contexts. However, when generalized, some key elements remain. Architects should reason and logically infer and deduce from conflicting, emerging or duplicate information. Architects should structure their problems soundly and define various options in order to achieve this premise. Additionally, list any assumptions, risks or constraints that are concerned with these options. Logically and equally weigh these options in order to decide which option best suits the situation is a key principle to utilize this process.

Currently, this process is utilized in limited fashion depending on a variety of factors. These factors include organizational size, the practice topic, business domain, project size or even individual preference. As to why, many barriers to entry still exist in the industry, barring growth in awareness and utilization. The main issue being an industry constraint, namely few standards exist in order to provide handles and support.

For a more elaborate literature analysis on the concept of design reasoning, refer to chapter 3.5, Design Reasoning in appendix 3, Literature Review. In this appendix, various definitions of design reasoning are analysed and compared. Also, the main principles of design reasoning are explored and the utilization in the industry is analysed.

5.4 Design Rationale

5.4.1 Overview

If design reasoning refers to the deliberation process when faced with design problems in an architecture, the product of this process can be called design rationale, i.e. the justification for a design decision (see Figure 5. Reasoning and Rationale Relationship).

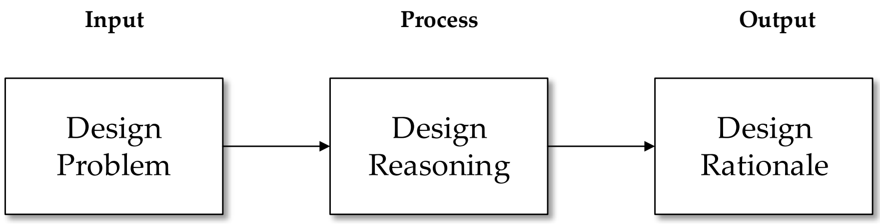


Figure 5. Reasoning and Rationale Relationship

The applications of design rationale are widespread, and include the ability to verify and evaluate designs, maintain and reuse them, communicate and teach them to others and to assist architects in the design process (Burge & Brown, 1998). If not used, architectures become increasingly brittle as time elapses due to architectural degradation (Perry & Wolf, 1992). Also, organization knowledge decreases due to *architectural knowledge vaporization* (Tang & van Vliet, 2008; Plataniotis et al., 2014).

These aforementioned concepts can be further explained with design science theory, more specifically when viewing the Function-Behaviour-Structure (FBS) ontology by Gero (2014). The FBS ontology describes a fundamental insight into design science as it describes the process steps required to bridge between function and structure. It conceptualizes design objects into three distinct elements: the function (*F*), i.e. what the object is meant to do, its behaviour (*B*), i.e. what the artefact does, and structure (*S*), i.e. the object's components and relationships. View the following figure.

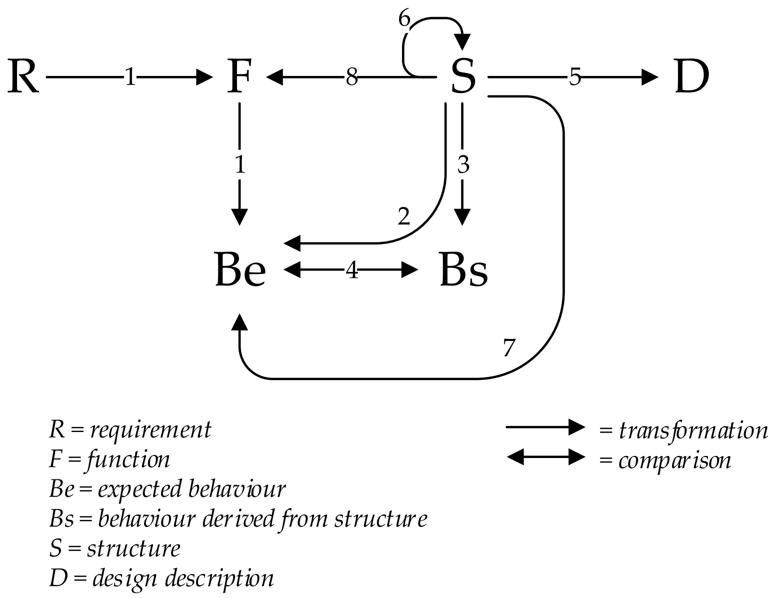


Figure 6. The Function-Behaviour-Structure Ontology (Gero, 2014)

If we map this ontology to architecture design, the goal is to achieve improvement in step 4. In step 4, the behaviour of the design object (system) is compared to the expected behaviour (architecture). In an ideal scenario, the expected behaviour is equal to the behaviour derived from the structure, i.e. '*Be*' = '*Bs*'. The difference between '*Be*' and '*Bs*' over time can be seen as architectural degradation (*Be – Bs * time*). Failing to mirror '*Be*' and '*Bs*' causes the architecture to become brittle (*Be – Bs*). In other words, the goal for an architecture is to represent the reality of the system as much as possible (*Be = Bs*). In order to do so, step 5 is an essential step in the design science ontology. '*D*' represents the design description, which is documented in step 5. This documentation is necessary in order to document and describe the expected behaviour of a structure. In terms of architecture design, however, a crucial element, namely design rationale, in this step is lacking. Improvements to step 5 (architecture description documentation) have to be made in order to mirror the expected behaviour (architecture) to the derived behaviour of the structure. If achieved, architectures are less and less brittle, and potential risks that arise due to unawareness of a system's actual behaviour (*Bs*) decrease.

Rationale can be split into various mutually inclusive types: argumentation-, history-, device-, process- and active document based (Burge & Brown, 1998). The ideal rationale is argumentation based, interwoven in the design process whilst taking the dimension of time into account. Much debate exists how rationale should be captured, however, and what elements it should contain. Also, various approaches to the capture itself exist. The ideal capture method is the methodological by-product approach due to its least intrusive nature (Tang et al., 2006). This method involves the rationale production occurring naturally by following a concrete method. The trick is, however, how to develop a method that stimulates rationale capture whilst being nonintrusive in the architecture design process. This is one of the key challenges addressed the next chapter.

For a more elaborate literature analysis on the concept of design rationale, refer to chapter 3.5, Design Rationale in appendix 3, Literature Review. In this appendix, various definitions of design rationale are analysed and compared. Also, the main benefits and applications of design rationale are defined and the utilization in the industry is analysed. The chapter also contains elaboration on the types of design rationale and how it is captured.

5.4.2 Rationale Representations

In order to use design rationale effectively, rationale representation methods are essential. These methods provide an approach to capture design rationale, including how it should be documented. There have been certain attempts at providing methods that solve the lack of design reasoning and rationale capture. Similar to previous chapters, where a table summarizes the findings of the analysis of architecture frameworks and architecture description languages, another table for the analysis of rationale methods is made (see Table 5. Rationale Representations Compared). Again, this information is found by analysing the representations themselves, accompanying documentation and the relevant academic papers. The columns represent the various rationale representation methods that are used in the industry. The rows are relevant properties identified and discussed during the literature review that the methods either do or do not possess.

	QOC	IBIS	DRL	ADDT	V&B	AREL
Capture Method	Method. by-product	Method. by-product	Reconstruction	Reconstruction	Reconstruction	Method. by-product
Notation	Semi-formal	Semi-formal	Semi-formal	Informal	Informal	Formal + semiformal
Access Method	User-initiated	User-initiated	User-initiated	User-initiated	User-initiated	User-initiated
Guides reasoning process?	Yes	Yes	Yes	No	Yes	No
Elaborates on rationale capture?	No	No	No	Yes	No	Yes

Table 5. Rationale Representations Compared

The table above lists the analysed rationale representations in terms of their capture method, level of formality and access method. It also determines whether the approach guides the deliberation and design process itself by providing a qualitative decision support system or if it elaborates on what rationale needs to be captured and in what detail. The analysed approaches do not attempt both. In chapter 3.6.5, Rationale Representations of appendix 3, Literature Review, the full analysis of rationale methods can be seen.

5.4.3 Design Rationale in Practice

The current state of the industry demonstrates a clear lack of a standard in design rationale utilization (Regli et al., 2000; Tang et al., 2006; Verries et al., 2008). There is a lack of a standard notation, representation method, reasoning guide and capture language. The analysis of existing design rationale methods demonstrates a distinct lack of a method that stimulates design reasoning principles in a nonintrusive manner, whilst also identifying which rationale elements should be defined and instructing how they should be documented for visibility and traceability. The suggestions and critiques put forth by other researchers guide the development of an approach that fills these gaps and can be seen in chapter 3.6.5 in appendix 3. For a full analysis of the industry's view on design rationale methods in the industry, see chapter 3.6.6, Rationale Methods in Practice in appendix 3, Literature Review.

''Architecture has many different domains and applications, each with different industry standards and frameworks. However, each type of architecture features design decisions which have to be reasoned. The extent in which reasoning and documentation is supported by these standards is very limited, pointing to a clear gap in the current industry.''

6 METHOD ASSOCIATION APPROACH

6.1 Design Philosophy

The main issue with design reasoning methods is that they are too resource intensive to use (Tang et al., 2006), i.e. they are either not intuitive, not approachable, take too much effort or too heavily disrupt the regular progression of architecture activities (Regli et al., 2000). The main goal is therefore to create an easy-to-pickup, intuitive, nonintrusive and simple method. However, this method should be comprehensive as well, meaning all design reasoning principles (identified in paragraph 3.5.2, Principles in appendix 3, Literature Review) should be addressed and included.

As mentioned previously and found during the literature review, the ideal capture method of rationale is as a methodological by-product (Gruber & Russell, 1992; Regli et al., 2000; Tang et al., 2006) because it least interferes with the standard progression of the design process. Therefore, the artefact should take shape as a method that follows consecutive steps that naturally forms the rationale.

A research by Karsenty (1996) suggests that argumentation based design rationale methods like QOC and IBIS are insufficient and ineffective. The study finds that the method would insufficiently capture rationale as half of the rationale inquiries could not be answered by the rationale document. The study concludes that solely relying on those approaches will produce rationale that is incomplete. Tang (2007) later confirms this premise by stating effective capture of design rationale should include the argumentation structure of rationale and should by simplified without losing key rationale information. Therefore, the artefact should include comprehensive rationale capture instructions and structure whilst guiding the deliberation process through a stepwise method.

A study by Buckingham Shum & Hammond (1994) confirms this direction and design philosophy and found that using argumentation based rationale approaches as a record of rationale information were ineffective. The study also concluded that there already was a tendency for these approaches to take shape as text based rationale, which supports the premise towards a combination of a stepwise approach and informal textual rationale.

6.2 Method Association Approach (MAA)

6.2.1 Overview

As determined by the literature review, the most likely successful method of rational capture is as a methodological by-product (Gruber & Russell, 1992; Regli et al., 2000; Tang et al., 2006) because it least interferes with the standard progression of the design process. In order to create this method 'method engineering' will be applied in order to structurally construct a situational design method based on existing industry methods (Luinenburg et al., 2008). This approach is fitting as it guides the consideration and comparison of industry methods into a relevant situational method to create a best practice relevant for the specific domain of architecture design. The overall process can be seen in the figure below.

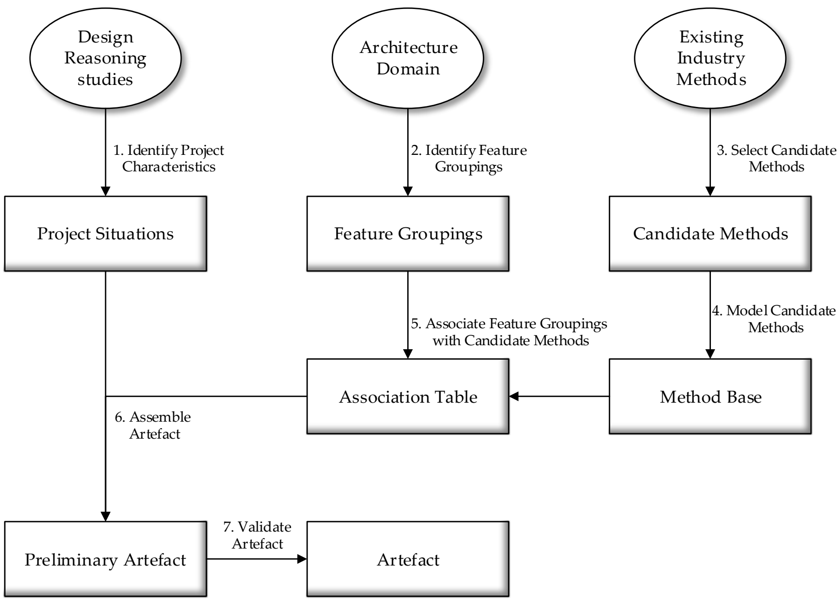


Figure 5. Method Association Approach (Luinenburg et al., 2008)

The full Method Association Approach, including all steps and the method base and its models, can be found in appendix 4, Method Association Approach. For the sake of readability of the thesis, the final artefact and end result of this process that was used during the experiment sessions is featured below.

6.2.2 Preliminary Artefact and Design

6.2.2.1 Visualization and Simplification

During the design of the artefact it was deemed necessary by the University supervisors and Sogeti representatives to transform the artefact into a visual model that distracts less and allows for a simpler interpretation. The PDD notation clutters too much and allows for too much ambiguity. The notation also requires too much training to understand easily and effectively. The final artefact that is used during the experiment phase to gather data can be seen in the figure below and the model is named the 'Rationale Capture Cycle'.

This model allows the user to quickly understand what is expected in order to capture rationale in a simple manner. The keywords remain and some textual context is given for enhanced understanding. The model is to be read from top to bottom. First, a problem has to be present in order for the design decision process to trigger. The architect should formulate any options and alternatives in order to tackle this problem or situation. From there, the model basically goes into an iterative loop of rationale elements. If these rationale elements (i.e. categories) are involved in the reasoning of the architect, a more well-rounded critical thinking process during architecture design is achieved. The middle circle represents the constant evaluation and reflection the architect should employ. This is vital because the explicit reasoning should prompt architects to reflect on previous assumptions and alternatives. This element is therefore modelled in the middle of the model, as it should happen constantly.

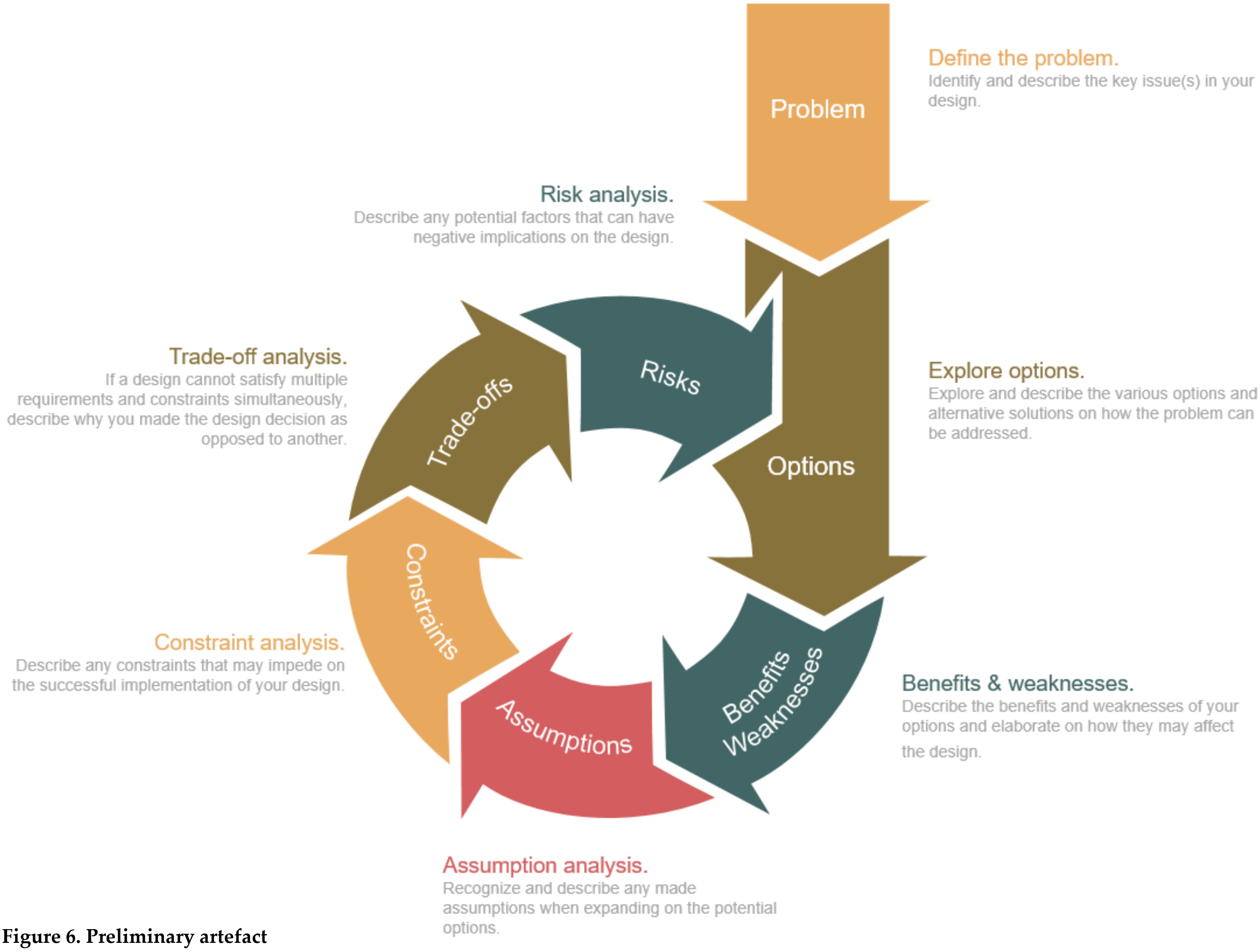


Figure 6. Preliminary artefact

6.2.2.2 Sequence

The sequence of rationale elements is based on industry literature (Bass et al., 2003; Tyree & Akerman, 2005; Tang, 2011), interviews with practitioners and supervision of University and Sogeti representatives. The first two elements (Problem & Options) is based off of the design planning and problem-solution co-evolution theory by Tang (2011), see Figure 7. Design Planning and Problem-Solution Co-evolution.

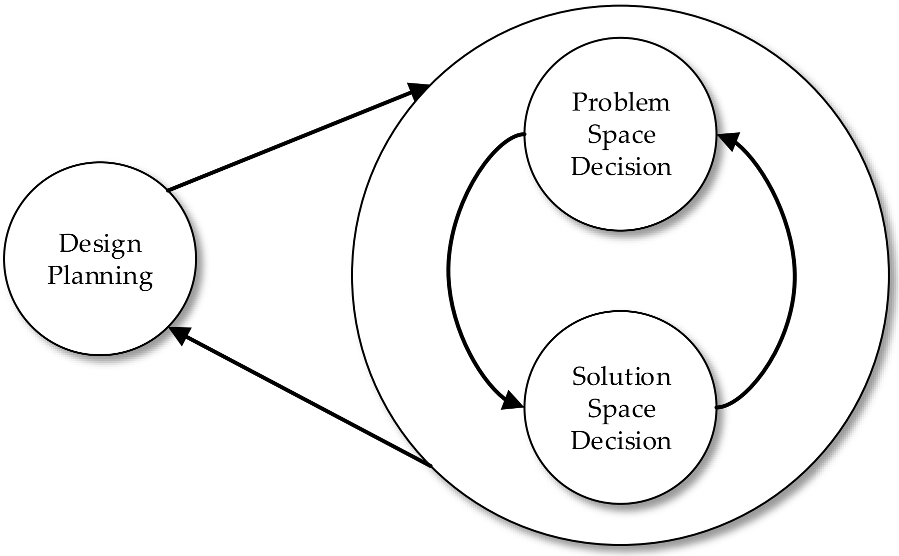


Figure 7. Design Planning and Problem-Solution Co-evolution (Tang, 2011)

According to the theory, designers should consider a high level design plan first. In other words, an overview of the requirements or design issues should be identified. Early decisions have been found to heavily influence the process in which design activities are carried out (Tang, 2011). In other words, in order to correctly design a solution architecture, the architect should identify and clearly decide on which issue is being tackled first. This premise has been adopted into the 'Rationale Capture Cycle'. Secondly, designers have to consider the potential solutions that actually tackle the identified issue. This iterative process has no real clear template. Architects have to consider multiple options and consider which option will fit the issue the best. Considering the lack of a clear recipe for this process, this element requires constant evaluation and reflection which is reflected in the shape and elements of the cycle.

When concrete options have been defined, the architect should continue to elaborate on those options by defining the key benefits and weaknesses of each design option. The exploration of these options should be handled with care and rigor, which is why the assumptions and constraints have to be acknowledged and incorporated into the decision process. Having identified and explored the options, the architect should identify trade-offs if necessary. Sometimes, not all requirements or constraints can be satisfied simultaneously. In that case, a decision has to be made between one and the other. The architect should identify and describe this process. Lastly, any unknown risk factors that may negatively impact the design option should be identified and explored. When done, a solid reasoning process has been performed and a rational, informed decision can be made.

6.2.3 Rationale Capture Cycle

The preliminary artefact was used to gather data during the experiment sessions. The final artefact, i.e. the Rationale Capture Cycle, is featured below. The model is the end result of various feedback and validation sessions.

Figure 8. Rationale Capture Cycle final

> *''The Method Association Approach allows for a structured way of assembling a preliminary method that might stimulate reasoning and rationale capture. After completion, various additions and modifications had to be made in accordance with expert feedback and validation to ensure user adoption.''*

7 EMPIRICAL EXPERIMENTS

In this chapter the main experiment will be elaborated on. For a full theoretical and scientific support found during the literature review phase, see chapter 2.4, Empirical Experiments in appendix 2. The main goal for the experiment is to determine the effectiveness of the design reasoning and –rationale artefact designed in the previous chapter. This effectiveness is operationalized through 4 constructs: intuitiveness, intrusiveness, comprehensiveness & design quality. In other words, the DR artefact has demonstrated its effectiveness if it can be easily applied, does not intrude in the regular design process, stimulate a comprehensive rationale capture and provide a higher quality architecture design. These three concepts are measured in paragraph 7.2.2, Measures.

7.1 Setup

The sessions take place at the Sogeti headquarters in Vianen, the Netherlands. The sessions take place at night, from 18:00 to 21:00, in closed-off locations to ensure concentration. Participants solve the same case individually, in their own preferred notation. When the solution is finished, the participant is asked to fill in the finishing time and sends the solution to the researcher. Or, if not made digitally, it can be handed in immediately. The architects are free to design their solution the way they see fit, yet are encouraged to provide rationale for their design decisions in a way they would normally do during their regular design activities.

7.1.1 Moment 1: Architecture Design

The participants are asked to complete an architecture case where they are given 2 hours to complete an architecture design. One half of the participants is given the Rationale Capture Cycle to support the design process. The goal is to identify a difference between the solutions produced by the architects who have used the artefact versus the architects who have not. The architects are supposed to provide architecture models through ArchiMate on the one hand, and their explicated reasoning process through a Word document on the other. The full case explanation and introduction can be found in appendix 5 and 6. The experiment is visualized in the following model.

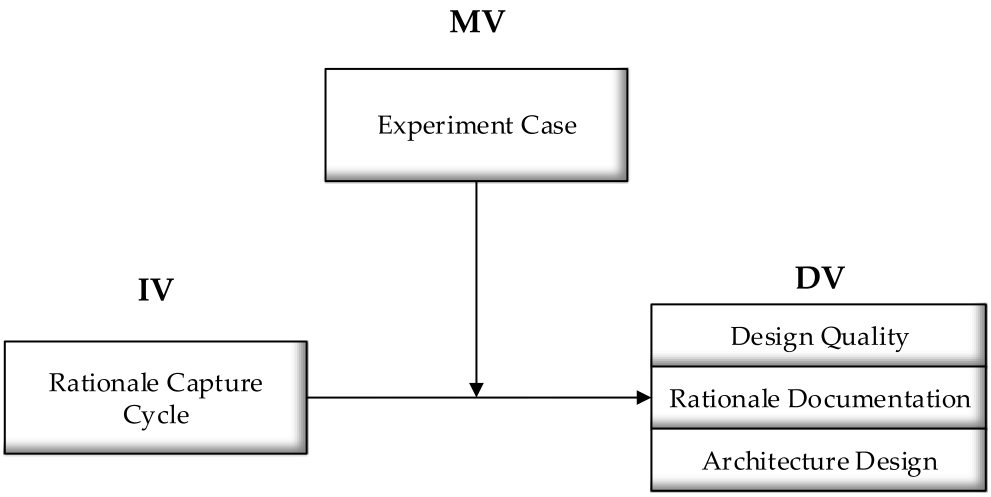


Figure 9. Experiment Session 1 Variables

7.1.2 Moment 2: Request for Change (RfC)

During the first experiment session the goal was to determine whether the capture of design rationale can be stimulated and reasoning can be supported. A second experiment is designed with the inclusion and exclusion of design rationale in mind. The architecture solutions from the first experiment session, including their rationale, is used as case material. In order to demonstrate the effect of having or not having design rationale, the architects were asked to reconsider the solutions made in moment 1 and implement changes or make modifications based on updated stakeholder requests and concerns. The test group would have access to the design rationale as it was made by the original architect during moment 1, the control group would have to make do with a brief description of the architecture. The assumption is that having access to the original rationale and context in which the architecture was designed would have a measureable and beneficial influence on architecture design. If successful, the results demonstrate that it is indeed beneficial to have access to design rationale on the one hand, and on the other hand a reasoning model is demonstrated to stimulate that rationale. The architects are supposed to provide architecture models through ArchiMate on the one hand, and their explicated reasoning process through a Word document on the other. The full case explanation and introduction can be found in appendix 5 and 6. The experiment is visualized in the following model.

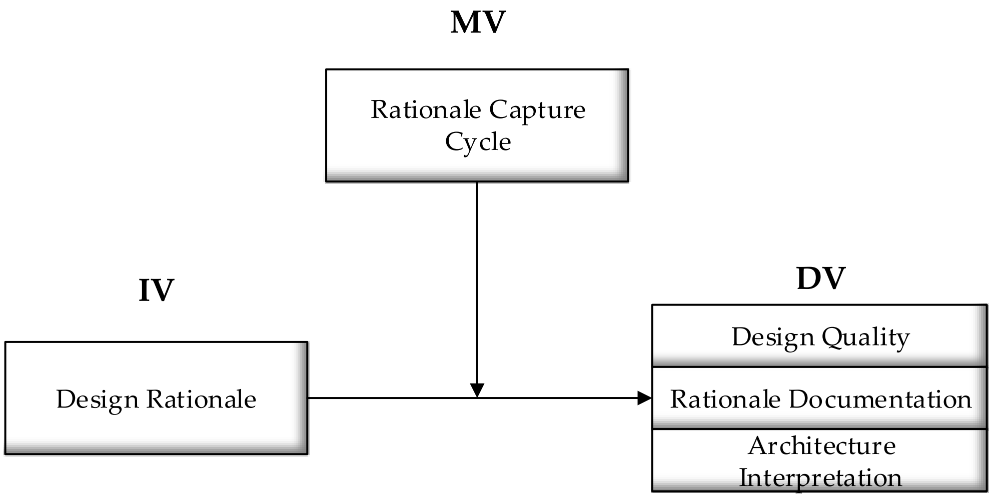


Figure 10. Experiment Session 2 Variables

7.1.3 Timeline

10 architecture practitioners from Sogeti will be the main participants of the first experiment. The second experiment will feature 8 architects. All participants are currently practicing architecture at various institutions in the Netherlands. To prevent bias in data, the interviewees were not shown any material or given any information related to the experiment. A summarizing timeline of the experiment can be seen in the following figure.

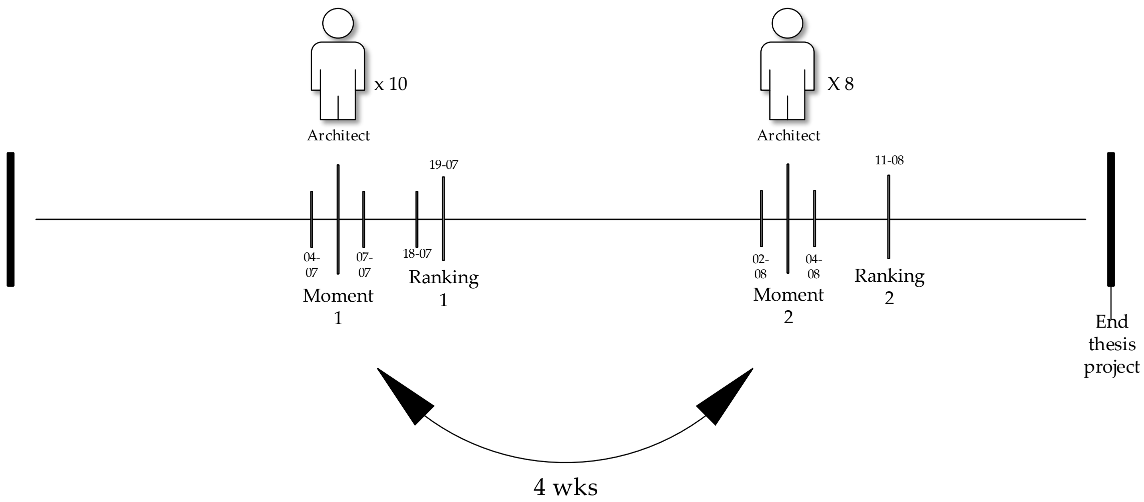


Figure 11. Experiment Timeline

7.2 Design

7.2.1 Research Design

The goal of the experiment is to explore a relationship between design reasoning effectiveness and the use of the Rationale Capture Cycle (Oates, 2005). The experiment is performed by way of static group comparison, where a post-test only control group design is chosen. A more in-depth elaboration on research design theory can be found in chapter 2.4, of the appendices. The aim of an experiment-based research strategy is to show that only one single variable is responsible for the observed effect or change (Oates, 2005). Considering, in this project, the experiment only occurs once all contributing factors have to be controlled at once. The following factors are the foreseeable influencing factors that need to be controlled:

Experience and age: regarding professionals, the test- and control group are non-randomly split into groups of equal experience, age and skill. This is to prevent individual expertise is causing the observed change in effect instead of the artefact.

Familiarity: due to limited availability of the Sogeti professionals, as many architects that were available were invited to participate in the experiment. Some of these participants were already interviewed during the interview phase, and were thus familiar with the concept of design rationale. In order to control for this factor, architects who were already interviewed are equally split into control and test groups.

Notation: in order to guarantee the observed change in effect is indeed caused by manipulating the independent variable (the artefact) the notation of the architect during the experiment has to be controlled. If not done, architects might be influenced by ranking a certain notation differently than others due to experience, familiarity or comfort. Therefore, all architects are asked to complete their solution with ArchiMate as a standard notation.

Tool: as mentioned before, the output has to be controlled. This includes the tool with which the architects are supposed to solve the case. Therefore, the architects are asked to solve the case using 'Archi', as it is a standard ArchiMate freeware tool.

Template: similarly, architects are supposed to rank the solutions of their peers on their merit and content, not in the way it was structured or noted. Therefore, a standard template for accompanying rationale documentation will be designed in order for the solutions to be as equal as possible. The only difference should be caused by the artefact, which is measured by the content that is produced. This template is described in 7.4.3, Rationale Documentation Template.

7.2.2 Measures

As mentioned before, the design reasoning / rationale effectiveness of the Rationale Capture Cycle is demonstrated by its ability to be easily applied, its non-intrusiveness nature and ability to improve architecture design quality. These concepts are operationalized through the following measures: ranking architecture designs in terms of quality, time spent to complete the case and the extent in which all rationale elements occur in the documentation.

7.2.2.1 Time

In order to measure the intrusiveness of the DR artefact, the time spent completing the case is also measured. The participants are asked to write down their start- and end time during the session. A student's t-test will be performed to measure if there is any significant difference between the times of the test group versus those of the control group.

7.2.2.2 Design Quality

During the long proposal phase it was deemed that carefully evaluating which architecture designs are 'better' is incredibly complicated. A work-around is applied by letting the architects themselves rank the architecture designs of their peers in terms of their quality. That way, the evaluation of architecture quality is delegated to the architecture practitioners themselves instead of the researcher. The criteria for the ranking are intentionally left implicit. However, the rankings represent the relative quality of an architecture design. The specific criteria are left to the architect's interpretation. The architects are allowed to spend a maximum of 30 minutes considering how to rank the assigned solutions. The rankings can be sent back to the researcher through conventional channels.

The ranking is done by both a fixed sum scale and pairwise ranking. Each architect is assigned 3 solutions from their peers. The architect has to select which solution was the better one and which one was worse. The architect can do so by dividing 100 points across the 3 solutions. Depending on where the allocated points are spent, each architect has ranked the solutions in the 'a > b > c' format. From those parameters, an automatic ranking is defined. From this matrix now can be counted how many times each solution has won versus the other solutions.

The same principle applies for the fixed sum scale, where another matrix will be drawn but instead of a binary win/loss format the total points given by the architects are filled in. From these rankings an overall ranking will be found by counting how many times a specific solution has 'won' a matchup. If every architect ranks 3 solutions, a total of 30 rankings can be filled in. However, you need at least 45 matchups to have every solution be matched with the others at least once. In order to fill in the missing values, another session will be planned with two architects to rank the other matchups. This design has one major benefit: it allows the architects to provide more information by dividing a 100 points across the solutions and it automatically generates a ranking in the amount of points each solution has received. Ultimately, you can generate two lists. One where the amount of points across test- and control group solutions are counted and another where the amount of matchup wins are counted. These two lists can then be correlated to see if there is any significant relationship between the point spending behaviour of architects and the final ranking of solutions. Additionally, each architecture will be reviewed using an architecture evaluation checklist. Using this checklist, each architecture can be ranked in terms of how they score when they are assessed. This ranking, done by the researcher, can then be compared to the other tests.

The rankings will also be fitted to the Bradley-Terry pairwise ranking model to generate another statistical result for another comparison. Elaboration on the Bradley-Terry model can be found in the next paragraph. To validate the end rankings, a sane-test session will be planned to see if a full ranking of the solutions by two external architects show similar results. The various evaluation tests are shown in Figure 12. Design quality ranking tests.

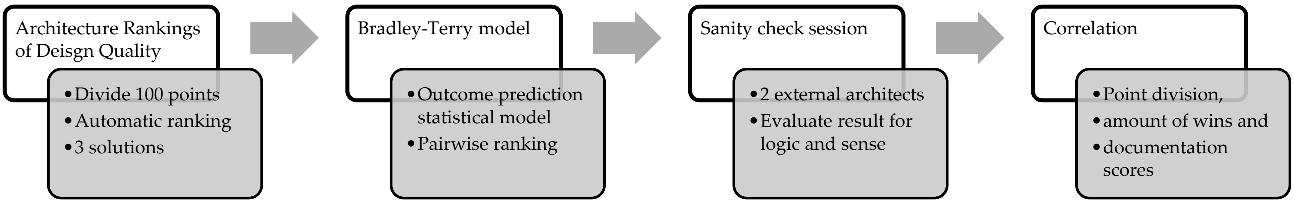


Figure 12. Design quality ranking tests

The idea of these multiple tests is that of cross-validation. Considering the sample is quite low and the tests are hard to generalize, multiple different tests are performed to evaluate and compare the results. Say, for example, if the solutions made by architects in the test group perform significantly well across all 4 different tests there is something to say about having demonstrated the effect of the rationale capture cycle reasoning model. This way, the potential for invalid results is minimized.

The goal of the ranking is to determine whether or not architects consistently rank solutions made with support of the artefact higher than the solutions who were not. To achieve this, pairwise comparisons will be made using the Bradley-Terry (BT) model (Bradley & Terry, 1952). The main goal of the BT model is to estimate the probability that an object ranks higher than another based on existing rankings. The BT model can be used to derive a full ranking. For example, it is very difficult for an architect to draft a ranking of 10 different solutions, but it is possible to compare a sample of pairs and say, for each pair, which solution was the 'better' architecture design.

The BT model defines a probability measure for each solution (between 0-1), which can be used to rank the solutions from higher to lower probabilities. The BT model is noted as follows:

$$P(i > j) = \frac{p_i}{p_i + p_j}$$

Where P is the probability that solution i ranks higher than solution j. p represents the design quality of solution i or j, estimated from the number of times i has 'won' a ranking. The rankings are based on the following Boolean logic statements, i.e. transitivity:

$$\begin{gathered} If\ x > y\ and\ y > z, then\ x > z \\ If\ x > y\ and\ y = z, then\ x > z \\ If\ x = y\ and\ y > z, then\ x > z \\ If\ x = y\ and\ y = z, then\ x = z \end{gathered}$$

Where x and y are the compared objects, > represents greater than and = represents an equivalence relation. The statistical test is done using XLstat and Microsoft Excel. XLstat offers a BT model extension that fits and tests the data. All data will be transformed into the following format:

Solution 1	Solution 2	Result
T2	C1	Win
C5	T2	Loss
C1	C3	Win
T5	C5	Loss
…	…	…

Table 6. Example data format BT model

The Bradley Terry model evaluates each matchup and continues to evaluate those wins with the matchups of other solutions. For example, it calculates the likelihood of T3 winning after T3 has lost to C5 whilst C5 has lost to T1 who in turn lost to T3. The model provides a full list of likelihood percentages, from which a full ranking can be derived.

7.2.2.3 Rationale Documentation

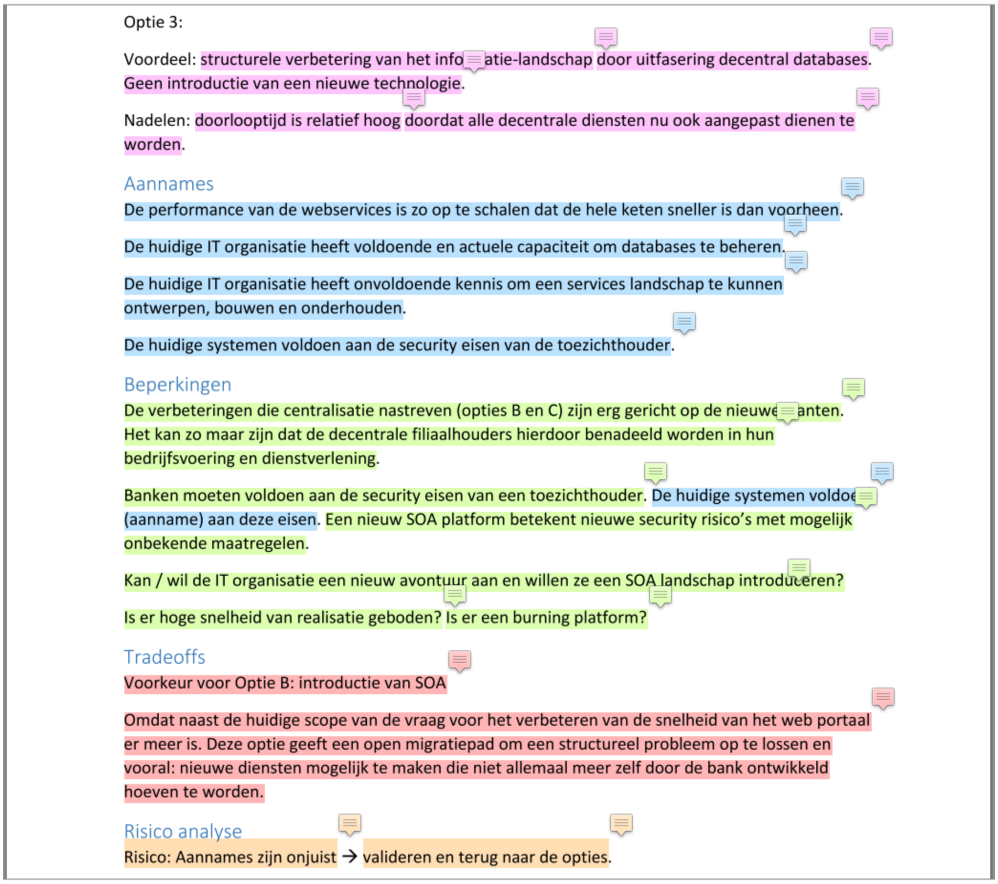
In order to identify separate rationale types for coding analysis, the decomposed elements of design rationale that surfaced during literature review are determined. For each of these elements the frequency in which they appeared are analysed and compared between the test- and control group. In order to measure how many times they appeared the documentation the participants produce during the experiment will be analysed and coded.

Rationale Type		Definition	Source
Design decision		The final decision that has been made during the reasoning process.	Tang (2011)
Problem analysis	Identification	Identifying the key issues or to be satisfied requirements in a design.	
	Definition	Defining and elaborating on the key issues and requirements in a design.	
Constraint analysis	Identification	Identifying the limitations placed on the design, either business or technical in nature.	Tang et al. (2006)
	Definition	Describing said limitations, elaborating on their nature and how the relate and materialize.	
Assumption analysis	Identification	The listing of unknown factors that might affect the design.	
	Definition	Describing said unknown factors, to what extent they may surface and how critical they are.	
Option analysis	Identification	Listing the various options and alternatives that might address the design problem.	Tang (2011)
	Definition	Exploring the various listed options, elaborating on they address the problem and explaining what elements they comprise of.	
Benefit analysis	Identification	The identification of the benefits a design option can deliver to satisfy the technical or functional requirements.	Tang et al. (2006)
	Definition	The description of how these benefits take effect, elaborating on how they originate and why they constitute a significant benefit to the design.	
Weakness analysis	Identification	The identification of the weaknesses of a design option, elements that an option cannot achieve.	
	Definition	The description of how these weaknesses take effect, elaborating on how they originate and why they constitute a significant adverse effect to the design.	
Trade-off Analysis	Identification	A trade-off exists when a design cannot satisfy all the requirements or constraints at the same time. It weighs and compares alternatives and its supporting rationales.	Bass et al. (2003)
	Definition	The description of the trade-off, which elements are weighed and compared, which element has won as opposed to the other and how this process took place.	
Risk Analysis	Identification	An unknown factor that can have negative implications on a design option.	Tang (2011)
	Definition	The description of the negative implications, to what extent they are harmful to the design and an elaboration on how they may impact the design.	
Evaluation		A feedback loop or validation from discourse to rethinking critical elements in the design rationale when presented with new insights or information.	McCall (2010)
Reflection		An analysis or verification of the architect's own reasoning process.	

Table 7. Rationale Coding Scheme

Important to note is that the 'analysis' segment of the rationale elements are split into *identification* and *definition*. The reason for that there is a difference between identifying a 'benefit', for example, and actually describing *why* that premise is beneficial and *how* the benefit materializes.

For a thorough coding analysis, these two have to be separated. An architect of the control group might, say, identify a similar amount of risks as another architect would. However, the Rationale Capture Cycle might stimulate the actual description and elaboration thereof. As such, the coding scheme has to be as specific to catch the nuance. A coded example of the rationale documentation can be found in Figure 13. Example coding on the following page.


Optie 3:

Voordeel: structurele verbetering van het informatie-landschap door uitfasering decentral databases. Geen introductie van een nieuwe technologie.

Nadelen: doorlooptijd is relatief hoog doordat alle decentrale diensten nu ook aangepast dienen te worden.

Aannames

De performance van de webservices is zo op te schalen dat de hele keten sneller is dan voorheen.

De huidige IT organisatie heeft voldoende en actuele capaciteit om databases te beheren.

De huidige IT organisatie heeft onvoldoende kennis om een services landschap te kunnen ontwerpen, bouwen en onderhouden.

De huidige systemen voldoen aan de security eisen van de toezichthouder.

Beperkingen

De verbeteringen die centralisatie nastreven (opties B en C) zijn erg gericht op de nieuwe klanten. Het kan zo maar zijn dat de decentrale filiaalhouders hierdoor benadeeld worden in hun bedrijfsvoering en dienstverlening.

Banken moeten voldoen aan de security eisen van een toezichthouder. De huidige systemen voldoen (aanname) aan deze eisen. Een nieuw SOA platform betekent nieuwe security risico's met mogelijk onbekende maatregelen.

Kan / wil de IT organisatie een nieuw avontuur aan en willen ze een SOA landschap introduceren?

Is er hoge snelheid van realisatie geboden? Is er een burning platform?

Tradeoffs

Voorkeur voor Optie B: introductie van SOA

Omdat naast de huidige scope van de vraag voor het verbeteren van de snelheid van het web portaal er meer is. Deze optie geeft een open migratiepad om een structureel probleem op te lossen en vooral: nieuwe diensten mogelijk te maken die niet allemaal meer zelf door de bank ontwikkeld hoeven te worden.

Risico analyse

Risico: Aannames zijn onjuist → valideren en terug naar de opties.

Figure 13. Example coding

7.3 Hypotheses

Hypotheses are defined for the various different measures and moments.

7.3.1 Hypothesis 1: Design Quality

H0: There is no difference in design quality (measured in assigned points) between designs produced by Sogeti architects from the test group and the control group when designing an architecture (moment 1).
H1: There is a difference in design quality (measured in assigned points) between designs of the test group and control group when designing an architecture (moment 1).

7.3.2 Hypothesis 2: Intrusiveness

H0: There is no difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

7.3.3 Hypothesis 3: Comprehensiveness

As mentioned before, the measure for artefact comprehensiveness is the extent in which all design rationale elements are addressed and present in the solutions of the participants. The design rationale elements are based off the artefact assembled during the MAA phase and are as follows: *problem identification, constraint identification, assumption recognition, option listing, benefit listing, weakness listing, trade-off analysis, risk analysis* and *reflection & evaluation*. For each of these design rationale elements hypotheses are defined in order to demonstrate to what extent the artefact is comprehensive. These hypotheses are valid for both experiment moments.

Total Rationale Frequency
H0: There is no difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Design Decisions
H0: There is no difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Problem analysis
H0: There is no difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Constraint analysis
H0: There is no difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Assumption analysis
H0: There is no difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Option analysis
H0: There is no difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Benefit analysis
H0: There is no difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Weakness analysis
H0: There is no difference in weakness analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture
H1: There is a difference in weakness analysis (measured in frequency of occu.rring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Trade-off analysis
H0: There is no difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Risk analysis
H0: There is no difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

Evaluation and reflection
H0: There is no difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

7.4 Case Design

The goal of the cases is to provide architects with an intricate and elaborate scenario where various different solutions exists. Architects need to be challenged with options and alternatives, however it has to be doable in a timespan of two hours. The case needs to mimic a real life scenario as close as possible so that architects are presented with a case that causes them to solve them as they would normally in their daily work routine. The reason for this is because if you present architects with a case that is too intricate, you force a stance and approach with increased rationale and reasoning. If a case is slightly routine, however, architects are more likely to solve the case as they would normally, without a forced rationale approach. This benefits the data and validity of its results.

The case is based on old Sogeti material and modified heavily in terms of architecture and design reasoning.

7.4.1 Case 1: Architecture Design

The main goal for this case was to provide architects with a case that mirrors real life scenarios, yet is equally challenging. It should also be feasible whilst maintaining the intricacy so rationale can be prevalent. If the case is too simple there is no room or need for rationale. If the case is too complex, you force rationale. Either extreme makes it so that the data is less valid. The case is based on architecture material provided by Sogeti in terms of subject and theme. The key challenges were modified into a case where multiple different approaches could be applied. Some key criteria include (non-exhaustive list):

- **Feasibility**: the case should be feasible within the 2 hour limit.
- **Challenge**: the case should be intricate enough to challenge an experienced architect and propose him or her with different approaches or alternative options.
- **Clarity**: the case should be clear and simple to understand in order for the focus of the architect should revolve around solving the case and designing an architecture instead of figuring out the material.
- **Realism**: as mentioned before, the case should mirror real scenarios.

See appendix 5 for the full case.

7.4.2 Case 2: Request for Change (RfC)

The same key criteria of the first case apply for the second. As opposed to case 1, the second case involves around architects having to implement a change in an existing architecture. This is interesting because the hypotheses now revolve around to what extent rationale helps to implement a change, instead of to what extent rationale helps design a complete architecture. In other words, does rationale indeed help new architects (facing an architecture from a peer) understand and work with the architecture? And how does this difference manifest in the models, documentation and survey? See appendix 6 for the full case and the next chapters for the results and analyses.

7.4.3 Rationale Documentation Template

As mentioned before, the output of the experiments produced by the participants needs to be controlled as much as possible. If the output is uniform in its structure, the observed change in effect can be attributed more so to manipulating the independent variable. To this end, a rationale documentation template is made to provide the architects with handles on how they should document the rationale of their architecture design. The complete template can be found in appendix 8.

7.4.4 Survey Design

The main purpose of the survey is to provide an outlet for the architects to express feedback and give input in order to validate the artefact. The complete survey can be found in appendices 5 and 6. The main philosophy behind the design was to provide multiple instruments by which architects can express their ideas and thoughts and to trigger enough choice moments for the architects to reason.

7.4.5 Case Validation

The case is validated by the first University supervisor and company supervisor. The main criteria were its feasibility, challenge, syntax, ambiguity and the extent in which it mirrored real life scenarios. Some key feedback points were:

- **Guidance**: in some areas, the case did not provide enough guidance to the participants in order for them to focus on the case material itself. For example, an ArchiMate meta-model was added to the case in order to clearly describe the exact models the architect needed to produce.
- **Realism**: the case does provide a real life scenario, where the board of a bank does not clearly know what they want or how they get there. However, in the typical architecture job, the architect has the ability to ask questions and perform research in order to find out what is required. Due to the nature of a short case, it is not possible to describe the whole organization in a few pages. To address this, the case now mentions the architect has to design out of their own ideal scenario. What would the architect do, if given unlimited resources? This also supports the architect in making decisions, allowing for more rationale documentation to be potentially present.
- **Software instructions**: the case did not always provide enough concrete software instructions on how to use 'Archi'. Extra instructions were added in order to guarantee the architects understood the software.

7.4.6 Pilot Test

A pilot test session was held with 4 Master students from Utrecht University in order to execute a dry run of the experiment. The main goal was to eliminate any ambiguity and remove overlooked mistakes. The key findings were:

- **Length**: during the pilot test it was clear the students did not have enough time in order to complete the entire case. The students had to spend the full duration of the case. The output required was shortened, in response. However, the challenge of limited time also requires an architect to make decisions in terms of what to pursue. Also, architects have more experience. Because of these reasons, some challenge and length was kept.
- **Challenge**: the students found the case to be challenging and lengthy. However, due to the same aforementioned reasons, the difficulty was not lowered.
- **Syntax**: the students found some of the terminology used to be confusing. Due to this, many definitions and terms were standardized in the case.

''Designing the experiment around two contact moments allows for a richer research. If not done, there would be a demonstration of how to stimulate reasoning without a follow-up of what this stimulus results in. Vice versa, a demonstration of what effect stimulated reasoning produces is less valuable without a demonstrated artefact that can incite it.''

8 RESULTS

This chapter outlies the varying results from the envisioned tests and measures that were executed during the research project. The goal of the various performed tests were to establish whether or not the use of the Rationale Capture Cycle notably influences the architecture design process. This chapter only covers the found results. Chapter 9, Analysis, attempts to gain insight into how this influence materializes and what findings could be derived from it. Chapter 10 summarizes the main conclusions of this data.

8.1 Moment 1: Architecture Design

8.1.1 Participants

10 architects participated in moment 1 of the experiment session. Their demographic data can be seen in the following table. The exact employers are explicated in a separate table in order to ensure the anonymity of the participants. The respective ages are also removed from this table for the same reason.

Architect	Title	Experience in IT (yrs)
C1	Information architect	13
C2	Sr. information architect	37
C3	Sr. information architect	38
C4	Sr. business analyst	33
C5	Sr. information architect	34
T1	Sr. information architect	29
T2	Business analyst / sr. information architect	35
T3	Enterprise architect	22
T4	ICT architect	22
T5	Sr. information architect	36

Table 8. Participant's demographic data moment 1

Current employers
Major financial institution
Large government agency
Real state organization
Major public transit operator
International IP connectivity provider
Large government agency
International technology solutions company
Major government agency
Large educational institution
International telecommunications business

Table 9. Employers of moment 1

8.1.2 Time

During the experiment session the total completion time per participant was recorded. The results can be found in Table 10. Experiment session durations.

Participant	Duration	Time (min)	Participant	Duration	Time (min)
C1	13:30 – 15:33	123	T1	18:15 – 20:30	135
C2	13:30 – 15:30	120	T2	18:15 – 20:30	135
C3	13:30 – 15:00	90	T3	13:30 – 15:30	120
C4	18:15 – 20:30	135	T4	13:30 – 15:39	129
C5	14:12 – 15:20	68	T5	13:30 – 15:41	131
Average		107.2	Average		130
Total		**536**	**Total**		**650**

Table 10. Experiment session durations moment 1

The experiment duration was monitored per participant. However, as virtually all participants stayed for the full duration of the session no concrete information can be derived from analysing the duration of the sessions. The only exceptions, and outliers of this data, are participant C3 and C5 who finished notably earlier than others regardless of group. When averaged, the control group spent 22.8 minutes less. In total, the control group spent 114 less minutes completing the session.

8.1.3 Design Quality

8.1.3.1 Ranking

The first test to gauge the effect of the Rationale Capture Cycle on the design quality of the architecture models and corresponding documentation was to assign the solutions to the 10 participants and have them rank the solutions. Each architect has been assigned 3 solutions and is given 100 points to allocate between them. The solutions are randomly assigned between the 10 participants with the help of Research Randomizer: a free, browser-based randomization tool. Also, Microsoft Excel is used for exclusion statements and randomization. Excel COUNTIF statements are used in order to confirm equal rankings between the groups, participants and solutions respectively.

The random allocation of solutions must adhere to the following limitations:

- Each participant can only rank 3 solutions.
- Each participant is allowed a maximum of 100 points, and all points must be spent. No ties can be made, i.e. the participants are not allowed to allocate points equally.
- Each participant is assigned at least 1 solution of both the test- and control group, i.e. a participant cannot be given just test group solutions.
- No participant can rank his own solution.
- Each solution must be ranked an equal number of times divided between the 10 architects, which is 3 times.
- The test- and control group must have an equal number of solutions, 15 each.
- The test- and control group must have an equal number of solutions that belong to either the test- and control group, i.e. the test / control solution ratio must be equal in either group.
- A combination of solutions must be unique, i.e. no given ranking can occur more than once.

Adhering to the requirements above, a solution matrix is made. This matrix can be found in appendix 7. Adhering to the limitations mentioned above, the assignment of solutions including results can be made.

Participant	Assigned solutions	Rank - Solution	Rank - Points	Participant	Assigned solutions	Rank - Solution	Rank - Points
C1	T1, T3, C5	1. C5 2. T1 3. T3	1. 50 2. 30 3. 20	T1	T2, C1, C2	1. T2 2. C1 3. C2	1. 50 2. 30 3. 20
C2	T3, T4, C4	1. T3 2. C4 3. T4	1. 45 2. 35 3. 20	T2	T3, C1, C3	1. T3 2. C1 3. C3	1. 50 2. 30 3. 20
C3	T5, T2, C4	1. T5 2. T2 3. C4	1. 50 2. 30 3. 20	T3	T1, T4, C4	1. T4 2. T1 3. C4	1. 75 2. 20 3. 05
C4	C2, C5, T5	1. C2 2. C5 3. T5	1. 50 2. 30 3. 20	T4	T5, T1, C1	1. T1 2. C1 3. T5	1. 45 2. 35 3. 20
C5	C2, C3, T4	1. T4 2. C3 3. C2	1. 65 2. 20 3. 15	T5	T2, C5, C3	1. T2 2. C5 3. C3	1. 50 2. 30 3. 20

Table 11. Results ranking moment 1

The following table represents the results per solution instead of the participant. A win is counted if the solution is ranked higher than another one in the same set. Two wins are counted if the solution is ranked first in the set as it will have beaten the second and third solution in that set.

Solution of	Wins	Points	Solution of	Wins	Points
C1	3	95	T1	4	95
C2	2	85	T2	5	130
C3	1	60	T3	4	115
C4	1	60	T4	4	160
C5	4	110	T5	2	90
Total	**11**	**410**	**Total**	**19**	**590**

Table 12. Results per solution ranking moment 1

The first observable difference is that the solutions made with the support of the Rationale Capture Cycle (test group) are awarded more points (590) than the control group solutions (410). The test group solutions also won more (19) than the control group solutions did (11). For further insight the solutions are ranked according to their evaluations.

The following table represents the results in order of points, first, and in order of amount of wins second.

Rank #	Solution of	Points	Wins
1	T4	160	4
2	T2	130	5
3	T3	115	4
4	C5	110	4
5	T1	95	4
6	C1	95	3
7	T5	90	2
8	C2	85	2
9/10	C3 / C4	60	1

Table 13. Solutions ranked moment 1

First and foremost, the top 3 solutions that were awarded the most points are all test group solutions. Additionally, the top 5 solutions are all test group solutions with the exception of C5. C5 did relatively well compared to the results of his peers in his research group. Out of the bottom 5 solutions, 4 are made by the control group participants.

8.1.3.2 Bradley Terry Probability Model

As described in paragraph 7.2.2, Measures, a generalized Bradley Terry (BT) is fitted. All data is transformed into a ''*T1 vs C1 = Win*'' format. In the table below, some descriptive statistics are explicated. Here, the total matches, wins, losses and percentages are calculated by the BT model.

Solution	Total matches	Wins	Losses	% wins	% losses	% ties
C1	6	3	3	50	50	0
C2	6	2	4	33.33	66.67	0
C3	6	1	5	16.67	83.33	0
C4	6	1	5	16.67	83.33	0
C5	6	4	2	66.67	33.33	0
T1	6	4	2	66.67	33.33	0
T2	6	5	1	83.33	16.67	0
T3	6	4	2	66.67	33.33	0
T4	6	4	2	66.67	33.33	0
T5	6	2	4	33.33	66.67	0

Table 14. Summary statistics BT model moment 1

The following table provides the estimated parameters, i.e. the extent in which the factors influence the probabilities of winning. For example, solution T2 has a very strong influence (2.525) on winning a certain match. The total of estimates are 10.

Solution	Estimate	Standard deviation	Lower bound	Upper bound
C1	0.721	0.418	-0.097	1.540
C2	0.342	0.247	-0.142	0.826
C3	0.186	0.179	-0.165	0.537
C4	0.199	0.205	-0.202	0.600
C5	1.671	0.745	0.210	3.132
T1	1.561	0.708	0.174	2.948
T2	2.525	0.906	0.749	4.300
T3	1.471	0.669	0.160	2.782
T4	0.957	0.466	0.045	1.869
T5	0.368	0.274	-0.169	0.905

Table 15. Estimated parameters BT model moment 1

Using the estimates above, the BT model calculates the winning probabilities per matchup based on the existing wins and losses of the dataset. The rows represent the likelihood of winning and the columns represent the likelihood of losing.

Solutions		**Likelihood of losing**										
		C1	C2	C3	C4	C5	T1	T2	T3	T4	T5	Average
Likelihood of winning	C1	**0.000**	0.678	0.795	0.784	0.301	0.316	0.222	0.329	0.430	0.662	**0.452**
	C2	0.322	**0.000**	0.648	0.632	0.170	0.180	0.119	0.189	0.263	0.482	**0.300**
	C3	0.205	0.352	**0.000**	0.483	0.100	0.107	0.069	0.112	0.163	0.336	**0.193**
	C4	0.216	0.368	0.517	**0.000**	0.107	0.113	0.073	0.119	0.172	0.351	**0.204**
	C5	0.699	0.830	0.900	0.893	**0.000**	0.517	0.398	0.532	0.636	0.820	**0.622**
	T1	0.684	0.820	0.893	0.887	0.483	**0.000**	0.382	0.515	0.620	0.809	**0.609**
	T2	0.778	0.881	0.931	0.927	0.602	0.618	**0.000**	0.632	0.725	0.873	**0.697**
	T3	0.671	0.811	0.888	0.881	0.468	0.485	0.368	**0.000**	0.606	0.800	**0.598**
	T4	0.570	0.737	0.837	0.828	0.364	0.380	0.275	0.394	**0.000**	0.722	**0.511**
	T5	0.338	0.518	0.664	0.649	0.180	0.191	0.127	0.200	0.278	**0.000**	**0.314**

Table 16. Winning probabilities BT model moment 1

Using the estimated likelihood parameters, a full ranking can be derived from the data based on the likelihood of winning.

Rank #	Solution of	Estimated parameter (out of 10)	Average winning probability
1	T2	2.525	0.697
2	C5	1.671	0.622
3	T1	1.561	0.609
4	T3	1.471	0.598
5	T4	0.957	0.511
6	C1	0.721	0.452
7	T5	0.368	0.314
8	C2	0.342	0.300
9	C4	0.199	0.204
10	C3	0.186	0.193

Table 17. Solutions ranked BT model moment 1

The first notable result is the fact that T2, according to the BT model, is an incredibly well performing solution. The high estimated parameter and average winning probability implies the solution is very likely to win against any other solution. This result is based on the wins it obtained versus other solutions and the wins and losses of those solutions against others. If we look into our data, the result makes sense. T2 is the only result that obtained 5 wins and has therefore the highest winning percentage of 83.33%. It also beat C5, which according to our data is the runner up. The averaged results can be seen in the following table, where it is clear the test group averages win out versus the control group scores.

Research Group	Sum of likelihood (out of 10) (estimated parameter)	Average winning probability
Control group	3.119	0.354
Test group	6.881	0.546

Table 18. BT model compared moment 1

All solutions being equal, the sum of likelihood for each solution would be 1. There are 10 solutions, so the sum of likelihood for all participants would be 10. In other words, if all solutions had the same scores and same amount of influence in determining the likelihood of winning a certain match all estimated parameters would be 1. Here, taking the data of the BT model into account, the sum of likelihood is totalled for each research group. The test group participants account for a much larger portion (6.881 / 10) than the control group (3.119 / 10). This shows that, based on the rankings, the test group has a higher degree of influence over winning matchups. This means that those solutions perform better during the ranking and are evaluated higher by the architects.

8.1.4 Rationale Documentation

In addition to ranking the solutions by the architects themselves and through the BT model, the architects were asked to provide additional documentation alongside their design. This documentation consisted of providing a description of the model and elaborate justification, i.e. rationale, for their design decisions. This paragraph consists of two parts.
First, the documentation is analysed using the coding scheme based on found literature, where all rationale elements are coded and their frequencies of occurrence measured. The results of the coding analysis of the first experiment session can be found in the following tables. The table with the participants labelled as a 'T' are those of the test group (participants who used the Rationale Capture Cycle model). The participants labelled as a 'C' belong to the control group. The numbers in the cells represent the frequency of occurrence.

Rationale Type		C1	C2	C3	C4	C5	*AVG*	**Total**	T1	T2	T3	T4	T5	*AVG*	**Total**
							per DR element							per DR element	
Design decision		2	3	2	2		*2.25*	**9**	4	8	3	1	4	*4*	**20**
Problem analysis	Ident.		1		3	4	*2.67*	**8**	3	4	3	4	2	*3.2*	**16**
	Descr.				2	4	*3*	**6**	2		4		1	*2.33*	**7**
Constraint analysis	Ident.											5		*5*	**5**
	Descr.											1		*1*	**1**
Assumption analysis	Ident.											5		*5*	**5**
	Descr.													*0*	**0**
Option analysis	Ident.	2	2	2	2	6	*2.8*	**14**	3	10	3	3	1	*4*	**20**
	Descr.	1	1	4	2	5	*2.6*	**13**	6	2	1	1	2	*2.4*	**12**
Benefit analysis	Ident.	5	6	3		4	*4.5*	**18**	1	6	3	7	1	*3.6*	**18**
	Descr.	1					*1*	**1**				3	1	*2*	**4**
Weakness analysis	Ident.											4		*4*	**4**
	Descr.											4		*4*	**4**
Trade-off Analysis	Ident.											1		*1*	**1**
	Descr.											1		*1*	**1**
Risk Analysis	Ident.									1		1		*1*	**2**
	Descr.									1		1		*1*	**2**
Evaluation														*0*	**0**
Reflection														*0*	**0**
Average		*2.2*	*2.6*	*2.75*	*2.2*	*4.6*	**CTRL total 69**		*3.17*	*4.57*	*2.83*	*2.8*	*1.71*	**TEST total 122**	
Total		**11**	**13**	**11**	**11**	**23**			**19**	**32**	**17**	**42**	**12**		

Table 19. Documentation results moment 1

The table shows that the participants of the test group totalled a number of 122 coded rationale elements. These were individual elements of rationale, elaborating on why they made certain design decisions. Interestingly, there are three elements that have no occurrence in the documents across the various participants: *evaluation*, *reflection* and *assumption definition*. These number 3 out of a total of 19 identified rationale types, which means roughly 16% of rationale types were omitted. This does not necessarily mean that they did not do so, solely that the documentation did not demonstrate it explicitly. Other elements proved to appear very few times, like *risk analysis* and *trade-off analysis*. Another notable difference is the difference in *total* rationale frequencies found when coding the documentation supplied by the architects. The test group totalled a number of 122 versus the 69 coded types of the control group, which is nearly double. The frequencies coded in the documentation of the test group is an increase of nearly 77% compared to the documentation of the control group. Also, the spread of which rationale types occur are more comprehensive in the test group.

The participants of the test group only omitted 3 rationale types (16/19 were present) as opposed to the 12 of the control group (7/19 were present). Some interesting combinations of identifications and definitions are notably present, like the difference between *problem identification* and *problem definition* (16 – 7 = 9), *option identification* and *option definition* (20 – 12 = 8) or *benefit identification* and *benefit definition* (18 – 4 = 14). These results show that the participants of the test group would identify, but not elaborate on, *problems*, *options* and *benefits* 9, 8 and 14 times respectively. When grouped and averaged, that means that in 54% of cases where there either a problem, option or a benefit it would be solely identified and not further explained. Another thing of note is the vast differences in frequencies between the participants themselves. Participant T4 is responsible for 42 coded elements (34.4% of total) versus the 12 of participant T5 (9.8% of total). The next closest in terms of contribution is T2, who provided 32 (26.2% of total) rationale elements. T2 and T4 together contribute 74 coded rationale types, which is 60.7% of the total. If we split the identification and definition of rationale types, the following results surface. For the control group 31 out of 71 rationale types were defined when identified. For the control group that is only 20 out of 40 types. In the case of defining and elaborating the control group scores better, even though the overall results are lower in terms of volume. The difference, however, is quite low at only 6.3%. See the following table.

	C	T	T-C
Identification of rationale types	40	71	31
Description of rationale types	20	31	11
Elaboration percentage	50%	43.7%	-6.3%

Table 20. Identification and description difference moment 1

In order to better understand the individual rationale elements, the results have to be aggregated and compared per rationale type. The results can be seen in the following table.

Rationale Type		**Code**	**C (#)**	**T (#)**	**Increase (f)**	**% increase (per rationale type)**	**% increase (of overall increase in frequency)**
Design decision		DD	9	20	11	122.22%	20.75%
Problem analysis	Identification	PI	8	16	8	100%	15.09%
	Description	PD	6	7	1	16.67%	1.89%
Constraint analysis	Identification	CI	0	5	5	-	9.43%
	Description	CD	0	1	1	-	1.89%
Assumption analysis	Identification	AI	0	5	5	-	9.43%
	Description	AD	0	0	0	-	0%
Option analysis	Identification	OI	14	20	6	42.86%	11.32%
	Description	OD	13	12	-1	-7.69%	-1.89%
Benefit analysis	Identification	BI	18	18	0	-	0%
	Description	BD	1	4	3	300%	5.66%
Weakness analysis	Identification	WI	0	4	4	-	7.55%
	Description	WD	0	4	4	-	7.55%

Rationale Type		Code	C (#)	T (#)	Increase (f)	% increase (per rationale type)	% increase (of overall increase in frequency)
Trade-off Analysis	Identification	TI	0	1	1	-	1.89%
	Description	TD	0	1	1	-	1.89%
Risk Analysis	Identification	RI	0	2	2	-	3.77%
	Description	RD	0	2	2	-	3.77%
Evaluation		EV	0	0	0	-	0%
Reflection		ER	0	0	0	-	0%

Table 21. Difference per rationale type moment 1

Overall design decisions sees the largest increase, increasing from 9 (control group) to 20 (test group). *Problem identification* sees the second largest increase, increasing from 4 (control group) to 14 (test group). Following the list *constraint identification, assumption identification and option identification* show the largest increase in absolute frequencies.
The major contributors to the overall increase of 53 are *design decisions, problem identification, constraint identification, assumption identification, option identification* and *weakness analysis*. Together, they account for 66% of the total difference in rationale type frequencies. It is interesting to note that the increase in coded rationale types seem to be widespread, and not due to isolated elements. There are 2 types that contribute 3.77%, 2 types that contribute 7.55% and 2 types that contribute 9.43%. There are 4 types that showed no increase, assumption definition, benefit identification, evaluation and reflection. 14 other types have all shown an increase. Interestingly, one rationale type actually decreased. The control group actually documented more on a single category, *option definition, although the decrease was only 1 single element.* Another interesting dimension is to look into the order and sequence in which the rationale types appear. Analysing this structure of the documentation and comparing it with the structure of the model might provide more insight into the influence of the model. See the following tables.

Sequence	C1	C2	C3	C4	C5	Total	T1	T2	T3	T4	T5	Total
Problem → Option sequence				3	1	**4**	2	4	2	3		**11**
Option → Benefit / Weakness sequence	4	1	2		3	**10**	1	5		3	2	**11**
Benefit / weakness → Assumption sequence						**0**				4		**4**
Assumption → Constraint sequence						**0**				4		**4**
Constraint → Risk sequence						**0**				2		**2**
Risk → Trade sequence						**0**				2		**2**
Total	**4**	**1**	**2**	**3**	**4**	**14**	**3**	**9**	**2**	**18**	**2**	**34**

Table 22. Rationale type sequence frequencies moment 1

Considering the overall rationale frequency is higher amongst the test group you would expect to attribute this to the Rationale Capture Cycle. The individual rationale types are looked at to identify in which context they appear, i.e. with which elements they appear together. The sequences represent the sequential order of the rationale types as they appear in the Rationale Capture Cycle. The control group accounted for a total of 14 rationale types that appeared together as they would appear following the model. The test group noted a total of 34 matches. This is an increase of 143% and seems to confirm that the overall increase in rationale frequencies is, at least in part, due to the model. Another interesting aspect is look at specific rationale types and see in what context they occur. For instance by looking at the relationships between commonly related types. See the following tables.

Rationale type		C1	C2	C3	C4	C5	Total	T1	T2	T3	T4	T5	Total
Problem analysis	Problem identification		1		3	4	**8**	3	4	3	4	2	**16**
	Problem definition				2	4	**6**	2		4		1	**7**
Option analysis	Option identification	2	2	2	2	6	**14**	3	10	3	3	1	**20**
	Option definition	1	1	4	2	5	**13**	6	2	1	1	2	**12**
Benefit analysis	Benefit identification	5	6	3		4	**18**	1	6	3	7	1	**18**
	Benefit definition	1					**1**				3	1	**4**
Total		**9**	**10**	**9**	**9**	**23**	**60**	**15**	**22**	**14**	**18**	**8**	**77**

Table 23. Problems, options and benefits relationships moment 1

As problems, options and benefits follow a clear structure and relationship you would expect to see this in the results. However, as shown in the previous table, that is not always reflected. For instance, the participants of the control group suggested 14 design options for 8 problems which is 1.75 options per problem on average. The test group, who managed to identify 16 problems, only suggested 20 design options. This is roughly 1.25 options per problem. Though the test group found more problems, no extra design options were suggested as a result. Relatively, these numbers show that the control group has a better design option per problem ratio. However, this may be due to architects of the test group identifying more related problem elements to a certain design problem. These problem elements are coded separately which causes a spike in problem identifications. However, all these problem identifications may be addressed by a single design option. The results do not tell us whether or not the spike in problem identifications are caused by the test group failing to offer enough design options or simply demonstrating greater rigor by identifying more aspects to a single problem. Another dimension to take into account is the background of the participant.

The results below feature the performance when background and recent work content are taken into account. The first table crosses the results with years of work experience.

Rationale Type frequency	C	T
Years of work experience (average)	31	28.8
Rationale type frequency	69	122
Rationale type frequency per work year	2.23	4.24

Table 24. Rationale type frequency per work year moment 1

The control group has, on average, 2.2 more years of experience in the IT industry. This is, however, a relatively small difference on the total of work experience to explain the increase in frequency. If the control- and test group split is ignored and the architects are sorted on their descending work experience, the following results surface. See Table 25. Rationale type frequency and work years.

Participant	Work experience (yrs)	Rationale type frequency
C1	13	11
T4	22	42
T3	22	17
T1	29	19
C4	33	11
C5	34	23
T2	35	32
T5	36	12
C2	37	13
C3	38	11

Table 25. Rationale type frequency and work years moment 1

One can expect there to be a positive relationship between work experience and rationale type results. An experienced architect should perform better than inexperienced architects. View the graph on the following page.

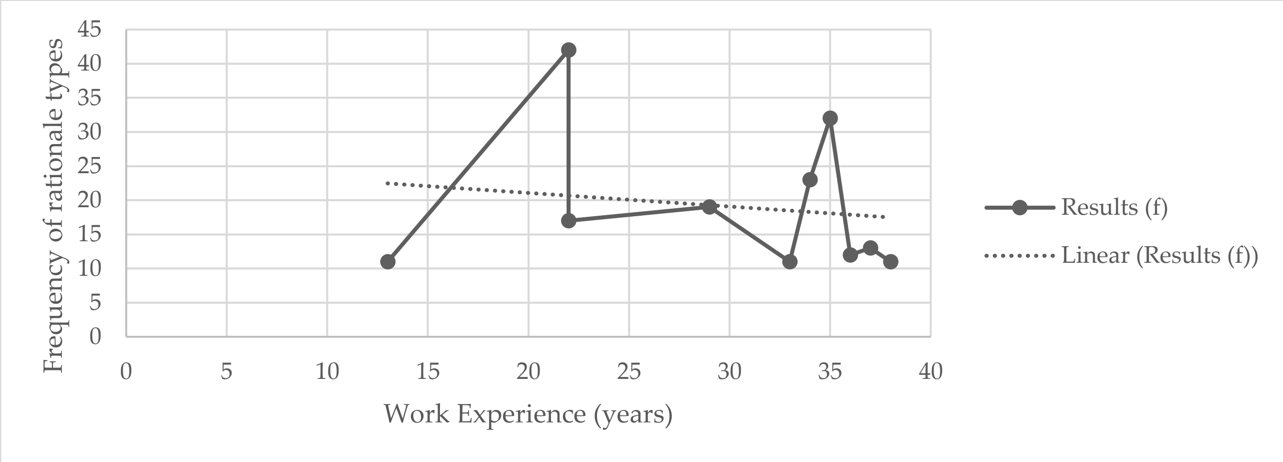


Figure 14. Results by work experience moment 1

As the graph shows, the relationship between work experience (x-axis) and the frequency of rationale types that were identified (y-axis) is actually negative. The linear (dotted line) of the results shows a downward slope. This implies that if you have less experience you will do comparatively better during the experiment. However, there is too little data to statistically confirm this is not due to chance. The results may simply show that the younger participants (with less experience) were more likely to adopt the Rationale Capture Cycle or were more familiar with the architecture software during the experiment and thus had more time to perform better.

To obtain better insight into this question it is interesting to look at the average age of the participant and compare it with their performance. Crossing the data with the age of the participants yields the following result.

Rationale Type frequency	C	T
Age (average)	57.6	54.8
Rationale type frequency	53	122
Rationale type frequency per year of age	0.92	2.23

Table 26. Rationale type frequency per year of life moment 1

The control group is, on average, 2.8 years older than the test group. This is, however, a relatively small difference on the total to explain the increase in frequency. If the control- and test group split is ignored and the architects are sorted on their respective descending ages, the following results surface. See the following table.

Participant	Age (years)	Rationale type frequency
C1	42	11
T4	47	42
T3	48	17
T2	59	32
C5	60	23
T1	60	19
T5	60	12
C2	62	13
C3	62	11
C4	62	11

Table 27. Rationale type frequency and age moment 1

The results are transformed into a graph in the following figure.

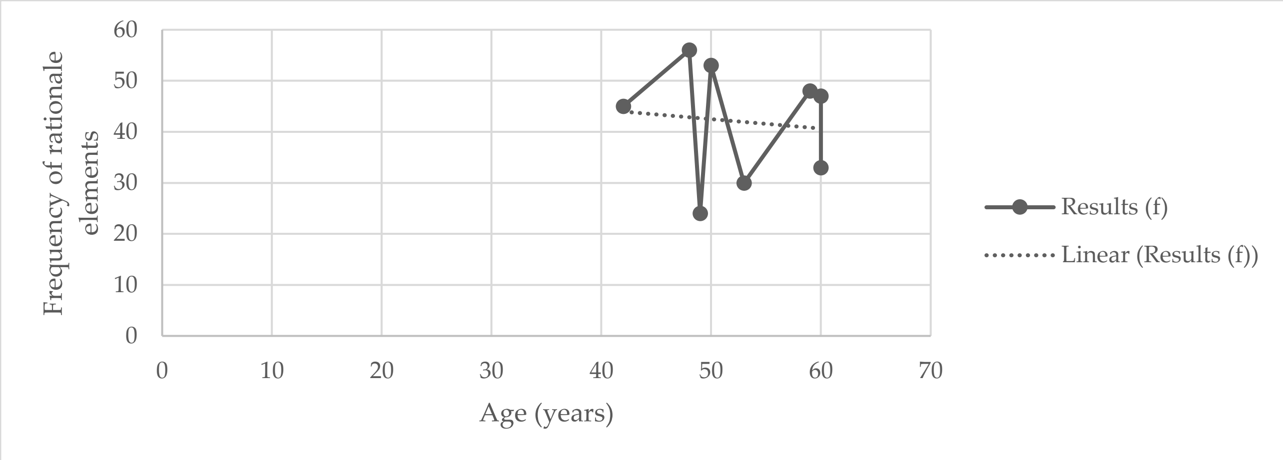


Figure 15. Results by age moment 1

As the graph shows, the relationship between the age of the architects (x-axis) and the frequency of rationale types that were identified (y-axis) is actually negative. The linear (dotted line) of the results shows a downward slope. This implies that if you are younger will do comparatively better during the experiment. However, there is too little data to statistically confirm this is not due to chance, especially considering the ages of the architects are too similar. The results may simply show that the younger participants were more likely to adopt the Rationale Capture Cycle or were more familiar with the architecture software during the experiment.

The following table shows the difference when the results are compared between the official titles the architects hold.

Role	**(Senior) Information architects**						**Other (enterprise architect, ICT architect, business analyst)**			
	T1	**T5**	**C1**	**C2**	**C3**	**C5**	**T2**	**T3**	**T4**	**C4**
Rationale type frequency	19	12	11	13	11	23	32	17	42	11
Total	**89**						**102**			

Table 28. Rationale type frequency difference per title moment 1

The information architects perform comparatively worse than the other participants, even when they number one more. The non-information architects perform slightly better. However, as most of the other architects were assigned to the test group the results do not demonstrate any significant difference due to title alone, as the Rationale Capture Cycle might be the influencing factor. Also, 3 out of 4 architects of the 'other' category hold the title of information architect as a secondary role. This makes the comparison even trickier.

The following table splits the architects into two categories again. This time, the distinction is made between architects who hold a more guiding role versus architects who perform operational modelling work. This distinction is made on the basis of whether or not the architect has had demonstrable experience in actually modelling and operational work instead of solely guiding architects and requirements conversations. In principle, all architects are of senior experience and thus have a more guiding role. Similarly, all architects model and do operational work as well. The distinction is made between whether or not the architect has had demonstrable operational work as a major responsibility in recent work experience as per the CV work description.

Work experience	**Guidance role**					**Modelling / operational role**				
	T1	**T3**	**C2**	**C3**	**C4**	**T2**	**T4**	**T5**	**C1**	**C5**
Rationale type frequency	19	17	13	11	11	32	42	12	11	23
Total	**71**					**120**				

Table 29. Rationale type frequency difference per recent work experience moment 1

Interestingly, the architects who fulfil a more operational role, i.e. architects who have demonstrable modelling experience in the past few years as per CV, show a vast increase in frequency. The difference of 71 versus 120, which is an increase of 69%. This might be due to the fact that modelling was a significant portion of the case and proved easier for those architects whom hold recent relevant experience. However, as most of the other architects were assigned to the test group the results do not demonstrate any significant difference due to title alone, as the Rationale Capture Cycle might be the influencing factor.

8.1.5 Survey

At the end of the survey, participants of both groups were asked to provide feedback and additional thoughts regarding the Rationale Capture Cycle, the case and the research overall. The major questions regarding the Rationale Capture Cycle model concerned added value, effectiveness, readability, usability, intuitiveness & ease of use.

- Added value & effectiveness: 2 out of 5 architects mention the model not having too much added value during the architecture design process. Out of these two, one architect mentions that is due to the case being too short and not providing large enough decisions for the model to be of use. The other architect offers similar reasoning, stating that the case offers the architects a scenario where not a lot of choices have to be made yet. Another architect mentioned a completely different story: the model helped him capture rationale more easily, especially when the model made him capture assumptions that he either would not have done. He also came up with another option while concentrating on the model that he normally would've omitted. He also found that the model adds structure to the process which is desperately needed. The fourth architect mentions something similar: the model helped in terms of providing structure to the process. It also guided him in the documentation of rationale and it made it easier to keep track of the documentation and reasoning process. The final architect mentioned that he did not have time to effectively use the model as he was too busy with using the Archi software and understanding the case.
- Readability & usability: 4 out of 5 architects mention that the model was easy to read and understand. 1 architect mentions the cycle was confusing as there is no real end to the process.
- Intuitiveness & ease: 4 out of 5 architects mention that the model did not intrude on the design process and was easy to use. 1 architect did not use the model at all due to focus on the software and the case. The last architect mentioned the model asks for a too high investment of time which is only realistic if the case would offer choice moments of larger size.

The participants were also asked to share their thoughts regarding the case. As mentioned before, 4 out of 10 architects mention that the case did not accurately mimic a real life scenario as the architect normally has a chance to ask and research the scenario. Unfortunately, this is inherent to the nature of a case where the architect is forced to make many assumptions. 5 out of 10 architects mention that they did not have enough time to complete the case, uttering that their models and documentation were unfinished by the time the session ended. 1 architect mentioned he rather liked the case, as it demonstrated the realistic scenario of a board that does not accurately know what it wants.

8.1.6 Significance Testing and Hypotheses

Aggregating the results from the previous chapter, some notable differences can be found between the control- and test group. Also, additional remarkable results were found when crossing the data with various other statistics. In this paragraph, all the comparisons and identified differences are tested for statistical significance. Per comparison, the distinction is made between performing a student's t test or Mann Whitney's U test. If the data adheres to the following 5 assumptions a student's t test will be chosen: a continuous dependent variable, independent categorical independent variables, independence of observation, no outliers, normally distributed dependent variable and homogeneity of variances. The Shapiro-Wilk test will be performed to test the data for normality. Levene's test will be performed to test for homogeneity of variances, if the resulting p value is lower than 0.05 the data is assumed significant and the variances are significantly different. If these assumptions are not met, Mann Whitney's U will be performed, which is the nonparametric variant for comparing means of different groups. All comparisons adhere to a significance level of 5%, i.e. the result is significant when $p < 0.05$. In the case of the Shapiro-Wilk test, the data is considered normal if $p > 0.05$. The results that are bold are significant. Some rationale types had no comparison data due to the control group omitting them completely. Only the rationale types that occurred in both research groups can be compared and tested for significance.

Comparison		Result / difference (Test group – control group)	Shapiro-Wilk test		Levene's test	Student's *t*	Mann Whitney's *U*
			C	T			
Ranking points		180	.410	.635	**.630**	.055	-
Ranking wins		8	.421	.135	.472	.069	-
Time		22.8 minutes	.485	.220	**.008**	-	.138
Total difference in rationale types		53	**.003**	.512	-	-	.072
Design decision		11	.135	.537	.345	.114	-
Problem analysis	Identification	8	.254	.314	**.035**	-	.127
	Description	1	**.046**	.314	-	-	.736
Constraint analysis	Identification	5	-	**.000**	-	-	.317
	Description	1	-	**.000**	-	-	.317
Assumption analysis	Identification	5	-	**.000**	-	-	.317
	Description	No comparison possible (0 difference)					
Option analysis	Identification	6	**.000**	**.027**	-	-	.326
	Description	-1	.254	**.023**	-	-	.913
Benefit analysis	Identification	No comparison possible (0 difference)					
	Description	3	**.000**	**.021**	-	-	.439
Weakness analysis	Identification	4	-	**.000**	-	-	.317
	Description	4	-	**.000**	-	-	.317
Trade-off Analysis	Identification	1	-	**.000**	-	-	.317
	Description	1	-	**.000**	-	-	.317
Risk Analysis	Identification	2	-	**.006**	-	-	.314
	Description	2	-	**.006**	-	-	.314
Evaluation		No comparison possible (0 difference)					
Reflection							
Background (Modelling – Guiding)		49	**.003**	.512	-	-	.072
Title (Other – inf. Arch.)		13	.071 (Inf.)	.709 (other)	**.010**	-	.229
Age and Frequencies Correlation		Negative	*r* = -.260			*p* = .469	
Work Years and Frequencies Correlation		Negative	*r* = -.286			*p* = .423	

Table 30. Statistical significance moment 1

Unfortunately, none of the results have been statistically significant when the control and test group are compared. Therefore, the following hypotheses can be answered.

8.1.6.1 Design Quality

H0: There is no difference in design quality (measured in assigned points and wins) between designs produced by Sogeti architects from the test group and the control group when designing an architecture (moment 1).

H1: There is a difference in design quality (measured in assigned points and wins) between designs of the test group and control group when designing an architecture (moment 1).

The hypothesis above is concerned with whether or not the use of the Rationale Capture Cycle has been beneficial in the evaluation of the relative design quality of an architecture design and its documentation. The measure used were the amount of points and wins the solutions were given by the architects. Both differences are not statistically significant, therefore the following statement can be defined:

This study found that solutions made by the test group participants were awarded more points (590) and wins (19) compared to the solutions made by the control group participants (410 and 11). The result is statistically insignificant, $t(8) = -2.241$, $p = .055$ (points) and $t(8) = -2.101$, $p = .069$ (wins). Therefore, the null hypothesis is retained.

8.1.6.2 Time

H0: There is no difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

The hypothesis above is concerned with whether or not the use of the Rationale Capture Cycle has been intrusive in the regular progression of architecture design activities. The measure used was the aspect of time, measured in the amount of minutes the participants of the research groups required to complete the case. The difference in time is not statistically significant, therefore the following statement can be defined:

This study found that participants of the test group required a statistically insignificant longer time (130 minutes) for completing the architecture design case compared to participants of the control group (107.2 minutes), $t(4.4) = -1.811$, $p = .138$. Therefore, the null hypothesis is retained.

8.1.6.3 Rationale Documentation

The following hypotheses are concerned with whether or not the Rationale Capture Cycle has been influential when rationale capture is concerned. The measure used was the amount of rationale elements that have been written down by the architect in the rationale documentation and was coded by the researcher. In the hypotheses the distinction between *rationale identification* and *rationale description* is made. The double numbers represent them both, respectively.

Total Rationale Frequency
H0: There is no difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.
H1: There is a difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture.

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (122) when documenting an architecture in terms of overall rationale elements as compared to participants of the control group (69), $z = -1.798$, $p = .072$. Therefore, the null hypothesis is retained.

Design Decisions
H0: There is no difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (20) when documenting an architecture in terms of design decisions as compared to participants of the control group (9), $t(8) = -1.773$, $p = .114$. Therefore, the null hypothesis is retained.

Problem analysis
H0: There is no difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (16 + 7) when documenting an architecture in terms of problem analysis as compared to participants of the control group (6 + 7), $t(5.6) = -1.789$, $p = .127$ (identification) and $z = .337$, $p = .736$ (definition). Therefore, the null hypothesis is retained.

Constraint analysis
H0: There is no difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 1) when documenting an architecture in terms of constraint analysis as compared to participants of the control group (0 + 0), $z = -1.000$, $p = .317$ (identification) and $z = -1.000$, $p = .317$ (definition). Therefore, the null hypothesis is retained.

Assumption analysis
H0: There is no difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 0) when documenting an architecture in terms of assumption analysis as compared to participants of the control group (0 + 0), $z = -1.000$, $p = .317$ (identification, definition did not occur) Therefore, the null hypothesis is retained.

Option analysis
H0: There is no difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 0) when documenting an architecture in terms of option analysis as compared to participants of the control group (0 + 0), $z = -.983$, $p = .326$ (identification) and $z = -.109$, $p = .913$ (definition). Therefore, the null hypothesis is retained.

Benefit analysis
H0: There is no difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (18 + 1) when documenting an architecture in terms of option analysis as compared to participants of the control group (18 + 4), and $z = -.775$, $p = .439$ (definition, identification offered no difference). Therefore, the null hypothesis is retained.

Weakness analysis
H0: There is no difference in weakness analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in weakness analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (4 + 4) when documenting an architecture in terms of weakness analysis as compared to participants of the control group (0 + 0), $z = -.1.000$, $p = .317$ (identification) and $z = -.1.000$, $p = .317$ (definition). Therefore, the null hypothesis is retained.

Trade-off analysis
H0: There is no difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (1 + 1) when documenting an architecture in terms of trade-off analysis as compared to participants of the control group (0 + 0), $z = -1.000$, $p = .317$ (identification) and $z = -1.000$, $p = .317$ (definition). Therefore, the null hypothesis is retained.

Risk analysis
H0: There is no difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (2 + 2) when documenting an architecture in terms of risk analysis as compared to participants of the control group (0 + 0), $z = -1.500$, $p = .134$ (identification) and $z = -1.500$, $p = .134$ (definition). Therefore, the null hypothesis is retained.

Evaluation and reflection
H0: There is no difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
H1: There is a difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when designing an architecture (moment 1).
As there is no data to compare, considering both research groups did not offer any evaluations and reflections explicitly in the documentation, the null hypothesis is retained.

8.2 Moment 2: Request for Change (RfC)

8.2.1 Participants

10 architects participated in moment 1 of the experiment session. Their demographic data can be seen in the following table. The exact employers are explicated in a separate table in order to ensure the anonymity of the participants. The respective ages are also removed from this table for the same reason.

Architect	Title	Experience in IT (yrs)
C1	Sr. information architect	36
C2	Sr. information architect	20
C3	Enterprise architect	25
C4	Enterprise architect	25
T1	Information architect	13
T2	Sr. information architect	34
T3	Enterprise architect	22
T4	Business analyst	29

Table 31. Participant's demographic data of moment 2

Current employers
Financial services institution
Major government agency
Major financial institution
International IP provider
Major government agency
Major government agency
International telecommunications business
International technology solutions company

Table 32. Employers of moment 2

8.2.2 Time

During the experiment session the total completion time per participant was recorded. The results can be found below.

Participant	Duration	Time (min)	Participant	Duration	Time (min)
C1	13:35 – 14:59	84	T1	12:11 – 13:40	89
C2	19:00 – 20:10	70	T2	13:30 – 15:20	110
C3	18:55 – 21:10	135	T3	18:50 – 19:50	60
C4	07:30 – 10:00	150	T4	18:53 – 20:30	97
Average		110	Average		89
Total		**439**	**Total**		**356**

Table 33. Experiment session durations of moment 2

In total, the test group spent 83 minutes less. On average, the test group participants spent 22 less minutes completing the case. C3 and C4 spent notably longer than all other participants. They are not related, as they completed the case on different times and dates. Furthermore, the variance between the participants of both groups are rather large. The control group results ranges from 70 to 150 minutes whilst the test group results ranges from 60 to 110 minutes. The observed difference is largely due to the contributions of C3 and C4 as they both spent significantly longer than the group average.

8.2.3 Design Quality

8.2.3.1 Ranking

A ranking similar to moment 1 is performed. The same rules and limitations apply. The results can be found below.

Participant	Assigned solutions	Rank - Solution	Rank - Points	Participant	Assigned solutions	Rank - Solution	Rank - Points
C1	C3, C4, T2	1. T2 2. C3 3. C4	1. 46 2. 39 3. 15	**T1**	T2, T3, C1	1. T2 2. C1 3. T3	1. 40 2. 35 3. 25
C2	T1, T4, C3	1. T1 2. T4 3. C3	1. 60 2. 25 3. 15	**T2**	T3, C2, C3	1. T3 2. C2 3. C3	1. 65 2. 20 3. 15
C3	C1, C4, T4	1. C1 2. C4 3. T4	1. 45 2. 35 3. 20	**T3**	T4, C2, C4	1. C4 2. C2 3. T4	1. 50 2. 35 3. 15
C4	T1, T3, C1	1. C1 2. T1 3. T3	1. 60 2. 30 3. 10	**T4**	T1, T2, C2	1. T1 2. T2 3. C2	1. 60 2. 25 3. 15

Table 34. Results ranking #2

During this ranking, the Sogeti supervisor of the research project had to step in to replace T3 due to availability restrictions of the participant. The main supervisor was chosen due to her close relation to T3, both in practice and title. Also, both individuals have intimate knowledge and experience with the topic and case. Therefore, this was the most accurate replacement.

The following table represents the results per solution instead of per participant. A win is counted if the solution is ranked higher than another one in the same set. Two wins are counted if the solution is ranked first in the set as it will have beaten the second and third solution in that set. The wins follow the transitivity rules described in paragraph 7.2.2, Measures.

Solution of	Wins	Points	Solution of	Wins	Points
C1	5	140	T1	5	150
C2	2	70	T2	5	111
C3	1	69	T3	2	100
C4	3	100	T4	1	60
Total control group solutions points:	**11**	**379**	**Total test group solutions points:**	**13**	**421**

Table 35. Results per solution ranking moment 1

The first observable difference is that the solutions made with the inclusion of design rationale (test group) are awarded more points (421) than the control group solutions (379). The test group solutions also won more (13) than the control group solutions did (11). However, these differences are quite small. For further insight the solutions are ranked according to their evaluations. The following table represents the results in order of points, first, and in order of amount of wins second.

Rank #	Solution of	Points	Wins
1	T1	150	5
2	C1	140	5
3	T2	111	5
4	C4	100	3
5	T3	100	2
6	C2	70	2
7	C3	69	1
8	T4	60	1

Table 36. Solutions ranked moment 1

First and foremost, the top solution was the solution of T1 with 150 points. C1 is a close second with 140 points. There is a small tendency for test group solutions to rank higher, but the difference is small. No distinct significant pattern can be distinguished from this data. In order to gain more insight into the results and what they mean, the Bradley Terry probabilities model is fitted.

8.2.3.2 Bradley Terry Probability Model

As described in paragraph 7.2.2, Measures, a generalized Bradley Terry (BT) is fitted, similar to moment 1. All data is transformed into a ''*T1 vs C1 = Win*'' format. In the table below, some descriptive statistics are explicated. Here, the total matches, wins, losses and percentages are calculated by the BT model.

Solution	Total matches	Wins	Losses	% wins	% losses	% ties
C1	6.00	5.00	1.00	83.33	16.67	0.00
C2	6.00	2.00	4.00	33.33	66.67	0.00
C3	6.00	1.00	5.00	16.67	83.33	0.00
C4	6.00	3.00	3.00	50.00	50.00	0.00
T1	6.00	5.00	1.00	83.33	16.67	0.00
T2	6.00	5.00	1.00	83.33	16.67	0.00
T3	6.00	2.00	4.00	33.33	66.67	0.00
T4	6.00	1.00	5.00	16.67	83.33	0.00

Table 37. Summary statistics BT model moment 2

The following table provides the estimated parameters, i.e. the extent in which the factors influence the probabilities of winning. For example, solution C1 has a very strong influence (2.693) on winning a certain match. The total of estimates are 8.

Solution	Estimate	Standard deviation	Lower bound	Upper bound
C1	2.693	0.926	0.878	4.507
C2	0.004	0.004	-0.003	0.011
C3	0.001	0.002	-0.002	0.005
C4	0.006	0.004	-0.002	0.014
T1	2.612	0.908	0.833	4.392
T2	2.619	0.917	0.822	4.415
T3	0.064	0.052	-0.039	0.166
T4	0.001	0.001	-0.001	0.004

Table 38. Estimated parameters BT model moment 2

The Bradley Terry model considers there are three major influencing solutions, C1, T1 and T2. All other solutions are deemed to be of less influence. This is likely due to these three solutions won 83.33% of their matchups and due to the low amount of rankings to begin with. Therefore, the BT model is likely to be less nuanced in its results.

Using the estimates above, the BT model calculates the winning probabilities per matchup based on the existing wins and losses of the dataset. The rows represent the likelihood of winning and the columns represent the likelihood of losing.

Solutions		Likelihood of losing								
		C1	C2	C3	C4	T1	T2	T3	T4	Average
Likelihood of winning	C1	0.000	0.998	0.999	0.998	0.508	0.507	0.977	1.000	0.748
	C2	0.002	0.000	0.751	0.426	0.002	0.002	0.062	0.783	0.253
	C3	0.001	0.249	0.000	0.197	0.001	0.001	0.022	0.545	0.127
	C4	0.002	0.574	0.803	0.000	0.002	0.002	0.082	0.830	0.287
	T1	0.492	0.998	0.999	0.998	0.000	0.499	0.976	1.000	0.745
	T2	0.493	0.998	0.999	0.998	0.501	0.000	0.976	1.000	0.746
	T3	0.023	0.938	0.978	0.918	0.024	0.024	0.000	0.982	0.486
	T4	0.000	0.217	0.455	0.170	0.000	0.000	0.018	0.000	0.108

Table 39. Winning probabilities BT model moment 2

Using the estimated likelihood parameters, a full ranking can be derived from the data based on the likelihood of winning. The result can be seen in the following table.

Rank #	Solution of	Estimated parameter (out of 10)	Average winning probability
1	C1	2.693	0.748
2	T2	2.619	0.746
3	T1	2.612	0.745
4	T3	0.064	0.486
5	C4	0.006	0.287
6	C2	0.004	0.253
7	C3	0.001	0.127
8	T4	0.001	0.108

Table 40. Solutions ranked BT model moment 2

The first notable result is that there are three clear winners among the solutions, according to the BT model. C1, T1 and T2 are the clear solutions that prove to be of major influence when predicting the likelihood of a win. These solutions also have the best average winning probabilities versus all other solutions. These differences might not seem very significant and largely dependent on individual solutions. However, when the results are generalized and compared they may seem clearer.

Research Group	Sum of likelihood (out of 8) (estimated parameter)	Average winning probability
Control group	2.704	0.354
Test group	5.296	0.521

Table 41. BT model compared moment 2

All solutions being equal, the sum of likelihood for each solution would be 1. There are 8 solutions, so the sum of likelihood for all participants would be 8. In other words, if all solutions had the same scores and same amount of influence in determining the likelihood of winning a certain match all estimated parameters would be 1. Here, taking the data of the BT model into account, the sum of likelihood is totalled for each research group. The test group participants account for a much larger portion (5.296 / 8) than the control group (2.704 / 8). This shows that, based on the rankings, the test group has a higher degree of influence over winning matchups. This means that those solutions perform better during the ranking and are evaluated higher by the architects.

8.2.4 Rationale Documentation

Similar to moment, in addition to ranking the solutions by the architects themselves, the architects were asked to provide additional documentation alongside their design. The results of the coding analysis of the first experiment session can be found in the following tables. The table with the participants labelled as a 'T' are those of the test group (participants who had access to the rationale documentation of the original solution in the case). The participants labelled as a 'C' belong to the control group, these participants only had access to the models and a brief description of the architecture. The numbers in the cells represent the frequency of occurrence.

Rationale Type		C1	C2	C3	C4	*AVG*	**Total**	T1	T2	T3	T4	*AVG*	**Total**
						per DR element						per DR element	
Design decision		4	4	3	4	*3.8*	**15**	2	6	2	2	*3*	**12**
Problem analysis	Ident.	4	2		5	*3.7*	**11**	10	4	5	2	*5.3*	**21**
	Descr.	2			9	*5.5*	**11**	4	8	2	2	*4*	**16**
Constraint analysis	Ident.			5		*5*	**5**			2	1	*1.5*	**3**
	Descr.			5		*5*	**5**			1		*1*	**1**
Assumption analysis	Ident.		5			*5*	**5**			3	1	*2*	**4**
	Descr.									1	3	*2*	**4**
Option analysis	Ident.	10	5	3	9	*6.8*	**27**	7	5	10	3	*6.3*	**25**
	Descr.	10	4	6	9	*7.3*	**29**	6	7	11	6	*7.5*	**30**
Benefit analysis	Ident.	1	1	10	6	*4.5*	**18**	7	6	3	5	*5.3*	**21**
	Descr.			3	2	*2.5*	**5**	6	5	1	1	*3.3*	**13**
Weakness analysis	Ident.	1	1	5	6	*3.3*	13	1	1	2	1	*1.3*	**5**
	Descr.			4	2	3	6		3		2	*2.5*	**5**
Trade-off Analysis	Ident.				1	1	1			2	1	*1.5*	**3**
	Descr.									2		*2*	**2**
Risk Analysis	Ident.	1	2	2		1.7	5	1	2	3		*2*	**6**
	Descr.			2		2	2	1		6		*3.5*	**7**
Evaluation													
Reflection													
Average		*4.13*	*3*	*4.36*	*5.3*	**CTRL total 158**		*4.5*	*4.7*	*3.5*	*2.3*	**TEST total 178**	
Total		**33**	**24**	**48**	**53**			**45**	**47**	**56**	**30**		

Table 42. Documentation results moment 2

The table shows there is a small difference in the amount of coded rationale types between the research groups. The difference is 20 rationale types, which is roughly an increase of 13%.

Even though the overall increase isn't as large, the spread of rationale types was very notable during the first session. This time around, however, the research groups have a similar spread of identified rationale types. The control group omitted 4 out of 19 rationale types whilst the test group omitted 2. This is to be expected, as both research groups had access to the Rationale Capture Cycle when documenting their architecture and likely used it. Interestingly, similar to the results of moment 1, evaluation and reflection have no mention in the documentation of architects. This does not necessarily mean the Rationale Capture Cycle did not stimulate those elements, only that these were not explicated as such in the documentation. The difference between identified and described rationale types, however, are different to moment 1. See the following table.

	C	**T**	**T-C**
Identification of rationale types	85	88	3
Description of rationale types	58	78	20
Elaboration percentage	68.2%	88.6%	21.4%

Table 43. Identification and description difference moment 2

In moment 1, the elaboration percentage for both research groups were very similar (50% and 43.7% respectively) with only a difference of 6.7% in favour of the control group participants. In moment 2, however, the difference is much larger (21.4%). Even though both research groups identified a very similar amount of rationale types, the test group elaborated on them in a larger amount of cases.

Interestingly, as the observed overall difference is only 20 rationale types, the difference can almost completely be attributed to the control group not elaborating on their rationale identifications. The differences per rationale type and their contribution to the differences can be seen in the following table.

Rationale Type		**Code**	**C (#)**	**T (#)**	**# increase**	**% increase (per rationale type)**	**% increase (of overall increase in frequency)**
Design decision		DD	15	12	-3	-20%	-15%
Problem analysis	Identification	PI	11	21	10	90.9%	50%
	Description	PD	11	16	5	45.5%	25%
Constraint analysis	Identification	CI	5	3	-2	-40%	-10%
	Description	CD	5	1	-4	-80%	-20%
Assumption analysis	Identification	AI	5	4	-1	-20%	-5%
	Description	AD		4	4	-	20%
Option analysis	Identification	OI	27	25	-2	-7.4%	-10%
	Description	OD	29	30	1	3.5%	5%
Benefit analysis	Identification	BI	18	21	3	16.7%	15%
	Description	BD	5	13	8	160%	40%
Weakness analysis	Identification	WI	13	5	-8	-61.5%	-40%
	Description	WD	6	5	-1	-16.7%	-5%
Trade-off Analysis	Identification	TI	1	3	2	200%	10%
	Description	TD		2	2	-	10%
Risk Analysis	Identification	RI	5	6	1	20%	5%
	Description	RD	2	7	5	250%	25%
Evaluation		EV				-	0%
Reflection		ER				-	0%

Table 44. Difference per rationale type moment 2

Problem identification sees the largest absolute increase, going from 11 to 21 types, which is an increase of 10. Percentage wise, however, benefit description sees the largest relative increase. The control group managed to describe 5 benefits whilst the test group described 13. This is a relative increase of 160%. However, the differences are relatively widespread and offer no clear pattern. This is partly due to the overall difference between the groups is quite small, therefore the individual differences are understandably erratic.

In order to further determine where this difference comes from, the amount of times the original rationale was captured when documenting the architecture is recorded. For example when an architect chooses for a design option because the original rationale mentioned a certain risk or assumption. These numbers represent the amount of times the architect used the rationale that was present as the independent variable, i.e. the rationale that was present in the case for the original solution. This rationale was supposed to be used by the participant to further improve and add to their own reasoning process. The extent in which this was made explicit is recorded in the following table.

Rationale Type		C1	C2	C3	C4	Total	T1	T2	T3	T4	Total
Design decision		2			3	**5**	1	5	8	2	**16**
Problem analysis	Identification		2	9	5	**16**	9	1	3		**13**
	Description		1	3	3	**7**	2				**2**
Constraint analysis	Identification										
	Description										
Assumption analysis	Identification		2			**2**					
	Description										
Option analysis	Identification	1			1	**2**	1	7	8	2	**18**
	Description	3			1	**4**		6	4	1	**11**
Benefit analysis	Identification							2	1	2	**5**
	Description										
Weakness analysis	Identification							3		1	**4**
	Description										
Trade-off Analysis	Identification								1		**1**
	Description										
Risk Analysis	Identification										
	Description										
Evaluation											
Reflection											
Total		**6**	**5**	**12**	**13**	**36**	**13**	**24**	**25**	**8**	**70**

Table 45. Identified original rationale types moment 2

The test group participants documented nearly double the amount of individual rationale types concerning the original case and its rationale as opposed to the control group. This is to be expected as the rationale was removed (independent variable) for the control group case. The occurring rationale types from the control group are therefore largely just stating the problems or original options, rationale types that can be deduced from reading the description and models themselves. The test group results however, draw far more varied rationales. This is most likely due to the fact the test group participants had access to the original rationale, allowing them to make more well-rounded decisions in terms of included rationale types. Another dimension to take into account is the background of the participant. The results below feature the performance when background and recent work content are taken into account.

Rationale Type frequency	C	T
Years of work experience (average)	26.5	24.5
Rationale type frequency	158	178
Rationale type frequency per work year	5.96	7.27

Table 46. Rationale type frequency per work year moment 2

The control group has, on average, 2 more years of experience in the IT industry. This is, however, a relatively small difference on the total of work experience to explain the increase in frequency. If the control- and test group split is ignored and the architects are sorted on their descending work experience, the following results surface.

Participant	Work experience (yrs)	Rationale type frequency
T1	13	45
C2	20	24
T3	22	56
C3	25	48
C4	25	53
T4	29	30
T2	34	47
C1	36	33

Table 47. Rationale type frequency and work years moment 2

One can expect there to be a positive relationship between work experience and rationale type results. An experienced architect should perform better than inexperienced architects. View the following graph.

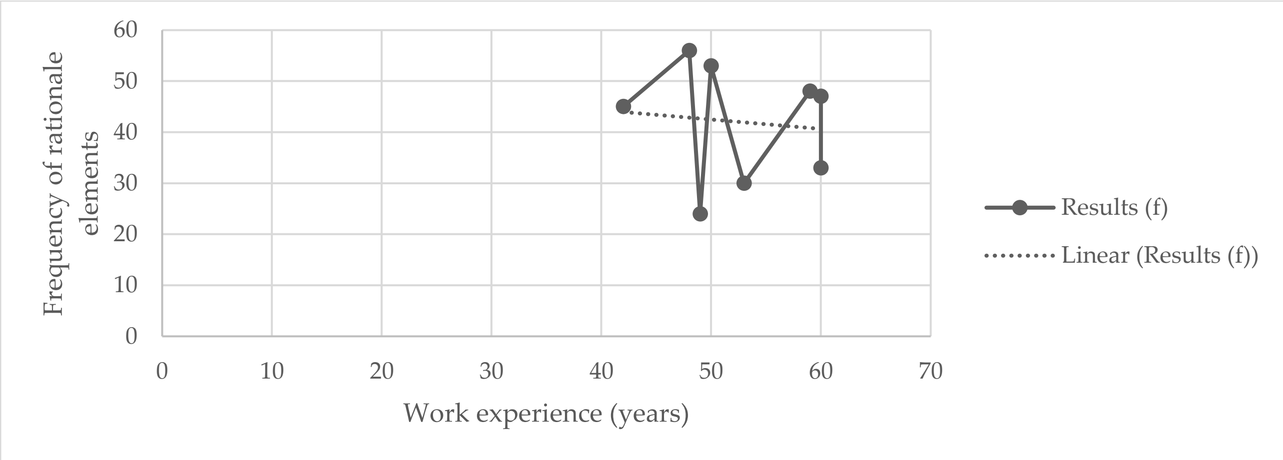


Figure 16. Results by work experience moment 2

As the graph shows, the relationship between work experience (x-axis) and the frequency of rationale types that were identified (y-axis) is actually negative. The linear (dotted line) of the results shows a downward slope. This implies that if you have less experience you will do comparatively better during the experiment. This is the same result as in moment 1. However, there is too little data to statistically confirm this is not due to chance. The results may simply show that the younger participants (with less experience) were more likely to use the original rationale or were more familiar with the architecture software during the experiment and thus had more time to perform better. To obtain better insight into this question it is interesting to look at the average age of the participant and compare it with their performance. Crossing the data with the age of the participants yields the following result.

Rationale Type frequency	C	T
Age (average)	54.5	50.8
Rationale type frequency	158	178
Rationale type frequency per year of age	2.90	3.50

Table 48. Rationale type frequency per work year moment 2

The control group is, on average 3.7 years older than the test group participants. This is, however, a relatively small difference to explain the increase in frequency. If the control- and test group split is ignored and the architects are sorted on their respective descending ages, the following results surface.

Participant	Age (yrs)	Rationale type frequency
T1	42	45
T3	48	56
C2	49	24
C4	50	53
T4	53	30
C3	59	48
T2	60	47
C1	60	33

Table 49. Rationale type frequency and age moment 2

Transforming the results into a graph produces the following figure.

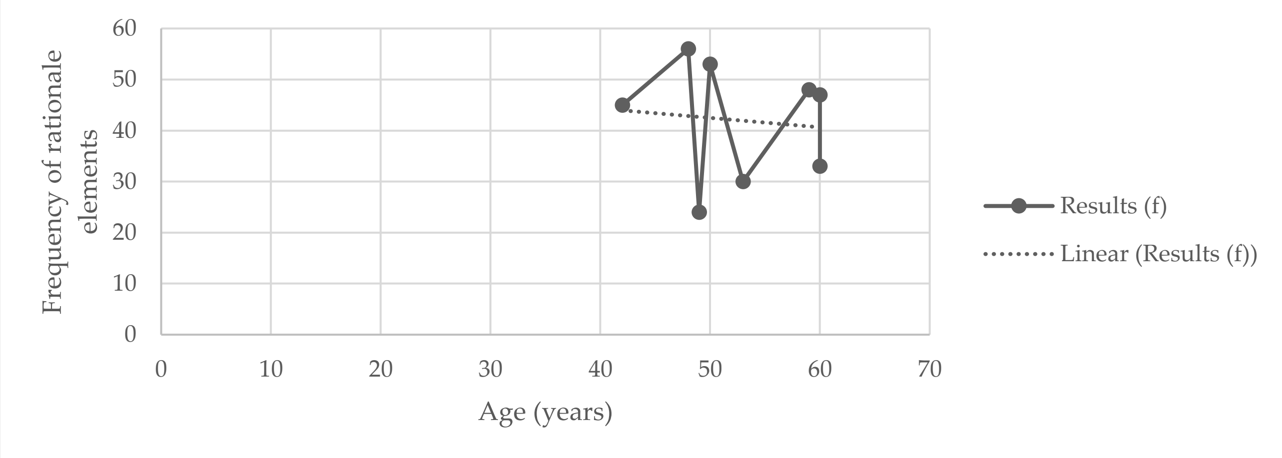


Figure 17. Results by age moment 2

As the graph shows, the relationship between the age of the architects (x-axis) and the frequency of rationale types that were identified (y-axis) is actually negative. The linear (dotted line) of the results shows a downward slope. This implies that if you are younger will do comparatively better during the experiment. However, there is too little data to statistically confirm this is not due to chance, especially considering the ages of the architects are too similar. The results may simply show that the younger participants were more likely to adopt the existing rationale or were more familiar with the architecture software during the experiment. The following table shows the difference when the results are compared between the official titles the architects hold.

Role	(Senior) Information architects				Other (enterprise architect, ICT architect, business analyst)			
	C1	C2	T1	T2	C3	C4	T3	T4
Rationale type frequency	33	24	45	47	48	53	56	30
Total	**149**				**187**			

Table 50. Rationale type frequency difference per title moment 2

The information architects perform comparatively worse than the other participants. The non-information architects perform slightly better. 3 out of 4 architects of the ‘other’ category hold the title of information architect as a secondary role, however. This makes the comparison even trickier and less meaningful.

The following table splits the architects into two categories again. This time, the distinction is made between architects who hold a more guiding role versus architects who perform operational modelling work. This distinction is made on the basis of whether or not the architect has had demonstrable experience in actually modelling and operational work instead of solely guiding architects and requirements conversations. In principle, all architects are of senior experience and thus have a more guiding role. Similarly, all architects model and do operational work as well. The distinction is made between whether or not the architect has had demonstrable operational work as a major responsibility in recent work experience as per the CV work description.

Work experience	Guidance role				Modelling / operational role			
	C3	C4	T3	T4	C1	C2	T1	T2
Rationale type frequency	48	53	56	30	33	24	45	47
Total	**187**				**149**			

Table 51. Rationale type frequency difference per recent work experience moment 2

Interestingly, the architects who fulfil a more operational role, i.e. architects who have demonstrable modelling experience in the past few years as per CV, perform comparatively worse. The difference of 187 versus 149, which is a decrease of around 20%. However, the CV split is very tricky to make as the description of actual work experience is difficult to estimate and evaluate. This result also conflicts with the result of moment 1, where the modelling architects performed better. This makes the comparison and result even more unreliable.

8.2.5 Survey

At the end of the survey, participants of both groups were asked to provide feedback and additional thoughts regarding the inclusion / exclusion of original rationale, the case and the research overall. Also, the survey provided another outlet for feedback regarding the Rationale Capture Cycle. The major questions regarding the Rationale Capture Cycle concerned, similar to the survey of moment 1, added value, effectiveness, readability, usability, intuitiveness & ease of use.

8.2.5.1 Interpretation and Implementation

Interpretation of the as-is architecture

4 out of 4 test group participants found the as-is architecture to be clear and easy to interpret. However, T3 mentioned having trouble understanding one single element.

4 out of 4 control group participants found the as-is architecture to be clear and easy to interpret. C2 also mentioned assumptions helped him interpret the architecture.

Missing information

4 out of 4 test group participants found there to be missing information in order to correctly interpret and analyse the architecture and its meaning. T2 mentioned the architecture should have defined limits and specific constraints. Another mentioned there should have been a stakeholder map. T3 mentioned there should have been a road map and T4 would have liked additional context.

4 out of 4 control group participants found there to be missing information in order to correctly interpret and analyse the architecture and its meaning. C1 would have liked additional context and C2 would have liked more non-functional requirements. C3 and C4 would have liked more business requirements.

Used information

2 out of 4 test group participants used *every* piece of information in the original design rationale available. T3 primarily used pros and cons, constraints and assumptions and T4 primarily used problem structuring and assumptions to help him reason.

Irrelevant information

2 out of 4 test group participants did not find any irrelevant information. All original design rationale was found useful. T3 mentioned not using trade-off analysis during this assignment due to not having to consider alternatives and T4 did not find the constraints useful.

Effective structure of original design rationale

2 out of 4 test group participants found that the original rationale was very structured and therefore useful. T2 also mentioned the structure of the Rationale Capture Cycle forces you to handle this objectively and removes impulsive human mannerisms. By writing this explicitly you are supported in evaluating others and removing your own impulses. Another mentioned clearly defining the various options was extremely useful. T3 suggested that the rationale should be more gravitated around constraints. T4 uttered the structure to be improved but did not mention in what way.

1 out of 4 control group participants found that the original rationale was very structured and therefore useful.

8.2.5.2 Rationale Capture Cycle

Effectiveness and added value

4 out of 4 test group participants found the model to be very useful and effective. T2 uttered that the model brings explicit attention to constraints and trade-offs, for instance, that you otherwise could neglect to explicate. T3 mentioned that it forces you to spend equal attention to all aspects of decision making. T1 mentions the Rationale Capture Cycle is somewhat confusing and slowed him down when documenting an architecture. He mentioned that these elements are already present in his reasoning process, and therefore the model is of little use to him. T4 also did not use the model too much but it did help him reason.

3 out of 4 control participants found the model to be very useful and effective. C2 mentioned it helped him greatly remember all aspects of reasoning, yet C1 mentioned using the reasoning model less due to the nature of the case. C1 uttered he would only use the model for complex and large decisions. C4 mentioned he did use it but also said the model is not applicable in all situations.

Readability, ease and intuitiveness
4 out of 4 test group participants found the model very easy to read and understand. T3 also mentioned that it forces a more analytical approach to architecture design and is worried that that comes at the cost of creativity. T4 uttered that he did not agree with the model sometimes, but that did force him to reason.

3 out of 4 control group participants found the model very easy to read and understand. No additional comments were made.

Structure and sequence of the model
2 out of 4 test group participants found the entire sequence structure of the model to be entirely correct. Both T2 and T3 mentioned that the assumptions should be handled before listing design options in order to list the assumptions of the problems more clearly. They also mentioned constraints and risks should be explicated before design options as they are of great import.

2 out of 4 control group participants found the entire sequence structure of the model to be entirely correct. No additional comments were made.

8.2.5.3 Miscellaneous feedback

T2 mentioned the research was very interesting and a common missing element in practice. Rationale is often missing, especially when projects are spanning longer periods. T3 said the structure in the reasoning model is very important and that the first step, defining the problem or requirement, is crucial in architecture design. T3 also said he used this for his assignment for the Chamber of Commerce and it helped him formalize an architecture driven approach. C4 suggested modelling the Rationale Capture Cycle in the ArchiMate notation. C2 mentioned the case was too short to completely and effectively use the Rationale Capture Cycle or interpret the rationale. This was repeated by C1.

8.2.6 Significance Testing and Hypotheses

In this paragraph, all the comparisons and identified differences are tested for statistical significance as done in moment 1. Per comparison, the distinction is made between performing a student's *t* test or Mann Whitney's *U* test. The Shapiro-Wilk test will be performed to test the data for normality. Levene's test will be performed to test for homogeneity of variances, if the resulting *p* value is lower than 0.05 the data is assumed significant and the variances are significantly different. If these assumptions are not met, Mann Whitney's *U* will be performed, which is the nonparametric variant for comparing means of different groups. All comparisons adhere to a significance level of 5%, i.e. the result is significant when $p < 0.05$. In the case of the Shapiro-Wilk test, the data is considered normal if $p > 0.05$. Only the rationale types that occurred in both research groups can be compared and tested for significance.

Comparison		Result (f) (Test group – control group)	Shapiro-Wilk test		Levene's test	Student's *t*	Mann Whitney's *U*
			C	T			
Ranking points		42	.272	.939	1.000	.688	-
Ranking wins		2	.850	.161	.356	.722	-
Time		-83 minutes	.406	.695	.056	.383	-
Total difference in rationale types		20	.633	.712	.384	.582	-
Design decision		-3	**.001**	**.001**	-	-	.225
Problem analysis	Identification	10	.798	.556	.593	.264	-
	Description	1	.066	.161	.454	.643	-
Constraint analysis	Identification	-2	**.001**	.272	-	-	.741
	Description	-4	**.001**	**.001**	-	-	.850
Assumption analysis	Identification	-1	**.001**	.161	-	-	.741

Assumption analysis	Description	4	-	.161	-	-	.131
Option analysis	Identification	-2	.513	.952	.585	.830	-
	Description	1	.650	.051	.544	.895	-
Benefit analysis	Identification	3	.274	.850	.053	.760	-
	Description	8	.224	.123	**.013**	-	.243
Weakness analysis	Identification	-8	.123	**.001**	-	-	.321
	Description	-1	.272	.224	.604	.844	-
Trade-off Analysis	Identification	2	**.001**	.272	-	-	.405
	Description	2	-	**.001**	-	-	.317
Risk Analysis	Identification	1	.272	.972	.506	.766	-
	Description	5	**.001**	**.034**	-	-	.508
Evaluation		No comparison possible					
Reflection							
Background (Modelling – Guiding)		29 (m)	.259	.476	.924	.276	-
Title (Inf. Arch – other)		29 (o)	.259	.476	.924	.276	-
Age and Frequencies Correlation		Negative	r = -.105			p = .805	
Work Years and Frequencies Correlation		Negative	r = -.148			p = .726	

Table 52. Statistical significance moment 2

Unfortunately, none of the results have been statistically significant when the control and test group are compared. Therefore, the following hypotheses can be answered.

8.2.6.1 Design Quality

H0: There is no difference in design quality (measured in assigned points and wins) between designs produced by Sogeti architects from the test group and the control group when implementing a change (moment 2).
H1: There is a difference in design quality (measured in assigned points and wins) between designs of the test group and control group when implementing a change (moment 2).

The hypothesis above is concerned with whether or not the inclusion of design rationale has been beneficial in the evaluation of the relative design quality of an architecture design and its documentation. The measure used were the amount of points and wins the solutions were given by the architects. Both differences are not statistically significant, therefore the following statement can be defined:

This study found that solutions made by the test group participants were awarded more points (421) and wins (13) compared to the solutions made by the control group participants (379 and 11). The result is statistically insignificant, $t(6) = -.421$, $p = .688$ (points) and $t(6) = -.374$, $p = .722$ (wins). Therefore, the null hypothesis is retained.

8.2.6.2 Time

H0: There is no difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in artefact intrusiveness (measured in time spent completing the case) between Sogeti architects of the test group and control group when implementing a change (moment 2).

The hypotheses above is concerned with whether or not the inclusion of design rationale has been intrusive in the regular progression of architecture design activities. The measure used was the aspect of time, measured in the amount of minutes the participants of the research groups required to complete the case. The difference in time is not statistically significant, therefore the following statement can be defined:

This study found that participants of the test group spent statistically insignificant less time (356 minutes) for completing the architecture design case compared to participants of the control group (439 minutes), $t(6) = .940$, $p = .383$. Therefore, the null hypothesis is retained.

8.2.6.3 Rationale Documentation

The following hypotheses are concerned with whether or not the Rationale Capture Cycle has been influential when rationale capture is concerned. The measure used was the amount of rationale elements that have been written down by the architect in the rationale documentation and was coded by the researcher. In the hypotheses the distinction between *rationale identification* and *rationale description* is made. The double numbers represent them both, respectively.

Total Rationale Frequency
H0: There is no difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in total rationale frequency (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (178) when documenting an architecture in terms of overall rationale elements as compared to participants of the control group (158), $t(6) = -.582$, $p = .582$. Therefore, the null hypothesis is retained.

Design Decisions
H0: There is no difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in design decisions (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (20) when documenting an architecture in terms of design decisions as compared to participants of the control group (9), $z(4) = -1.214$, $p = .225$. Therefore, the null hypothesis is retained.

Problem analysis
H0: There is no difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in problem analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (16 + 7) when documenting an architecture in terms of problem analysis as compared to participants of the control group (6 + 7), $t(6) = -1.231$, $p = .264$ (identification) and $t(6) = -.488$, $p = .643$ (description). Therefore, the null hypothesis is retained.

Constraint analysis
H0: There is no difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in constraint analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 1) when documenting an architecture in terms of constraint analysis as compared to participants of the control group (0 + 0), $z = -.331$, $p = .741$ (identification) and $z = -.189$, $p = .850$ (description). Therefore, the null hypothesis is retained.

Assumption analysis
H0: There is no difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in assumption analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 0) when documenting an architecture in terms of assumption analysis as compared to participants of the control group (0 + 0), z = -.331, p = .741 and z = -.1.512, p = .131 (description). Therefore, the null hypothesis is retained.

Option analysis
H0: There is no difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in option analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (5 + 0) when documenting an architecture in terms of option analysis as compared to participants of the control group (0 + 0), $t(6)$ = .225, p = .830 (identification) and $t(6)$ = -.137, p = .895 (description). Therefore, the null hypothesis is retained.

Benefit analysis
H0: There is no difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in benefit analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (18 + 1) when documenting an architecture in terms of option analysis as compared to participants of the control group (18 + 4), and $t(6)$ = -.320, p = .760 and z = -1.169, p = .243 (description). Therefore, the null hypothesis is retained.

Weakness analysis
H0: There is no difference in weakness analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in weakness analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (4 + 4) when documenting an architecture in terms of weakness analysis as compared to participants of the control group (0 + 0), z = -.992, p = .321 (identification) and $t(6)$ = .206, p = .844 (description). Therefore, the null hypothesis is retained.

Trade-off analysis
H0: There is no difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in trade-off analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (1 + 1) when documenting an architecture in terms of trade-off analysis as compared to participants of the control group (0 + 0), z = -.833, p = .405 (identification) and z = -1.000, p = .317 (description). Therefore, the null hypothesis is retained.

Risk analysis
H0: There is no difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).
H1: There is a difference in risk analysis (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

This study found that participants of the test group demonstrated a statistically insignificant higher frequency of rationale types (1 + 1) when documenting an architecture in terms of trade-off analysis as compared to participants of the control group (0 + 0), $t(6)$ = -.311, p = .766 (identification) and z = -.661, p = .508 (description). Therefore, the null hypothesis is retained.

Evaluation and reflection

H0: There is no difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

H1: There is a difference in evaluation and reflection (measured in frequency of occurring rationale elements) between Sogeti architects of the test group and control group when implementing a change (moment 2).

As there is no data to compare, considering both research groups did not offer any evaluations and reflections explicitly in the documentation, the null hypothesis is retained.

''Even though technically all alternative hypotheses are rejected, the results are not invalid. The lack of statistical significance is likely due to the low sample and not the research's main aim. Additional analysis into the results could provide valuable insights.''

9 ANALYSIS

This chapter builds on the results of chapter 8 and continues the search for patterns, results, differences, structures and relationships.

9.1 Moment 1: Architecture Design

As the results have shown, none of the standard comparisons between the test and control group have been statistically significant in terms of either a difference in identified rationale frequencies or time spent completing the case. Therefore, the null hypotheses have to be retained to maintain scientific practice. However, as the sample size is very low and the expected variance between architects very high, the lack of statistical significance was to be expected. Significance was also not an absolute requirement and goal of this research as it does not fit the objectives and data of the study. The goal was to demonstrate whether or not the influence of the model can somehow be demonstrated. This chapter analyses the results further in order to achieve that premise.

9.1.1 Influence of the Rationale Capture Cycle

9.1.1.1 Time

Even though the differences in time are not statistically significant, there is an identified difference in time between architects of the test- and control group. The difference of 22.8 minutes is an increase of 21.27% when the Rationale Capture Cycle is in use. However, this difference is largely in part due to C3 (90 minutes) and C5 (68 minutes) whom both spent less than the average. The other participants of the control group all clocked in at around the average of both the test and control group. So the lack of the Rationale Capture Cycle does not necessarily demonstrate it is the sole factor. It can still be due to chance. If we view the results of C3 and C5 to understand this very fast result we also do not see an obvious difference. C3 identified 11 rationales, which is the same as others in the same group. C5, who spent the least (68 minutes), even had the best result of 23 identified frequencies. Their recent background and type of work also offers no explanation as C3 and C5 also belonged to different roles, 'Guiding' and 'Modelling' respectively. Unfortunately, the results cannot clarify as to why they were faster than the other participants of the same group. This may be due to chance or other unknown factors.

9.1.1.2 Design Quality

Even though the differences in both points and wins are not statistically significant, the identified difference is still obviously present. The difference of 180 points between the test group solutions (590) and the control group solutions (410) might not seem large, however that difference is deceptive. Each architect is forced to rank all solutions and divide 100 points. In nearly half of the cases where architects spend points, the pattern of 50-30-20 points was used. This is likely due to not wanting to give a solution by their colleagues 0 points, and each solution will have some value. Therefore, the points of the control group are still significant. On top of that, the differences are *nearly* statistically significant. The differences in points ($p = .055$) and wins ($p = .063$), respectively, cannot be ignored. Technically, the alternative hypotheses are rejected in order to maintain scientific practice considering the significance value of $p < .05$ are not achieved. However, these results still show that out of a 1000 repeated experiments an incorrect result can surface 55 and 63 times respectively.

9.1.1.3 Rationale Documentation

Even though the differences in identified rationale frequencies when coding the rationale documentation are not statistically significant, there is an obvious difference in rationale frequencies between architects of the test- and control group. The difference of 53 rationale types is an increase of nearly 77% when the Rationale Capture Cycle is in use. This difference, however, is largely in part due to two notable participants, T2 and T4. T2 and T4 identified 32 and 42 rationale types respectively, which is around 60% of the total test group results. This causes a high variance and insignificant statistical result. A Mann-Whitney U test is performed (as the data violates the normality assumption) to compare the results of T2 and T4 versus the entirety of other participants. From this data, it can be concluded that rationale capture in terms of identified rationale types was higher for participants T2 and T4 than it was for the other participants ($U = 0.00$, p = **.034**. As the results of participants T2 and T4 provided a statistically significant better result than both their test group peers and entirety of both research groups, this result is worth analysing further. In order to identify why these two participants performed so well, and even better than their test group peers, further research into their results is needed.

In order to establish whether or not the results are, in fact, due to the use of the Rationale Capture Cycle the rationale documentation is analysed again for additional patterns. For example, the order and sequence in which rationale types appear and whether or not this reflects the Rationale Capture Cycle. If the test group participants did, in fact, use the Rationale Capture Cycle to help them capture rationale, this would be reflected in the way they provide rationale documentation. The Rationale Capture Cycle has the following sequence:

Problems → Options → Benefits / Weaknesses → Assumptions → Constraints → Risks → Trade-Offs

As was shown in Table 22. Rationale type sequence frequencies in the previous chapter, the test group accounted for 34 of these sequences compared to the 14 of the control group. However, this difference is almost solely due to the sequences of T2 (9 sequences) and T4 (18). Together, they comprise nearly 80% (27/34) of all identified sequences as they appear in the model. The other test group participants all had around the same amount of sequences as the control group participants. In other words, the participants T2 and T4, whom both performed better in terms of overall identified rationale types, also identified those difference more often in a sequence that reflects the same order of rationale types of the Rationale Capture Cycle. Their rationale types occurred in the context you would expect them to occur if they used the Rationale Capture Cycle. This result seems to agree with the better results they showed in terms of identified rationale types and demonstrates that the increase in frequencies has a relationship with an increase in the specific sequences as they appear in the Rationale Capture Cycle. Pearson's *r* is calculated to assess the relationship between the amount of identified rationale frequencies and the amount of identified rationale sequences. There was a very strong positive correlation between the two variables, $r = .918$, $n = 10$, $p =$ **.000**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the amount of identified rationale types and the amount of sequences, as they appear in the Rationale Capture Cycle. This result does not confirm a cause and effect relationship, however it does demonstrate the influence of the Rationale Capture Cycle on the increase of identified rationale types through analysing the amount of sequences.

Secondly, in order for an additional confirmation of the influence of the Rationale Capture Cycle the survey results are looked into. T2 and T4 both offered extensive answers to the survey questions. Both participants answered positive to question 1 whether or not the Rationale Capture Cycle was of added value. Both answered positive when evaluating the Rationale Capture Cycle's effectiveness (question 2). Also, both offered additional remarks in order to improve the model even further. Lastly, the participants remarked the model was easy to read and understand and concluded the model helped to add structure. This result seems to confirm the premise that T2 and T4 had better results due to the influence of the model. This difference is not only statistically significant, but this difference takes shape in the form of the Rationale Capture Cycle. This relationship is also tested for a correlation, which produced a significant positive relationship. To confirm this entire notion, the survey results of both T2 and T4 suggested definite positivity and adoption towards the Rationale Capture Cycle.

9.1.2 Comparisons and Correlations

This paragraph attempts to find correlations and relationships between the previously observed results. 4 questions are asked for further insight into the results. First: is there a relationship between the amount of points and wins during the ranking? Second: is there a relationship between the amount of points during the ranking and the amount of coded rationale types during documentation? Third: is there a relationship between the amount of wins during the ranking and the amount of coded rationale types during documentation? Fourth: how is the Rationale Capture Cycle related to these variables?

	Ranking points	Ranking wins	Rationale frequency	RCC sequence frequency
Ranking points	1	$r = .836$, $p =$ **.003**	$r = .906$, $p =$ **.000**	$r = .814$, $p =$ **.004**
Ranking wins	-	1	$r = .717$, $p =$ **.020**	$r = .489$, $p = .151$
Rationale type frequency	-	-	1	$r = .918$, $p =$ **.000**
RCC sequence frequency	-	-	-	1

Table 53. Correlations moment 1

Pearson's *r* is calculated to assess the relationship between the amount of points the solutions were awarded during the ranking and the amount of wins. There was a very strong positive correlation between the two variables, $r = .836$, $n = 10$, $p =$ **.003**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the points and wins of the solutions during the ranking.

Pearson's *r* is calculated to assess the relationship between the amount of points the solutions were awarded during the ranking and the amount of identified rationale types during documentation. There was a very strong positive correlation between the two variables, $r = .906$, $n = 10$, $p =$ **.000**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the evaluations of the architects and the coding analysis of the researcher.

Pearson's *r* is calculated to assess the relationship between the amount of wins the solutions achieved during the ranking and the amount of identified rationale types during documentation. There was a strong positive correlation between the two variables, r = .717, n = 10, p = **.020**. The result is significant at the .05 level (2-tailed). This confirms that there is a significant positive relationship between the wins of the solutions and the coding analysis of the researcher.

Pearson's *r* is calculated to assess the relationship between the amount of rationale sequences are identified in the solutions during the ranking and the amount of points the solutions were awarded during the ranking. There was a strong positive correlation between the two variables, r = .814, n = 10, p = **.004**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the use of the Rationale Capture Cycle during documentation and the evaluations of the architects.

Pearson's *r* is calculated to assess the relationship between the amount of identified rationale frequencies and the amount of identified rationale sequences. There was a very strong positive correlation between the two variables, r = .918, n = 10, p = **.000**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the amount of identified rationale types and the amount of sequences, as they appear in the Rationale Capture Cycle.

There is one insignificant relationship between the amount of rationale sequences as they appeared in the Rationale Capture Cycle and the wins of each solution. Pearson's *r* is calculated to assess the relationship between the amount of rationale sequences are identified in the solutions during the ranking and the amount of wins the solutions obtained during the ranking. There was a moderate positive relationship between the two variables, r = .489, n = 10, p = .151. The result is insignificant. This implies there is no significant relationship between the wins and the amount of Rationale Capture Cycle sequences. It seems these two elements are too distant to be significant. However, with the strength of the relationship between the Rationale Capture Cycle and the points, this result is not deemed as important.

These answers do not confirm a cause and effect relationship between either elements. However, it does demonstrate a positive relationship exists. First, the results demonstrate there is a relationship between points and wins. This implies that you obtain more wins if you receive more points, which is logical. Second, the results show a positive relationship between the points or wins and the rationale documentation. This relationship is also very strong. This demonstrates that the solutions with a more elaborate and varied design rationale also performed better when evaluated by architects. This implies there is a high level of agreement between the evaluation of the architect and the documentation analysis of the research.

9.1.3 Final Ranking

This paragraph deals with finalizing the results and comparing the various different tests into a final ranking.

Architecture Solution	**Rank in**			**Average number**	**Average rank**
	Architect ranking	Bradley-Terry probability ranking	Rationale Documentation ranking		
C1	6	6	8	6.67	6
C2	8	8	6	7.33	8
C3	10	10	10	10.00	10
C4	9	9	9	9.00	9
C5	4	2	3	3.00	3
T1	5	3	4	4.00	4/5
T2	2	1	2	1.67	1
T3	3	4	5	4.00	4/5
T4	1	5	1	2.33	2
T5	7	7	7	7.00	7

Table 54. Ranking comparison moment 1

Using the three different rankings, an average number is calculated. These numbers are then sorted from low to high, low being the highest rank. The various different tests have a high measure of agreement, with the exception of T4. The BT model estimates that T4 is not the most likely to win in many scenarios. This is interesting as it directly contradicts the two different rankings. In order to determine the agreement between the rankings, a rank correlation can be performed. This tests for ordinal association, i.e. the relationship between rankings of different ordinal variables. In this case, we have 3 ordinal variables: the 3 rankings. The rank correlation coefficient will measure the extent in which the rankings are similar.

Spearman's ρ *is* calculated to assess the relationship between the architecture rankings and the Bradley-Terry ranking. There was a very strong positive correlation between the two variables, ρ = .842, *n* = 10, *p* = **.002**. The result is significant at the .01 level (2-tailed).

Spearman's ρ is calculated to assess the relationship between the architecture rankings and the rankings based on the amount of identified rationale types. There was a very strong positive correlation between the two variables, ρ = .915, *n* = 10, *p* = **.000**. The result is significant at the .01 level (2-tailed).

Spearman's ρ is calculated to assess the relationship between the Bradley-Terry rankings and the rankings based on the amount of identified rationale types. There was a strong positive correlation between the two variables, ρ = .830, *n* = 10, *p* = **.003**. The result is significant at the .01 level (2-tailed).

	Architect ranking	**Bradley-Terry probability ranking**	**Rationale Documentation ranking**
Architect ranking	1	ρ = .842, *p* = **.002**	ρ = .915, *p* = **.000**
Bradley-Terry probability ranking	-	1	ρ = .830, *p* = **.003**
Rationale Documentation ranking	-	-	1

Table 55. Agreement Correlations moment 1

These results show a very high significant correlation coefficient between the three rankings. This implies that the level of agreement between the rankings is very high. A correlation coefficient of 1 would be perfect, i.e. the exact same ranking. These results imply that there are very few differences between the three rankings in terms of the order of their rankings. As we have determined an average rank of three different rankings and we have determined the agreement is very high and significant, a final ranking can be derived. This produces the following final ranking of solutions, based on the multiple different tests.

The final averaged ranking and additional contextual information of the ranking and its solutions can be found below.

		Architecture Ranking		**Bradley Terry probability model**		**Rationale documentation coding analysis**	
Final (average) rank #	**Solution**	Points	Wins	Estimated likelihood parameter (out of 10.0)	Average winning probability	Rationale Frequency	Rationale Capture Cycle sequences
1	**T2**	130	5	2.525	0.697	32	9
2	**T4**	160	4	0.957	0.511	42	18
3	**C5**	110	4	1.671	0.622	23	4
4/5	**T1**	95	4	1.561	0.609	19	3
4/5	**T3**	115	4	1.471	0.598	17	2
6	**C1**	95	3	0.721	0.452	11	4
7	**T5**	90	2	0.368	0.314	12	2
8	**C2**	85	2	0.342	0.300	13	1
9	**C4**	60	1	0.199	0.204	11	3
10	**C3**	60	1	0.186	0.193	11	2

Table 56. Final ranking data moment 1

T2 is declared the best solution, taken the three tests into account, followed by T4. C5 is the best control group candidate and performed relatively well compared to peers of his research group. 2/3 solutions of the top 3 are made in the test group. 4/5 solutions in the top 5 are made in the test group. There is a very distinct pattern that test group solutions are, according to various tests and comparisons, relatively better in terms of design quality and rationale documentation. This ranking is determined by coding analysis of the rationale documentation by the researcher, a statistical probability model and design quality rankings by the architects themselves.

9.2 Moment 2: Request for Change (RfC)

Similar to the previous paragraph, this paragraph will search for any additional patterns, relationship and notable comparisons for the second experiment. The goal was to demonstrate whether or not the inclusion or exclusion of the original design rationale has an influence on architecture design. This chapter analyses the results further in order to achieve that premise.

9.2.1 Presence of Design Rationale

9.2.1.1 Time

Even though the differences in time are not statistically significant, there is an identified difference in time between architects of both research groups. The difference of 83 minutes in favour of the test group is a decrease of nearly 19% when the original rationale is included. You would expect the inclusion and presence of extra documentation to have a negative effect on time needed to complete the case. The results show otherwise.

However, this difference is largely in part due to C3 (135 minutes) and C4 (150 minutes) whom both spent way more than both the control group (110 minutes) and total average (99.5 minutes). The other participants of the control group all clocked in at around the average or less of both research groups. So the exclusion of the original design rationale does not necessarily demonstrate it is the sole factor. C1 (84 minutes) and C2 (70 minutes) achieved far better times, for instance. It can still be due to chance. If we view the results of C3 and C4 to understand this slow time we may see a possible explanation. C4 and C4 both had excellent documentation scores, 48 and 53 identified rationale types respectively, as opposed to the quick times of C1 (33 types) and C2 (24 types), whom had worse documentation scores. At first glance it would seem that when an architect spends more time, the identified rationale types go up. However, this is debunked when comparing it to the test group results. Here, the best scores are also the fastest. T3 identified 53 rationale types in 60 minutes and T4 identified 48 in 97 minutes. Therefore, no clear conclusion can be found or pattern can be identified. Unfortunately, the results cannot clarify as to why they were faster than the other participants of the same group. This may be due to chance or other unknown factors.

9.2.1.2 Design Quality

Even though the differences in both points and wins are not statistically significant, the identified difference is still slightly present. The difference of 42 points between the test group solutions (421 points and 13 wins) and the control group solutions (379 points and 11 wins) are not large enough to conclude there is any noticeable pattern. This may be due to the fact the original rationale was not used intensively enough or that this did not have the desired effect. It may simply be due to the original design rationale not having a significant effect on architecture design so that architects evaluate them better. The alternative hypotheses are rejected in order to maintain scientific practice considering the significance value of $p < .05$ are not achieved. However, these results still show that out of a 1000 repeated experiments an incorrect result can surface 55 and 63 times respectively.

9.2.1.3 Rationale Documentation

First and foremost it would seem that the inclusion of the original design rationale has a positive effect on rationale capture. The test group participants identified 20 more rationale types (178 – 158). However, the difference is not absurdly large and is also statistically insignificant. Additionally, the elaboration percentage for both research groups in moment 1 were very similar (50% and 43.7% respectively) with only a difference of 6.7% in favour of the control group participants. In moment 2, however, the difference is much larger in favour of the test group participants (21.4%). Even though both research groups identified a very similar amount of rationale types (85 and 88 respectively), the test group elaborated on them in a larger amount of cases. Interestingly, as the observed overall difference is only 20 rationale types, the difference can almost completely be attributed to the control group not elaborating on their rationale identifications as the total difference is also 20.

Therefore, it would seem that the inclusion of the original design rationale has a positive effect on rationale capture as a whole. In order to confirm this, the rationale documentation is analysed in order to record the frequency in which an architect identified the original design rationale in order to form their own rationale. For example, an architect would state benefit x of the original solution is used as an assumption when offering design option y as a possible candidate to tackle problem z. Each time this occurs, this sentence would be coded separately. As seen in the previous chapter, the control group managed to do this 36 times. The test group nearly doubled this result with 70 original rationale types. The spread of the different rationale types of the test group is also more varied whilst the control group only used the original problems and options. These were elements that could be derived from viewing the models themselves or the description of these models, elements that were included in the case. The original design rationale, however, was only present for the test group. And that difference manifests when recorded the original rationale types separately.

However, this does not necessarily confirm the observed total difference of 20 rationale types between both research groups can be explained by the inclusion or exclusion of design rationale. In order to gain more insight into this question, these two items have to be compared and analysed. Pearson's *r* is calculated to assess the relationship between the amount of identified rationale frequencies and the amount of identified original design rationale elements. There was a strong positive correlation between the two variables, *r* = .788, *n* = 8, *p* = **.020**. The result is significant at the .05 level (2-tailed). This confirms that there is a significant positive relationship between the amount of identified rationale types and the amount of original design rationale elements, as they appeared in the test group case. This result does not confirm a cause and effect relationship, however it does demonstrate the influence of the including original design rationale on the increase of overall rationale capture in their documentation, albeit a slight difference.

9.2.2 Comparisons and Correlations

This paragraph attempts to find correlations and relationships between the previously observed results. 4 questions are asked for further insight into the results. First: is there a relationship between the amount of points and wins during the ranking? Second: is there a relationship between the amount of points during the ranking and the amount of coded rationale types during documentation? Third: is there a relationship between the amount of wins during the ranking and the amount of coded rationale types during documentation? Fourth: how is the inclusion of original design rationale related to these variables?

	Ranking points	Ranking wins	Rationale frequency	Original Rationale
Ranking points	1	*r* = .910, *p* = **.002**	*r* = .253, *p* = .545	*r* = .177, *p* = .675
Ranking wins	-	1	*r* = .098, *p* = .818	*r* = .170, *p* = .688
Rationale type frequency	-	-	1	*r* = .717, *p* = **.020**
Original Rationale frequency	-	-	-	1

Table 57. Correlations moment 2

Pearson's *r* is calculated to assess the relationship between the amount of points the solutions were awarded during the ranking and the amount of wins. There was a very strong positive correlation between the two variables, *r* = .910, *n* = 8, *p* = **.002**. The result is significant at the .01 level (2-tailed). This confirms that there is a significant positive relationship between the points and wins of the solutions during the ranking.

Pearson's *r* is calculated to assess the relationship between the amount of points the solutions were awarded during the ranking and the amount of identified rationale types during documentation. There was a weak positive correlation between the two variables, *r* = .253, *n* = 8, *p* = .545. The result is insignificant.

Pearson's *r* is calculated to assess the relationship between the amount of wins the solutions achieved during the ranking and the amount of identified rationale types during documentation. There was a weak positive correlation between the two variables, *r* = .098, *n* = 8, *p* = .818. The result is insignificant.

Pearson's *r* is calculated to assess the relationship between the amount rationale types identified that belonged to the original design rationale of the original case during the ranking and the amount of points the solutions were awarded during the ranking. There was a weak positive correlation between the two variables, *r* = .177, *n* = 8, *p* = .675. The result is insignificant.

Pearson's *r* is calculated to assess the relationship between the amount of identified rationale frequencies and the amount of identified original design rationale elements. There was a strong positive correlation between the two variables, *r* = .788, *n* = 8, *p* = **.020**. The result is significant at the .05 level (2-tailed). This confirms that there is a significant positive relationship between the amount of identified rationale types and the amount of original design rationale elements, as they appeared in the test group case.

Pearson's *r* is calculated to assess the relationship between the amount rationale types identified that belonged to the original design rationale of the original case during the ranking and the amount of wins the solutions obtained during the ranking. There was a weak positive correlation between the two variables, *r* = .170, *n* = 8, *p* = .688. The result is insignificant.

9.2.3 Final Ranking

Unfortunately, most variables seem unrelated and are statistically insignificant. This result makes sense due to the differences between rationale documentation results and the architect evaluations. For example, some of the higher evaluated solutions (C1 and C4) only identified 6 and 13 original rationale types respectively. This comparison in rankings is shown in the following table.

Architecture Solution	Rank in			Average number	Average rank
	Architect ranking	Bradley-Terry probability ranking	Rationale Documentation ranking		
C1	2	1	6	3	1/2/3
C2	6	6	8	6.67	7
C3	7	7	3	5.67	6
C4	4	5	2	3.67	5
T1	1	3	5	3	1/2/3
T2	3	2	4	3	1/2/3
T3	5	4	1	3.33	4
T4	8	8	7	7.67	8

Table 58. Ranking comparison moment 2

Using the three different rankings, an average number is calculated. These numbers are then sorted from low to high, low being the highest rank. The architecture ranking and Bradley-Terry probability ranking have a high degree of agreement. The disagreement comes from the Rationale Documentation ranking, interestingly. In order to determine the agreement between the rankings, a rank correlation can be performed. This tests for ordinal association, i.e. the relationship between rankings of different ordinal variables. In this case, we have 3 ordinal variables: the 3 rankings. The rank correlation coefficient will measure the extent in which the rankings are similar.

Spearman's ϱ *is* calculated to assess the relationship between the architecture rankings and the Bradley-Terry ranking. There was a very strong positive correlation between the two variables, ϱ = .905, n = 8, p = **.002**. The result is significant at the .01 level (2-tailed).

Spearman's ϱ is calculated to assess the relationship between the architecture rankings and the rankings based on the amount of identified rationale types. There was a weak positive correlation between the two variables, ϱ = .119, n = 8, p = .779. The result is insignificant.

Spearman's ϱ is calculated to assess the relationship between the Bradley-Terry rankings and the rankings based on the amount of identified rationale types. There was a weak positive correlation between the two variables, ϱ = .143, n = 8, p = .736. The result is insignificant.

	Architect ranking	Bradley-Terry probability ranking	Rationale Documentation ranking
Architect ranking	1	ϱ = .905, p = **.002**	ϱ = .119, p = .779
Bradley-Terry probability ranking	-	1	ϱ = .143, p = .736
Rationale Documentation ranking	-	-	1

Table 59. Agreement Correlations moment 2

There seems to be a high sense of disagreement between the rationale coding analysis and the evaluations of architects. One possible explanation can be the moderating variable, the Rationale Capture Cycle. During this experiment, both research groups were given the Rationale Capture Cycle to document their architectures. The rationale documentation ranking is expressed in the amount of identified rationale types. Therefore, the use of the Rationale Capture Cycle during this experiment may have caused both research groups to identify similar amounts of rationale. If the documentation is more or less equal, the ranking of architects based on their documentation is less and less valid. Therefore it seems that if the inclusion and exclusion of design rationale is the independent variable, measuring this difference in terms of the amount of identified rationale types is an ineffective measure. It seems that, due to the Rationale Capture Cycle, both research groups submitted documentation that offered no significant differences.

However, the results do show a very high significant correlation coefficient between the architect rankings and the Bradley-Terry probability model. This implies a very high extent of agreement between the two rankings and serves to confirm the accuracy of the Bradley Terry probability model. A correlation coefficient of 1 would be perfect, i.e. the exact same ranking. These results imply that there are very few differences between the two rankings in terms of the order of their rankings. As we have determined an average rank of two different rankings and we have determined the agreement is very high and significant, a final ranking can be derived. This produces the following final ranking of solutions, based on the three different rankings.

Interestingly, a three-way first place split has occurred between T1, T2 and C1. All three averaged a rank of 3 between the three rankings. The final averaged ranking and additional contextual information of the ranking and its solutions can be found below.

		Architecture Ranking		**Bradley Terry probability model**		**Rationale documentation coding analysis**	
Final (average) rank #	**Solution**	Points	Wins	Estimated likelihood parameter (out of 8.0)	Average winning probability	Rationale Frequency	Original Rationale Frequency
1/2/3	**T1**	150	5	2.612	0.745	45	13
1/2/3	**T2**	111	5	2.619	0.746	47	24
1/2/3	**C1**	140	5	2.693	0.748	33	6
4	**T3**	100	2	0.064	0.486	56	25
5	**C4**	100	3	0.006	0.287	53	13
6	**C3**	69	1	0.001	0.127	48	12
7	**C2**	70	2	0.004	0.253	24	5
8	**T4**	60	1	0.001	0.108	30	8

Table 60. Final ranking data moment 2

T2 is declared the best solution, taken the three tests into account, followed by T4. C5 is the best control group candidate and performed relatively well compared to peers of his research group. 2/3 solutions of the top 3 are made in the test group. 4/5 solutions in the top 5 are made in the test group. There is a very distinct pattern that test group solutions are, according to various tests and comparisons, relatively better in terms of design quality and rationale documentation. This ranking is determined by coding analysis of the rationale documentation by the researcher, a statistical probability model and design quality rankings by the architects themselves.

10 CONCLUSIONS AND DISCUSSION

The following chapter will outline the major findings and conclusions of the thesis. The answers culminate towards answering the main research question and the prominent findings and conclusions of the study. This chapter also addresses the various recommendations and discussions. The main research question was concerned with attempting to find an effective way to embed design reasoning principles into architecture design and measuring its effects. This chapter will focus on these efforts.

10.1 Current Rationale Utilization

According to literature review, current design reasoning utilization and rationale capture is limited. There are various barriers to entry like the lack of industry standards, project resources and awareness among practitioners. This premise was confirmed during the interview phase of this thesis, where Sogeti practitioners agreed that these theories see little use in practice.

During the experiment phase, these theoretical assumptions were confirmed. The control group participants, during the first experiment, had no access to the Rationale Capture Cycle. These results can therefore be seen as the status quo, i.e. a baseline result that represents how an architect would design and document architectures. The control group participants were found to only document a few of all identified rationale types, and in way fewer frequency. Considering the participants knew the topic of the experiment and were forced to document their reasoning process, the utilization in practice will be worse.

Architects seem to be satisfied not providing rationale for their design decisions when a requirement is present. This is obvious when analysing the difference between identified and described rationale types. These ratios were only 50% (control group) and 43.7% (test group). This implies that for every element of justification and reasoning only half were actually described or explained. On many occasions, when faced with having to satisfy a requirement, the architect would simply choose a design solution without elaborating any further. The rationale would be labelled: ''I chose design pattern *a* because of requirement *b*''. However, no elaboration as to *why* pattern *a* satisfies requirement *b* is mentioned. Logically, this implies that, somehow, pattern *a* is the only design option that can satisfy requirement *b*. In reality, however, there are always more than one design options to be considered when attempting to satisfy a requirement. By not providing this context the architect does not demonstrate he has considered other design options, even though he might have done. Instead, the reasons for opting against them for valid reasons will remain unknown.

Even though the Rationale Capture Cycle had concrete benefits and effects, both research groups did not have stellar performance. Participants of both groups averaged around a 50% elaboration ratio, meaning that for every other justification no explanation was deemed required. Also, all participants had a very strong tendency towards certain rationale types. The singular problem-option-benefit pattern was very common during analysis. During the first experiment, the control group identified 69 individual rationale types. 60 of those fell into the problem, option and benefit category, meaning that architects were satisfied with stating a problem, figuring out a solution for it and justifying this option with a benefit. For the test group participants, this ratio was 77 out of 122. These numbers increased during the second experiment session however, but by this time the architects grew in familiarity and experience with the research. Also, both groups had access to the Rationale Capture Cycle, which increased performance. Therefore, the first experiment session best represents the average performance and awareness of the industry and is observed to be lacking.

10.2 Influence of the Rationale Capture Cycle

The original research goal was to design and develop an instrument that stimulates reasoning. Additionally, the study attempts to gain insight into what effect increased effort on reasoning and rationale capture has on architecture design and documentation, taking into account the subjective experience of the architect doing so. Using a method engineering process, the Rationale Capture Cycle was constructed: a structure model that is designed to guide reasoning and stimulate rationale capture. The effect of the Rationale Capture Cycle is measured by observing the differences between two research groups performing their architecture design process. The Rationale Capture Cycle would be given to one of the research groups so as to compare the differences. A total of 10 practicing architects from Sogeti have participated by solving an architecture case that was designed for this study.

In order to determine the influence of the Rationale Capture Cycle, the participating architects were asked to evaluate the anonymous solutions of their peers. Each architect was to spend 100 points between 3 randomly assigned solutions. During the experiment a notable difference was observed between the two research groups, both in terms of amount of points and wins of each solution. Even though these differences are not statistically significant ($p = .055$ and $p = .063$), the difference of 180 points cannot be ignored. Test group solutions scored higher, won more and ended up higher in the rankings.

Additionally, a pairwise ranking comparison model was fitted to test the architecture evaluations. This model evaluates each single matchup of solutions and crosses them with all other matchups in order to determine the likelihood of it winning a matchup. The results of this test show a very similar result, where out of the top 5 solutions 4 originated from the test group participants. The pairwise ranking probability model determines that, overall, test group solutions are notably more likely predictors for winning matchups and have demonstrably higher average winning probabilities.

Another obvious difference is observed when comparing the results of the research groups in terms of their documentation. The test group participants provided notably larger and more varied design rationale in their documentation. However, this difference is statistically insignificant due to large variance between the participants of the test group. Two participants of the test group are largely responsible for the difference between the groups, T2 and T4. Additional research into the results of those participants reveal that they made intensive use of the Rationale Capture Cycle, whilst other test group participants did less so. In order to establish the Rationale Capture Cycle was indeed the influencing factor, the documentation and the Rationale Capture Cycle were intensively analysed. A confirming premise is found through comparing the context in which the larger frequency of rationale was captured to the structure of the Rationale Capture Cycle. This was done by analysing the specific structure of the Rationale Capture Cycle and comparing it to the structure of the rationale documentation. A very strong significant positive relationship was found between the two elements. Additionally, survey answers confirm this premise. The same participants provided more constructive feedback and are comparatively more positive towards the Rationale Capture Cycle as an instrument. The Rationale Capture Cycle was found to be useful, effective, of added value during architecture design, according to these participants. The other participants answered the same questions, yet with more scepticism and signs of non-use.

In order to establish a link between the architecture rankings and the rationale documentation, correlation tests were ran. Very strong significant positive relationships were found between the evaluations of architects and the rationale documentation analysis, implying agreement between using the Rationale Capture Cycle, stimulated rationale capture and architect evaluations.

Three rankings can be derived from the data. The evaluations of the architects provide a ranking, the rationale documentation scores provide a ranking and a full ranking can be derived from the pairwise comparison probability model. For reliability purposes, an average rank will be taken from these three rankings to produce a final ranking. All three rankings are found to have a high degree of agreement between them. In this final ranking, test group solutions take the first 5 spots. An exception is found in 1 participant from the control group, who took third place. Therefore, the test group participants and their solutions consistently rank higher across multiple different tests and from different perspectives.

The data was able to demonstrate rationale capture can be stimulated by using a rationale structure model to support reasoning. This model was created through method engineering, combining relevant fragments of related studies and interviews with experts. A positive effect was observed through expert evaluations, the quality of rationale documentation and the design experience of the architect when the Rationale Capture Cycle is utilized. Therefore, reasoning can be stimulated by using a reasoning structure model during architecture design. This way, design reasoning principles can be embedded into architecture design and its beneficial effects measured and demonstrated.

10.3 Discussion

10.3.1 Presence of Original Design Rationale (Moment 2)

During the first experiment session the goal was to determine whether the capture of design rationale can be stimulated and reasoning can be supported. The first session found that the Rationale Capture Cycle did indeed do so. The second experiment session attempted to gain more insight into the effect of having design rationale present during architecture activities. A second experiment was designed with the inclusion and exclusion of design rationale in mind. The architecture solutions from the first experiment session, including their rationale, was used as case material. In order to demonstrate the effect of having or not having design rationale, the architects were asked to reconsider the solutions made in moment 1 and implement changes or make modifications based on updated stakeholder requests and concerns. The test group would have access to the design rationale as it was made by the original architect during moment 1, the control group would have to make do with a brief description of the architecture. The assumption is that having access to the original rationale and context in which the architecture was designed would have a measureable and beneficial influence on architecture design. A total of 8 practicing architects from Sogeti have participated by solving an architecture case that was designed for this study.

Using the same measures as in the first experiment moment, slight differences can be found between the research groups in favour of the test group. Yet these differences are all statistically insignificant. The test group solutions were found to be awarded slightly more points and obtained a few more wins during the architecture rankings. Also, the rationale documentation of the test group participants identified and described slightly more rationale types. The frequency in which the original design rationale was used during documentation was recorded separately. Test group participants using the original rationale a lot more frequently than the control group participants did. The use of the original rationale has a significant positive relationship on the increased rationale frequency scores of the test group participants, yes, but the overall increase is too slight to be notable. Additionally, a pairwise ranking comparison model was fitted to test the architecture evaluations. The pairwise ranking probability model determines that, overall, test group solutions are notably more likely predictors for winning matchups and have demonstrably higher average winning probabilities.

Even though there was a high degree of inter agreement between the architect rankings and the pairwise ranking probabilities model, the rankings of the rationale documentation disagreed. A possible explanation is that the influence of the Rationale Capture Cycle as a moderating variable is too significant and disrupted the results of one the coding analysis as a measure. During this experiment, both research groups were given the Rationale Capture Cycle to document their architectures. The rationale documentation ranking is expressed in the amount of identified rationale types. Therefore, the use of the Rationale Capture Cycle during this experiment may have caused both research groups to identify similar amounts of rationale. If the documentation is more or less equal, the ranking of architects based on their documentation is less valid as the architects will base their evaluations on the models alone. Therefore it seems that if the inclusion and exclusion of design rationale is the independent variable, measuring this difference in terms of the amount of identified rationale types is an ineffective measure. It seems that, due to the Rationale Capture Cycle, both research groups submitted documentation that offered no significant differences to the architect. This may have caused the architect to base their evaluations more on the models instead. This theory would make sense, as the coding analysis results are very close to each other across both research groups. This would make a 1-10 ranking increasingly ineffective and explain the disagreement between the rankings.

Another theory for the lack of unambiguity in the results may be the way the second experiment case was formed. All 8 participants mentioned missing information through the survey, which may indicate a larger problem. Furthermore, the results of solutions in terms of content vary wildly and are basically incomparable. For example, the priorities for each solution are completely different. One architect would focus on a complete infrastructure model whilst another only models certain information streams. This may be due to the fact that the problems that are indicated in the case are not clear enough, which causes the solutions to become increasingly erratic. The stakeholder concerns that are indicated in the case are not explained clearly, as they only number one term. The reason for this is the length of case, which already was becoming too lengthy and diluted. A choice was made for the case to offer 3 different options, with each of these options having their own unique stakeholder concern. This did lengthen the case, as there were more options to tackle, but limited the depth. As the stakeholder information decreased, the case was not complex enough. Architects were more often inclined to provide a simple textual remedy of the stakeholder concern, instead of providing architecture designs or implementing changes. This caused the experiment to become ineffective, as some architects were not challenged or forced to make complex design decisions. In turn, this may have caused architects to not make use of the original design rationale as it was not necessary to complete the case. This may have been the factor that caused the results to be erratic and invalid. As some architects provided forced changes and larger models, whilst other architects interpreted the case as a simple textual alleviation of various stakeholder concerns. This large variance between the individuals, especially with a low sample size, caused the data to be ineffective and unusable. Unfortunately, the fault most likely lies with the way the second experiment was designed and executed. The theory that original design rationale will help an architect to interpret the architecture still probably holds true, but the experiment results was not designed in a way that this result would indeed surface during the research.

10.3.2 Research Validity

This paragraph will deal with any threats that may negatively impact the validity of the research results. The goal is to address any concerns regarding internal and external validity and reliability. The theory behind the validities can be found in appendix 2.4.4, Validity.

10.3.2.1 Internal validity

First and foremost, there are some validity threats regarding the empirical experiment and its participants. Familiarity is a potential influencing variable; the extent in which some participants are more familiar with the research than others. For one, some of the participating architects were interviewed in an earlier stage. During the first experiment session, this was not an issue as all participants of moment 1 were interviewed. In moment 2, however, some of the participants were not interviewed beforehand. This threat was addressed in two ways. First, the architects that were not interviewed are equally split between both research groups to make sure the difference, if any, is equal in both groups.

Second, all participants are given the same elaborate introduction before each experiment session. A 30 minute presentation and introduction is given before each experiment session to ensure all participants have equal knowledge before going into the experiment. Each architect is different. This may influence results, so the architects have to be split as best as possible. While assigning the participants into groups the following influencing variables are taking into account to ensure as best a comparison as can be made: the age, experience, content of that experience and the title of the architect. The controlled variables can be seen in chapter 8, Results. Here, the results are analysed with this context in mind.

Additionally, some factors regarding the case have to be controlled in both moments. In order to make sure the independent variable is the only influencing factor, other elements have to be controlled. For example, the notation, template, tool, language and other case specific items are equalized to make sure the only comparable difference can be measured in the dependent variable is the independent variable. However, some architects are more familiar with this type of architecture as it converges on design. Therefore, the equal split of architects is an important distinction to make. One participant, who nearly did not submit anything due to not being at home with the content of the case, was removed from the comparison in moment 2.

Another influencing factor with regards to the quality rankings are the familiarity of all participants with each other. It may be possible that some architects are very familiar with the way of writing, style and output habits of the participants they have to rank. As there is no way to control this, considering there is no way to know which architects work more closely with each other than others, all architects are to rank 3 solutions. If the scenario occurs when an architect recognizes whom he has to rank, there is still a third solution that is involves. Additionally, an important emphasis is placed on the ranking and the integrity of the participants: the architects are supposed to rank the solutions and not the individuals. This premise is emphasized thoroughly, both during the experiment and during the ranking e-mail with instructions.

10.3.2.2 External validity

Due do the explorative nature of this study, generalizability is not the main goal. The research does not attempt to generalize these results onto a larger population. For that goal, the sample size of 18 architects split across two experiment moments is too low. This is emphasized during the significance testing, where a large portion of the comparisons show an insignificant difference. Therefore, to maintain scientific integrity and practice, all alternative hypotheses were rejected. However, when additional analysis is performed, interesting significant results surface regarding two notable participants. When thoroughly analysed, a significant link with the independent variable was found. This link implies there is a positive relationship between rationale capture and the Rationale Capture Cycle. It does not ‘prove’ a causal link, yet it does show a significant *demonstration* of the instrument’s positive influence. This was the main goal of the thesis.

10.3.2.3 Reliability

Multiple active steps are taken to ensure the results are reliable. First, the entirety of the quality ranking test serves as a reliability test for the other results. As the rankings are determined by professionals, this provides an extra layer of reliability. Especially when the rankings agree with the coding analysis of the documentation. The rankings are performed by practicing architects and are not influences by the researcher. The rankings of both experiment moments is found that the relative evaluations of the architects agree with the results of the rationale documentation coding analysis. The agreement of the results by the rankings of the architects provides a confirmation of the thesis results and provides an extra element of reliability to the study. Also, the Bradley Terry model provides an extra layer of reliability as it uses probabilities statistics to determine the likelihood of winning versus other solutions. Comparing and averaging these results leads to a thorough shield against reliability threats.

Additionally, two sanity check sessions are held with external architects. During these sessions, the goal is to quickly determine whether the results of the experiment can be true or follow any logic or sense. The following experts have participated during the sanity check sessions. For anonymity purposes, the employers and titles are random.

Architect	Background in	Current employers
A	Both Enterprise Architecture and Information Management	Major financial institution / large educational institution
B		

Table 61. Sanity check session participants

The sessions progress as follows: first, the participants are given an introduction to the research and its goals, objectives and methods. In the case of this study, the case material is read by the participant to understand the various solutions in context. Then they are presented with the results as they have appeared during the research. All 18 solutions are presented hardcopy on table, ranked as they are during the experiment session.

The participants are asked to quickly evaluate each solution and judge them by their result. The participants are asked whether or not the result seems likely and logical, given the researches' goals and methods. The results are as follows:

In the first session, architect number 1 started by analysing the results of the best (#1) and worst (#10) solution. Immediately, solution 10 was found to be rather meaningless whilst solution 1 was *''by far the best''*. From a managerial perspective, solution 1 provided much more insight into the reasoning process than solution 10. Solution 1 offered a logical process and structure into their design and clearly evaluated different options. This proved to be the most effective solution. Additionally, architect number 1 mentioned that the first 5 solutions are good and from solution 6 upwards become worse and worse. This mirrors the result of the experiment sessions greatly. In other words, the evaluation of architect number 1 greatly agrees with the result of the experiment session. Architect number 1 even ranked the solutions himself and ordered them differently. However, the split between 1 through 5 and 6 through 10 remained the same. Only the order was different, as is shown in the following table.

Solution	Rank during research	Sanity Check rank
T2	1	1
T4	2	3
C5	3	2
T1 / T3	4	5
T1 / T3	5	4
C1	6	9
T5	7	6
C2	8	10
C4	9	8
C3	10	7

Table 62. Sanity check session moment 1

The experiment session result is the average rank of both the architect rankings and documentation analysis. Even though architect number 1 did not match the original result, the overarching result is in agreement. The first five and last five remain on the same sides, only in different orders. Spearman's ϱ is calculated to assess the relationship between final ranking of moment 1 and the ranking done during the Sanity Check session. There was a strong positive correlation between the two variables, $\varrho = .830$, $n = 10$, $p =$ **.003**. The result is significant at the .01 level (2-tailed). Therefore, the result of the sanity check session for architect number 1 is considered to be confirming the result.

In the second session, the second participant immediately found a pattern of increasing elements and aspects in the visual models the higher the rank was of the solution. However, no real clear pattern in terms of content could be found. Additionally, the distinction between specific solutions were increasingly vague and unclear. Architect number 2 could not find any real reason as to why certain solutions ranked better than others. He also found that, in terms of the content, each solution had results that varied wildly. Solutions prioritized differently or answered completely different problems. To summarize, architect number 2 found no concrete patterns or logical order in the ranking. This confirms the results and analysis of experiment session 2, where the solutions indeed performed very similarly both in terms of documentation and design quality. As the results of the second experiment session were not unambiguously, so was the evaluation the second architect during this session. This more or less confirms the analysis of the results of the second experiment moment, even if those results do not show the desired effect.

10.4 Recommendations

10.4.1 Using the Rationale Capture Cycle Reasoning Model

During practice, the reasoning model is often seen as a stepwise procedure. It is essential to not view rationale as a checklist where each and every element has to be checked in order to continue. It will be up to the architect to gauge and determine to what extent the elements need to be explicated. In some circumstances, certain assumptions or risks need not be applied. The reasoning model was designed with simplicity adoption in mind. It is not a strict procedure but can be applied if needed. It is also very important not to ignore certain elements of the model once you've passed them. The model is designed to support reasoning. One of those elements is to combat anchoring bias. It is essential that for every step you re-evaluate your own options and completed work. Design reasoning should ideally be captured as a by-product. It should naturally be captured if the model is seen as an instrument during regular architecture design activities. The model should not be seen as a product itself. It should, however, be taken into account when working under and with architecture.

10.4.2 User Adoption

The model was very specifically designed with user adoption in mind. This is reflected in the visual design and simplicity of the reasoning model. In order to guarantee user adoption in Sogeti, the artefact was made through various feedback, validation and input stages. During the interview phase, architects were asked to provide input with what they though rationale entailed and what rationale elements are important. During both experiment sessions architects had the opportunity to provide input through a survey. Also, architects were asked what rationale information they used most or least. Additionally, during the experiment sessions, plenary discussion session were held to discuss the Rationale Capture Cycle.

10.4.3 Place in Architecture Process

During the research it was clear that architects did not always understand when to use this reasoning model. It is very important to determine when this model can be applied, yet that is not up to the research to prescribe. One of the main goals of the Rationale Capture Cycle is that it has to be general enough to be applicable in most situations. This also benefits user adoption. In order for the reasoning model to be used, it cannot be specific to certain situations or moments during the architecture process. For one, there is no concrete architecture process. This process is very different per organization, department, domain, topic and individual. Therefore, the reasoning model is designed to be used as a tool when the architect deems it necessary. If prescribing or forcing the model into certain documents or procedures, it defeats the purpose of critical thinking and evaluation.

10.5 Limitations

This paragraph discusses various recognized limitations of the study and their impact on the results and how they are addressed or justified.

10.5.1 Risk versus Reward

The research revolves around the premise that the utilization of design reasoning is beneficial. However, no hard calculations are made on to what extent this is true compared to the costs that may be require to use it. This is not done due to the complexity of such calculations. Also, this calculation differs per organization and department and is unfeasible to determine precisely. The benefits of design reasoning during this study are therefore purely theoretical and not proven in any conclusive manner. The benefits would have to be financially determined in terms of maintenance costs and risk analysis or in terms of architecture quality evaluation. However, both fields are entirely different studies.

10.5.2 Generalizability

Quantitative analysis is limited due a low sample size. Considering the exploratory nature of the study and the qualitative nature of the research method this is also not suitable. However, the qualitative aspect does have an impact of the generalizability of the research results. The limited resources of this project also constitute a post-test only control group design as the participants cannot participate in the project more than once. In the case of the Sogeti professionals, the test group selection is also random to control the influencing factor of experience which varies per professional. This might cause the experiment results to be less valid. Also, the test results are hardly generalizable considering the only participants were Sogeti professionals. Results may vary when architects of other organizations participate.

10.5.3 Rationale Capture Cycle Usage

As not all test group participants fully utilized the model, the results are smaller than they could have been. Only 2 participants made full use of the Rationale Capture Cycle during the first experiment, causing a large difference in the comparison. Further research into how such a model can be fully implemented during architecture design is needed. We believe that with more research into actively embedding the reasoning model in a more intuitive manner, further results can be achieved.

10.5.4 Domain Applicability

Many studies and assumed theory used by this research is specific to software architecture. This study is based around the assumption that design reasoning principles and theories are generalizable to systems architecture as a whole, including all types. The key focus is the process of architecture design itself, regardless of architecture domain. However, limited concrete studies have been performed to support this as design reasoning is usually studied in the context of software architecture. A paper by Plataniotis et al. (2014) suggests that, indeed, the omission of design rationale is a concept stemming from software architecture. However, it very much so applies to enterprise architecture as well, including its risks.

Also, multiple statements refer to the generalizability of design reasoning to other types of systems architecture. Namely, Bass et al. (2003), authors of *Software Architecture in Practice*. They mention systems and enterprise architecture have a great deal of similarity with respect to software architecture. All can be similarly designed, evaluated and documented. They are designed to satisfy stakeholder demands, consist of elements, structures and relationships and answer to requirements. The difference is in the scope of the architecture and utilized technical tools.

This sentiment is confirmed by Rozanski & Woods (2012), authors of *Software Systems Architecture*, who mention in their book that enterprise architecture is similar to software architecture yet is a broader type of scope. Lankhorst (2009) supports this sentiment, as he mentions many enterprise architecture frameworks can be extended to systems architecture in general. Also, this sentiment is supported by the supervisor of this thesis, who is assistant professor in Software Architecture at Utrecht University. This assumption is also supported by the thesis supervisor who represents Sogeti, who holds a PhD in Enterprise Architecture from Utrecht University.

10.5.5 Case & Experiment Setup

Certain limitations occurred during research when designing and executing the two different experiment sessions. These limitations are both regarding the case and overall experiment design. Firstly, many times architects did not finish time. Statements regarding the lack of time were quite common. Unfortunately, due to the lack of time, there is no way to determine what kind of results would have surfaces if the architects were given more time. Due to the nature of the case and experiment setup however, more time would not be feasible. A lot of time is necessary given the nature of data input and analysis, however. Also, architects mentioned that the case did not always mimic a real life scenario. Therefore, the status quo data might not be how architects actually design and document. This also contributes to the controlled experiment setting the architects experienced. The architects also knew they were supposed to document and provide rationale, therefore the results might not be as accurate as they can be. Additionally, architecture design activities are not done individually. During this case, all data and input is gathered through individual cases. In reality however, architects rarely work alone. They prefer to gather knowledge through discussion. Unfortunately, due to the nature of the case, this is not possible. The risk of this would be a loss of data when too much discussion arises on a certain topic. Discussion can also lead off-topic. Also, recording each discussion is very resource intensive to do. For an individual researcher, individual cases with equal data is the most effective choice for data gathering. Lastly, there are some minor limitations regarding the experiment setup. Most experiment sessions were held at night, which may influences the results as architects will be incentivized to quickly finish the case as they can go home. Also, the two hours which architects were given are found to be too short for this type of research. As virtually no architect finished in time, especially during the first experiment session, results may be biased. Lastly, architects may not take the subject material serious as the case is fairly long and the researcher is a student and no expert.

10.6 Future Research

10.6.1 Reasoning and Design Psychology

During this thesis, the design reasoning principles are based off of general human psychology. Much more research can be done into the psychology of an architect during design. For example, when does an architect list assumptions and during what activities? You could perform similar studies and include a Belbin personality test. Through this test, extra personality context is taken into account when performing design activities. During this research many architects provided wildly different results. Therefore, more research into individual habits and personalities can lead to interesting results in the context of architecture design.

10.6.2 Tool Support and Knowledge Management

For now, most architecture is design ad hoc and documentation is kept through Word templates. A logical follow-up would be to research how to explicate knowledge automatically. Using a tool to capture rationale would drastically reduce the overhead of documentation of architectures. Design rationale should, as mentioned before, be captured as a by-product. The most efficient way to achieve this is through proper tooling and automation software. Unfortunately, as of now, this software is not effective yet in the state of the art. Additionally, if this knowledge proves to be valuable, how can this then be used for knowledge management purposes? There are various studies that perform automatic rationale explication through recording audio sessions. If this knowledge can be correctly explicated, how can this knowledge be automatically uploaded into knowledge banks?

10.6.3 Further Experimentation

It would be very interesting to conduct more experiments. One of the limitations is the generalizability of the results. For one, most architects came from Sogeti. However, all of these are working in various different organizations. This does contribute to the generalizability somewhat. Unfortunately, only 18 architects participated across the two experiment sessions. The results of the Rationale Capture Cycle are promising, but the inclusion of design rationale was not yet. Perhaps with more participants from various demographics the results might prove very valuable.

Also, another interesting aspect would be to include various different demographics into the research. During this research, the average age of the participant is around 50 years. The average work experience in years is around 20 years. This group might have significant differences from other architects and designers, as the group may well be unlikely to adapt to new ways of working. Additionally, nearly every senior architect never had formal education in this domain which may influences results.

10.6.4 Implementation into Architecture Processes

This thesis attempts to raise awareness and tries to demonstrate how to capture rationale in a simple manner. It also tries to show it has benefits to rationale documentation and the evaluations of peers (architects). This research does not show how to implement this model into current business practices. Further research should be done on how this principles can effectively be embedded into modern architecture templates and practices. These would require specific case studies, as there are no real standards when it comes to architecture documentation templates or processes. However, very in depth research into embedding architecture design into real departments or organizations can lead to great results.

10.7 Significance

The research aimed to find and demonstrate a comprehensive and intuitive manner in which architects can consistently employ design reasoning principles in the architecture design process. The distinction between scientific and practical contribution is made due to their different natures. The goal of research is to add to the body of existing knowledge and expand its borders. The goal of practical contribution is to add value, in some way, to the business process of architecture design.

10.7.1 Scientific Contribution

The main goal of science is to add knowledge to the knowledge base and to expand the borders of the concept of knowledge. Therefore, the aim of this research is to add knowledge in some way, shape or form. The main scientific contribution can be formulated as a new insight in which a nonintrusive and simple method can effectively demonstrate the stimulation of design reasoning among architecture designers and a clear step towards standardization among design rationale capture and documentation methods.

As explored in the literature review, the lack of a standardized notation and language to guide design reasoning and represent design rationale is a clear industry problem. This research has analysed the status quo and attempted to distinguish clear issues among the practitioners in the industry. The method that is to be produced works towards a simple and effective manner that stimulates design reasoning principles, whilst working towards standardization in the industry. In other words, the borders of knowledge have slightly expanded in terms of the simplicity in which design reasoning principles can be effectively applied and demonstrated. But also, this study works towards standardization in the field of capturing and documenting design rationale by designing a method that is based on analysing current industry practices and their strengths and shortcomings.

10.7.2 Practical Contribution

Having addressed the scientific aspect of this research, the practical element cannot be ignored. Since this project is hosted and facilitated by Sogeti Netherlands B.V., the results have to constitute business value in some way. The main business value the research produces can be expressed as follows: a nonintrusive and comprehensive method in which design reasoning principles are intuitively embedded, that has been shaped by, and for, architecture designers of Sogeti. The research has demonstrated that the utilization of this method effectively stimulates the use of design reasoning among architects in the architecture design process. Concurrently, the literature review has theoretically determined that the use of design reasoning has concrete benefits. Therefore, the conclusion of this research produces a method that is semi-specific to Sogeti architects that has concrete benefits to their architecture business process.

> *''It is found that current reasoning and rationale capture is very limited. During the thesis a reasoning model could be constructed and was found to be beneficial in combating this shortcoming. However, the research has some shortcomings and limitations as the data is not perfect. Yet, the definite influence of the Rationale Capture Cycle cannot be ignored.''*

11 REFERENCES

Baron, J., & Hershey, J. C. (1988). Outcome bias in decision evaluation. *Journal of personality and social psychology*, 54(4), pp. 569.

Bass, L., Clements, P., Kazman, R. (2003). *Software Architecture in Practice*. Addison-Wesley, Boston.

Bosch, J. (2004) Software architecture: the next step. *Proceedings of the 1st European Workshop on Software Architecture* (EWSA), St. Andrews.

Bradley, R. A., & Terry, M. E. (1952). Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika, 39(3/4), 324-345.

Burge, J., & Brown, D. C. (1998). Design Rationale Types and Tools. *AI in Design Group*.

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and software technology*, 38(4), pp. 275-280.

CIO Council. (1999). Federal enterprise architecture framework version 1.1.

Clements, P., Garlan, D., Bass, L., Stafford, J., Nord, R., Ivers, J., & Little, R. (2002). *Documenting software architectures: views and beyond*. Pearson Education.

Conklin, E. J., & Yakemovic, K. C. (1991). A process-oriented approach to design rationale. *Human-Computer Interaction*, 6(3), pp. 357-391.

C4ISR Architecture Working Group. (1997). C4ISR architecture framework version 2.0.

Deneckère, R., Hug, C., Onderstal, J., & Brinkkemper, S. (2015). Method Association Approach: Situational construction and evaluation of an implementation method for software products. *Research Challenges in Information Science* (RCIS). IEEE 9th International Conference on (pp. 274-285). IEEE.

Dietrich, C. (2010). Decision making: factors that influence decision making, heuristics used, and decision outcomes. *Student Pulse*, 2(02).

Do, E. Y. L., & Gross, M. D. (1996). Drawing as a means to design reasoning. *AI and Design*.

Edwards, W. (1968). Conservatism in human information processing. *Formal representation of human judgment*, pp. 17-52, New York: Wiley.

Epley, N., & Gilovich, T. (2006). The anchoring-and-adjustment heuristic why the adjustments are insufficient. *Psychological science*, 17(4), pp. 311-318.

Evans, J. S. B. (2003). In two minds: dual-process accounts of reasoning. *Trends in cognitive sciences*, 7(10), pp. 454-459.

Feiler, P. H., Gluch, D. P., & Hudak, J. J. (2006). The architecture analysis & design language (AADL): An introduction (No. CMU/SEI-2006-TN-011). Carnegie-Mellon University, Pittsburgh PA. Software Engineering Inst.

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. I. (1991). Making argumentation serve design. *Human–Computer Interaction*, 6(3-4), pp. 393-419.

Friedenthal, S., Moore, A., & Steiner, R. (2014). A practical guide to SysML: the systems modelling language. Morgan Kaufmann.

Foorthuis, R.M., Brinkkemper, S. (2008). Best Practices for Business and Systems Analysis in Projects Conforming to Enterprise Architecture. *Enterprise Modelling and Information Systems Architectures*, Vol. 3, No. 1, pp. 36—47.

Gero, J.S. & Kannengiesser, U. (2014). The function-behaviour-structure ontology of design. In A. Chakrabarti and L.T.M. Blessing (Eds.). *An Anthology of Theories and Models of Design.* Springer, pp. 263–283.

Greefhorst, D., & Proper, E. (2013). The role of enterprise architecture. *In Architecture principles*, pp. 7-29.

Gruber, T. R., & Russell, D. M. (1992). Derivation and use of design rationale information as expressed by designers, 39.

Hämäläinen, N., & Markkula, J. (2007). Quality Evaluation Question Framework for Assessing the Quality of Architecture Documentation.

Hilliard, R. (2007). All about IEEE Std 1471. *IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Std 1471-2000).* New York: IEEE Computer Society.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, 28(1), pp. 75–105.

Hong, S., Goor, G. van den, Brinkkemper, S. (1993). A formal approach to the comparison of object-oriented analysis and design methodologies. *Proceedings of the 26th Annual Hawaii International Conference on System Sciences*, Hawaii, pp. 689–699.

ISO/IEC/IEEE. (2011). Systems and software engineering – Architecture description, ISO/IEC/ IEEE FDIS 42010:2011, *International Organization for Standardization*, Geneva.

Josey, A. (2011). TOGAF® Version 9.1-A Pocket Guide. Van Haren.

Kahneman, D, Shane, F. (2002). Representativeness Revisited: Attribute Substitution in Intuitive Judgment. In Thomas Gilovich; Dale Griffin; Daniel Kahneman. Heuristics and Biases: The Psychology of Intuitive Judgment. Cambridge University Press, pp. 51–52.

Karsenty, L. (1996). An empirical evaluation of design rationale documents. *Chi 1996*, 150–156.

Kruchten, P. (1995). Architectural Blueprints — the "4+1" View Model of Software Architecture. *IEEE Software*, 12 (6), pp. 42-50.

Khatri, N., & Ng, H. A. (2000). The role of intuition in strategic decision making. *Human relations*, 53(1), pp. 57-86.

Kunz, W., & Rittel, H. W. (1970). Issues as elements of information systems (Vol. 131). Berkeley, California: Institute of Urban and Regional Development, University of California.

Lankhorst, M. (2009). *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer: Berlin, Heidelberg.

Lee, J. (1997). Design rationale systems: Understanding the issues. *IEEE Expert-Intelligent Systems and Their Applications*, 12, pp. 78–85.

Lee, J. (1989). Decision representation language (DRL) and its support environment.

Leite J., Oquendo F., Batista T. (2013) SysADL: A SysML Profile for Software Architecture Description. In: Drira K. (eds) Software Architecture. ECSA 2013. Lecture Notes in Computer Science, vol 7957. Springer, Berlin, Heidelberg

Luinenburg, L., Jansen, S., Souer, J., Van De Weerd, I., & Brinkkemper, S. (2008). Designing web content management systems using the method association approach. *In Proceedings of the 4th International Workshop on Model-Driven Web Engineering (MDWE 2008)* (pp. 106-120).

MacLean, A., Young, R. M., Bellotti, V. M., & Moran, T. P. (1991). Questions, options, and criteria: Elements of design space analysis. Human–computer interaction, 6(3-4), 201-250.

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

Mather, M.; Shafir, E.; Johnson, M.K. (2000). Misrememberance of options past: Source monitoring and choice. Psychological Science, 11 (2), pp. 132–138.

McCall, R. (2013). Critical conversations: Feedback as a stimulus to creativity in software design. In Creativity and Rationale (pp. 11-40). Springer London.

Miles, M. B., & Huberman, A. M. (1994*). Qualitative data analysis: A sourcebook*. Beverly Hills: Sage Publications.

Milkman, K. L., Chugh, D., & Bazerman, M. H. (2009). How can decision making be improved? *Perspectives on psychological science*, 4(4), pp. 379-383.

Oates, B. J. (2005). *Researching information systems and computing*. Sage.

Object Management Group. (2011). BPMN 2.0 Spec. OMG.

Oswald, E., Grosjean, S. (2004). Confirmation Bias. In Pohl, Rüdiger F. Cognitive Illusions: A Handbook on Fallacies and Biases in Thinking, Judgement and Memory. Hove, UK: Psychology Press. pp. 79–96.

Parnas, D. L., & Clements, P. C. (1986). A rational design process: How and why to fake it. IEEE transactions on software engineering, (2), 251-257.

Perry, D. E., & Wolf, A. L. (1992). Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*, 17(4), 40-52.

Peterson, J. L. (1981). Petri net theory and the modelling of systems.

Plataniotis, G., de Kinderen, S., & Proper, H. A. (2014). Capturing design rationales in enterprise architecture: A case study. *Lecture Notes in Business Information Processing*, *197*, 133–147.

Poort, E. (2014). RCDA: Risk- and Cost-Driven Architecture: A Solution Architect's Handbook.

Rai, L.; Kang, S.J. (2008). Rule-based modular software and hardware architecture for multi-shaped robots using real-time dynamic behaviour identification and selection. *Knowledge-Based Systems*, 21 (4), pp. 273–283.

Regli, W. C., Hu, X., Atwood, M., & Sun, W. (2000). A Survey of Design Rationale Systems: Approaches, Representation, Capture and Retrieval. *Engineering with Computers*, *16*(3–4), 209–235.

Rittel, H. W. (1987). The reasoning of designers. Montreal: IGP.

Rittel, H. W., & Webber, M. M. (1973). Dilemmas in a general theory of planning. Policy sciences, 4(2), 155-169.

Rozanski, N., Woods, E. (2011). Software Systems Architecture: Working with Stakeholders using Viewpoints and Perspectives. Addison-Wesley.

Runeson, P., Skoglund, M. (2009). Reference-based search strategies in systematic reviews. *In Proceedings of the 13th International Conference on Empirical Assessment and Evaluation in Software Engineering*. Electronic Workshops in Computing (eWIC). UK: BCS, Durham University.

Runeson, P., Host, M., Rainer, A., & Regnell, B. (2012). *Case Study Research in Software Engineering*. John Wiley & Sons, Inc.

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified modelling language reference manual*. Pearson Higher Education.

Scheer, A. W. (2012). *Architecture of integrated information systems: foundations of enterprise modelling*. Springer Science & Business Media.

Schilling, J. (2006). On the pragmatics of qualitative assessment. *European Journal of Psychological Assessment*, 22(1), 28-37.

Shum, S. B., & Hammond, N. (1994). Argumentation-based design rationale: what use at what cost. *International Journal of Human-Computer Studies*, *40*(4), 603–652.

Soley, R. (2000). Model driven architecture. *Object Management Group white paper*, 308(308), 5.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM systems journal*, 31(3), 590-616.

Tang, A., Babar, M. A., Gorton, I., & Han, J. (2006). A survey of architecture design rationale. *Journal of Systems and Software*, 79(12), pp. 1792–1804.

Tang, A., Jin, Y., & Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. *Journal of Systems and Software*, 80(6), 918–934.

Tang, A., Tran, M. H., Han, J., & Van Vliet, H. (2008). Design reasoning improves software design quality. In S. Becker, F. Plasil, & R. Reussner (Eds.), Quality of Software Architectures. Models and Arhcitectures. 4th International Conference on the Quality of Software-Architectures, QoSA 2008, Germany, Karlsruhe (Vol. 5281 LNCS, pp. 28–42). Berlin, Heidelberg: Springer

Tang, A., & van Vliet, H. (2009). Software Architecture Design Reasoning. *In Software Architecture Knowledge Management*, pp. 155–174.

Tang, A. (2011). Software designers, are you biased? In Proceedings of the 6th International Workshop on Sharing and Reusing Architectural Knowledge (pp. 1-8). ACM.

Taylor, R. N., Medvidovic, N., & Dashofy, E. M. (2009). Software architecture: foundations, theory, and practice. Wiley Publishing.

The Open Group. (2012). Archimate 2.0 Specification. The Open Group, Berkshire, UK.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive psychology*, 5(2), pp. 207-232.

Tversky, A., & Kahneman, D. (1975). Judgment under uncertainty: Heuristics and biases. *In Utility, probability, and human decision making*, pp. 141-162, Springer, Netherlands.

Tyree, J., & Akerman, A. (2005). Architecture decisions: Demystifying architecture. IEEE software, 22(2), 19-27.

Vaishnavi, V., & Kuechler, W. (2004). Design research in information systems.

Verries, J., Sahraoui, A. E. K., & Paludetto, M. (2008). Design rationale in system design. *Proceedings of 19th International Conference on Systems Engineering, ICSEng 2008*, 380–385.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). Experimentation in software engineering. Springer Science & Business Media.

Zsambok, C. E., & Klein, G. (2014). Naturalistic decision making. Psychology Press.

Zachman, J. A. (1987). A framework for information systems architecture. IBM systems journal, 26(3), 276-292.

APPENDICES

1 INDUCTIVE INTERVIEWS

The information below features the interview process, which was held early during the thesis phase. It offers the initial approach, participants and results.

1.1 Aim & Approach

Interviews will be held with Sogeti architecture practitioners in a one on one setting. The interviews will adhere to a semi structured approach, meaning the main areas and topics of conversation are predefined but the conversation is not completely scripted (Oates, 2005).

The main aim of the interview phase is to gain expert knowledge on the subject area. As the research is of exploratory nature, expert knowledge can provide valuable context and information. Because of this, the interviews will follow an inductive approach, which means the knowledge found may shape the research as opposed to a deductive approach that tests predefined theories and ideas (Bhattacherjee, 2012). As in the interviews are mainly exploratory and inductive, complete transcription is deemed unnecessary. The knowledge gathered is meant to serve as input and feedback, rather than concrete testing and validation. The findings are therefore not generalizable, nor do they need to be. The strictly provide context for the main results of the experiment and serve as input and influence the artefact. The audio recordings are still made, however, for the sake of transparency.

The interview protocol is semi structured, which means overall covered themes are predefined. These themes stem from the proposal phase, where literature review has shaped the overall topics regarding the research subject. These themes include, but are not limited to, the following:

- Architecture Process
- Design Reasoning utilisation
- Barriers to Design Reasoning and Rationale
- Rationale Capture
- Industry Opinion
- Personal Experience

The questions asked are centred on his or her own experience and opinion of the subject matter, finding concrete examples in their work process and discussion on how he or she would approach the topic and tackle the issue. Interviews were held among 12 professionals from Sogeti Netherlands, currently working from various external clients in the Netherlands. The goal for interviewee selection was to select architects that were closely involved with the architecture working process. In other words, the selected architects make daily design decisions. The interviewees were all currently practicing architecture and consisted of 2 solution architects, 4 enterprise architects and 6 information architects. The professionals practiced architecture in various domains and organizations, including banks, governmental institutions, public transport organizations, private financial organizations and others. The interviews took place at the offices of the clients where the architects were currently working from.

1.2 Results

This paragraph summarized the main findings that surfaced during the interviews. These points contain the main conceptions, interpretations and opinions that were mentioned most among the architects, whether they agreed or not. The summarized points are selected on the frequency of mention and relative importance or emphasis placed by the interviewee.

1.2.1 Awareness and Utilization

Virtually all interviewees mentioned that, indeed, naturalistic decision making is prevalent in the industry. In the majority of cases the architects admitted, depending on the maturity of the organization and nature of the project, design rationale capture was omitted or limited. Also, little attention was given on reflecting on the reasoning process. The lack of maturity in this topic manifests in various different ways, as uttered by the interviewees. For example, architects felt that they had to assume too many rationales behind design decisions for this information was not kept during previous projects. This causes architects to spend more time finding knowledge in the organization through interviews, causing a clear loss in efficiency. They also mentioned they would prefer to be able to use rationale in order to improve collaboration among architects. In many cases, architects discuss finished architecture designs instead of the reasoning process which causes ineffective communication and collaboration.

Some interviewees were more nuanced, however, stating that they do indeed provide elaborate rationale of design decisions in Project Starting Architectures (PSA) or Architecture Definition Documents (ADD) when the nature of the project and hosting organization required them to do so. This was only the case for very mature organizations who deal with regulatory influences or financial principles daily, e.g. banks and tax offices. In many cases when rationale was, in fact, produced, the architect would only do so for a single moment in time, e.g. a stakeholders meeting or board approval session. Rationale is not kept as a process: it is not consistently produced nor updated throughout a project's lifetime.

A handful of architects, who didn't participate in a full interview, declined due to not feeling as though they could provide useful input. The reason given was that they did not provide much rationale in their daily work or have not considered the concepts and their benefits at all. However, in most cases, the awareness was present. Most architects considered the ideas of design reasoning useful, clearly lacking in current working processes and needed attention and growth in maturity. However, some barriers to full utilization still exist that prevent full maturity on this topic.

1.2.2 Issues and Barriers

Considering the majority of architects agreed design reasoning and rationale were important topics that do not receive enough attention, the question was asked to what extent the architects themselves and organizations they were working for apply it. The consensus was very limited, both in quality and quantity. There were a number of reasons why these concepts are unknown or underused in the current industry, according to the architects. These are summarized in overarching themes below, in no particular order.

- **Project Constraints** – Under certain circumstances architects do not have the freedom to spend more time producing rationale for design decisions. Projects are cutthroat and rigid sometimes causing architects to have no time to spend on additional activities. These activities are not yet known in the current climate causing a lack of drive for these activities to be allotted time during a project.
- **Stakeholder Buy-In** – The lack of drive mentioned above is very much so related to the behaviour of stakeholders in the industry. If rationale production is not requested by the stakeholder, why spend more time capturing it? This is especially true when project resources are scarce. Stakeholders or management needs to be aware and convinced of these concepts and their benefits if the architect is to allocate time to this activity. This point is related to the client – advisor relationship between the stakeholder and the architect and the nature of that dynamic.
- **Lack of Industry Standards** – No clear standards exist on architecture documentation or tooling, let alone rationale documentation. Also, architecture faces the issue of having no clear formats, languages, documents, methodologies or notations. In the current industry climate, stakeholder needs in terms of presentation differ wildly on a project per project basis. Stakeholders are often young and unexperienced as the project domains evolve too soon. The quick rise of technological innovations cause even more disruption. The architects speculate the industry standardizing in the coming years, however the dynamic nature of architecture and IT as a whole is responsible for a significant portion of these barriers.
- **Domain Maturity** – The specifics mentioned previously are, among other details, all related to a lack of maturity in this domain. Most architects working today have had no formal education on the subject and learn as they go. Industry projects usually do not take architecture and its drive into account. The architecture domain is young and current stakeholders are usually not aware of its purpose and benefit. Architects face a career where project switches or cuts are too common and architects often switch careers themselves. There are no clear steps towards standardization, nor are there industry standards in terms of methods or tooling. All these smaller elements aggregate towards a distinct lack of maturity in this subject and highlights an overarching theme of domain immaturity contributing towards a slow progression.

1.2.3 Feasibility

The consensus among the architects is one of conservative enthusiasm. The architects do, in fact, recognize the potential benefits of design reasoning whether it be in terms of organizational and personal knowledge management, improved communication, improved collaboration, reflective thinking and architecture stability. However, most architects did show some signs of scepticism regarding feasibility of its effectiveness and usefulness.

In short, the main discussion point comes up when debating on how to approach the first step towards standardization in a method or instrument. Usability and effectiveness are key terms this instrument should have in order for successful adoption to take place. Therefore, ease of use and intuitiveness are important characteristics. On the one hand an instrument should be generic and abstract enough to be able to be applied in +- 95% of situations. If this is not the case, the instrument becomes too situation-dependent for widespread adoption. If the instrument is too specific it will not only hamstring the 'application-in-every-situation' characteristic it requires but prevents ease of use and intuitiveness as well.

On the other hands, the instrument cannot be too generic so that it becomes too evident and obvious for it to be useful in any regard whatsoever.

This thine line is an important opposition that needs to be tread in order to produce an effective instrument, according to the architects. In virtually all interviews the architects agreed that a sweet spot between these two ultimatums definitely does exist and therein lies the challenge of this project.

2 RESEARCH DESIGN THEORY

This appendix chapter offers all theory and literature covered and used whilst designing the experiment and research structure.

2.1 Literature Review

2.1.1 Introduction

As seen the previous paragraph, three of the four sub questions are of theoretical nature and need to be answered as such. Background knowledge is imperative since it forms the theoretical basis for the remainder of the experiment and resulting solution. This process requires extensive analysis and literature study and demands a consistent and effective process (Oates, 2009).

The main aim the literature study during this project is to gather knowledge relevant to the topic that supports the research project in gaining new knowledge. Similarly, literature review assembles knowledge that supports the claim this thesis' goals are worthwhile and realisable. Also, the gathered knowledge should support the thesis in that it does not merely repeat the work of peers and that gained knowledge was previously unknown on uncertain. It should also point to clear gaps in the existing knowledge base and clearly demonstrate how the knowledge found by the thesis fills that gap.

2.1.2 Approach

Relevant material will be retrieved from online databases such as Google, Google Scholar, Gartner, ACM Digital Library, DC library etcetera, by way of simple keyword queries. The utilization of multiple databases allows for a broader coverage, however it may result in duplicate studies which will have to be manually removed (Wohlin, Runeson & Höst, 2012).

The search approach is done by way of *snowballing*. The snowballing procedure means to follow references between papers to find other relevant papers (Runeson & Skoglund, 2009). Both backward and forward ad hoc snowballing will be employed. Backward snowballing follows the references in a specific paper and forward snowballing refers to viewing the papers that have cited the specific paper.

2.2 Interviews

2.2.1 Introduction

As is previously stated, in order to answer research questions RQ2 and RQ3 qualitative interviews are to be held. The main goal for these interviews are obtain detailed and intricate information that is too complex to find through regular means. Interviews can either be structured, semi-structured or unstructured (Oates, 2005) and offer good and detailed information albeit highly contextual and less generalizable.

Qualitative interviews demand more time and effort to execute. They are not suitable when high volumes of generic data is required. In the case of this project, however, expert knowledge is required. This is needed because in-depth and contextual information and knowledge that is not available through regular search engines is required.

In order to answer SQ2, which deals with the definition of the architecture process and design reasoning utilization among architecture practitioners, interviews are highly useful. To some extent, literature review can be used in order to answer what textual definitions of the architecture process exist. Literature review can also give an answer to what extent design reasoning is utilized in the industry. These answers, however, only provide superficial answers. Researching the same questions among architecture experts in Sogeti and comparing the results with literature answers might provide interesting new insights. This also yields a more contextualized and richer answer to the sub question and, ultimately, the main research question. On top of that, this also offers Sogeti business value when the architecture design process in the eyes of Sogeti employees is mapped and compared. Holding interviews with architecture experts might provide new insights and perspectives to the research project as a whole.

2.2.2 Interview Process

For the interviews, it is desired that respondents to go into more detail and have freer reign over their answers considering the explorative nature of the research. Therefore a semi structured approach will be chosen, allowing respondents to elaborate and think of their answers but still follow predetermined themes and issues. Interviews will be held in a 1-on-1 format and 3-6 architecture designers of different levels are to be interviewed depending on availability.

Considering the experience and seniority is an important aspect of the research, interviewees of different levels of architecture design will participate. If group interviews take place, there is a risk of higher ranking employees dominating the interview with their expertise. In this research, it is important that novice designers' data are equally noted.

Audio recording equipment will be present in order to capture everything that is said during the meeting, allowing the interviewer the concentrate on the interview process itself. The interview itself will take place in closed off rooms where the interviewee will be asked if recording the interview is allowed. If not, taking notes will serve as an alternative capture method.

2.2.3 Interview Analysis

Qualitative analysis will be performed through an inductive approach. This means that information observed through the interview data forms theories (Bhattacherjee, 2012). As opposed to the deductive approach, which holds preconceived theories that are to be tested during the interview. As the study is exploratory in nature, the inductive approach is more fitting. New ideas and theories may take shape, which is important for the relatively unknown nature of the material and artefact creation. Interviewees should be able to speak freely on the subject so that new perspectives or ideas concerning the design reasoning artefact may arise. This approach also takes full advantage of the expertise the interviewees have on the subject.

Full transcription is deemed not necessary and is not done due to time constraints. First, all data will be read through to gain a general impression of the overarching themes of the data. Generally, three major segments will be used to categorize data to gain a simplified view of the large volumes of data (Bhattacherjee, 2012). Elements of the data that are deemed irrelevant to the research is one, which will no longer be used for analysis. Elements of data that provide contextual or descriptive information is another category. The final category, which will form the data for the actual analysis, are made of elements that are directly relevant to the material.

The final category will be further split into categories, or units, that divide the data into relevant information segments. These categories evolve and change as qualitative analysis continues. New information and knowledge are prioritized based on frequency, importance and relevance of mentioned information (Oates, 2005). Once all information is divided into categories, refinement is necessary. Categories might merge or split depending on relevance and volume of data. Once the categories are refined, analysis for patterns and themes starts. For further transparency, a table will be made to show how often each category occurs or is mentioned and in which interview that took place. The process elaborated on is simplified in Figure 18. Qualitative analysis of textual data (Oates, 2005).

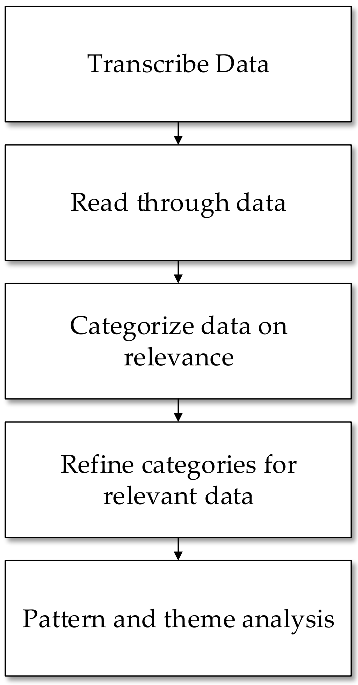


Figure 18. Qualitative analysis of textual data (Oates, 2005)

2.3 Artefact Design and Creation

2.3.1 Introduction

This paragraph concentrates on the actual development of the proposed approach as an IT artefact. March & Smith (1995) propose a research approach that suggest IT design- and natural science studies have 4 types of research outputs:

- **Constructs:** constructs form a conceptualization vocabulary of a specific domain. They constitute *concepts*, and specify terms by providing definitions.
- **Models:** a model is a set of statements that together form relationships among constructs. An entity-relationship diagram is an example of a model.
- **Methods:** a method is a set of consecutive steps to perform a task. A method might also be seen as guidelines or a principled approach.
- **Instantiations:** instantiations realize an IT artefact in its environment i.e. a working system that demonstrates constructs, models or methods in a system.

This approach that is to be designed and created will be an example of a method, for it guides the architect in how it should employ design reasoning. The proposed approach will provide guidelines on how design rationale should be made explicit. In this case, the IT artefact itself is the main contribution of knowledge and it should therefore be followed by another strategy to evaluate its effectiveness.

According to Oates (2005), the design and creation research strategy grows in popularity due to researchers being keen to investigate what happens when the new IT artefact is used in a real scenario. Therefore, a design and creation strategy synergizes well with the experiment strategy to understand and evaluate the IT artefact in use. Emphasis should not solely be placed on the evaluation of the artefact. When the artefact itself is the main contribution of knowledge, a design and creation strategy is necessary in order to have a systematic and scientific process for the development of said artefact.

2.3.2 Development

Vaishnavi & Kuechler (2004) propose a five-step problem solving approach for the design and creation process, based on guidelines set out by Hevner (2004) for design science research projects:

- **Awareness:** awareness represents the background of the problem, where and how it originated and further contextual information on how the problem is introduced.
- **Suggestion:** this phase serves as the creative part of the process as it involves thinking of tentative solutions to the problem.
- **Development:** development represents the phase where the ideas thought of in the suggestion phase are actually implemented.
- **Evaluation:** the evaluation phase analyses the implemented artefact and attempts to assess its effectiveness.
- **Conclusion:** the last phase draws up conclusions from the information and knowledge gained.

This process is not rigid but is meant to be interpreted as an iterative process that continually runs and serves as a scientific basis that supports the creative process. As shown before in Figure 3. Research Process Conceptual Framework A, the design and creation process runs parallel to the research process, continually evolving and changing based on input delivered by the research process. This process and its relation to the overall research process is highlighted in Figure 19. Research Process Conceptual Framework C.

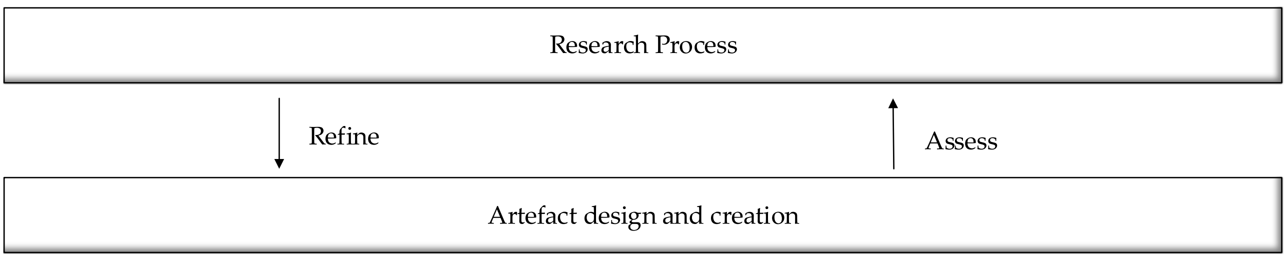


Figure 19. Research Process Conceptual Framework C

Oates (2005) states that for the actual development of the proposed method a development methodology is required. This development methodology differs from the research methodology in that it solely concentrates on the development of the method that is to be proposed as a solution in response to the problem. This difference is highlighted in Figure 20. Development Methodology.

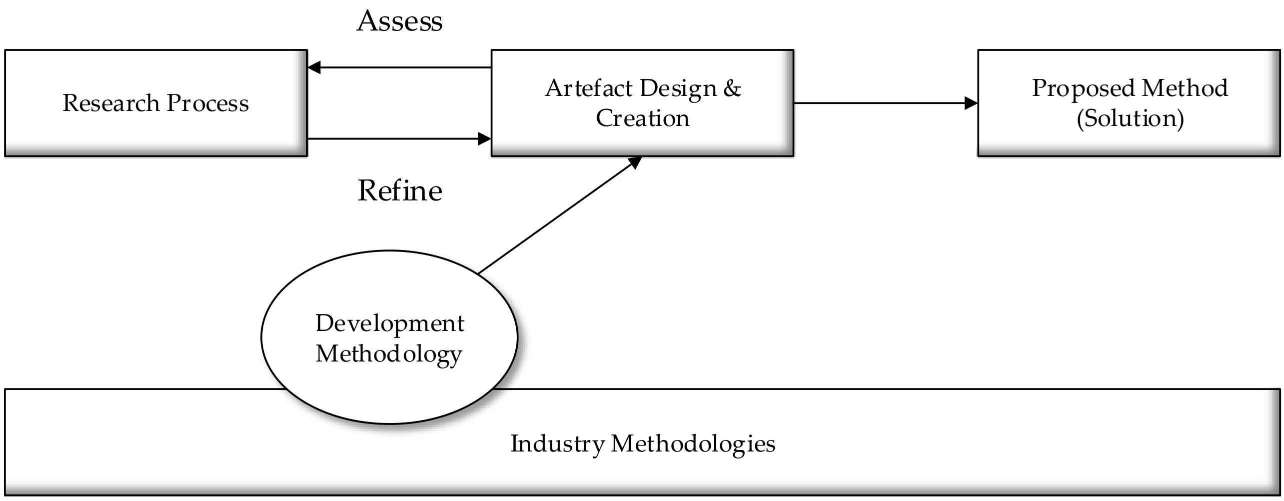


Figure 20. Development Methodology

As shown before, the research process refines the artefact by delivering input that is gained from literature review, interviews and empirical experiments. The artefact it also assessed by this same process. In order to have a systematic and scientific process to the design and creation phase a development methodology is needed. Some consistency and approach is required if an intricate method is to be engineered. The methodology also ensures an element of transparency to the design and creation phase. This development methodology is picked from the large base of existing industry methodologies. In this case, *method engineering* will be employed as a development methodology in order to create the method. Further elaboration as to how, why and what is shown in the next paragraph.

2.3.3 Method Engineering

Method engineering is a discipline that allows the user to design, construct and adapt methods for systems development (Brinkkemper, 1996). This definition fits the research strategy rather well since its intended use is to design and construct methods, which is exactly what's required. Method Engineering also emphasizes the use of existing methods and tuning them to specific situation. This supports the research project considering the proposed method or approach will be based on existing industry standards. This is called situational Method Engineering (Brinkkemper, 1996) and makes it a good fit to serve as development methodology for the design and creation of the artefact.

Similar to software engineering, method engineering is a discipline that involves all aspects of creating a system. However, method engineering focuses on creating methods instead of software systems, solidifying the fit for the research project. Method Engineering relies on meta-modelling techniques to design and evaluate methods. These techniques are based on UML industry standards, making them universally easy to interpret and use and compliant with other methodologies.
The concrete step-wise approach in order to do this is called the Method Association Approach (MAA) (Deneckere, R., Hug, C., Onderstal, J., Brinkkemper, S., 2015). The MAA is used to help engineers design methods for specific situations out of other methods. The consecutive steps are as follows (see Figure 21. Design Reasoning applied through the MAA method (Luinenburg et al., 2008).

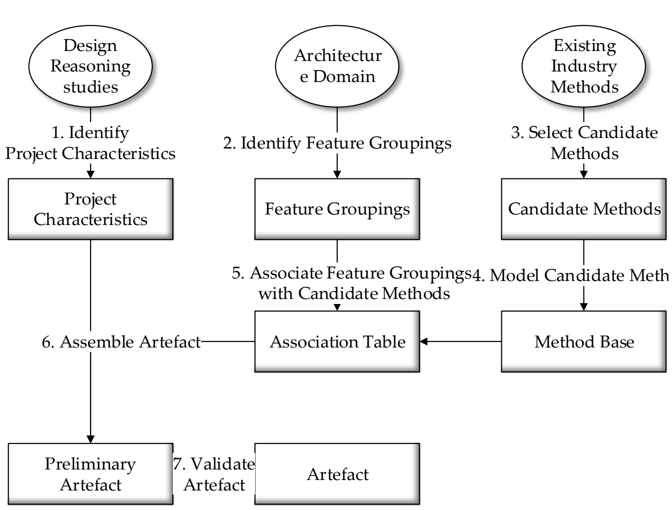


Figure 21. Design Reasoning applied through the MAA method (Luinenburg et al., 2008).

1. **Identify project situations:** the first step aims to characterize the project situation in order to gauge project needs and its scope.
2. **Identify feature groups:** in the second steps key feature groups have to be identified that serve as criteria for the selection of existing methods.
3. **Select candidate methods:** the third step actually selects candidate methods out of existing industry methods based on the criteria of step 2.
4. **Model candidate methods:** in step 4 a Process Deliverable Diagram (PDD) will be modelled. A PDD is a meta-modelling technique based on the UML standard that shows both the process- and deliverable perspective of a method.
5. **Assemble situational implementation method:** in step 5 the MAA association table is used to associate feature groups with a candidate method.
6. **Validate situational implementation method:** in step 6 the situational method that arose will be validated. In the case of this study, this validation is done by the previously mentioned pilot experiment of the artefact.

2.3.4 Artefact validation

The analysis of the effectiveness of the artefact is fairly straightforward as it happens iteratively. The artefact will be presented and evaluated by students and practitioners whom will participate in the experimentation phase. This will occur several times, allowing the participants identify points for improvements to be made to the artefact. The experiment phase will be elaborated on in the next paragraph. Also, the artefact will go through various back and forth feedback sessions with the University supervisor and Sogeti supervisor to incorporate further input. Lastly, the artefact will undergo a pilot test among MSc students from Utrecht University before the first experiment to identify flaws and critique points.

2.4 Empirical Experiments

2.4.1 Introduction

There are two different research paradigms, namely *exploratory* and *explanatory* research (Wohlin et al., 2002). Exploratory research concentrates studying concepts in their natural environment. Theories and findings surface from those observations. Explanatory research is concerned with identifying a cause and effect relationship. Here, a concrete theory is already defined and focuses on quantitative data. In the case of this study, an exploratory research design is chosen. The research is primarily informed by qualitative data and is flexible towards changing phenomena. This research approach is also known as the inductive approach as it attempts to discover findings without any predetermined factors. It is therefore not a fixed research design, but a flexible one.

The goal of an experiment is to investigate a cause and effect relationship (Oates, 2005). These relationships are formed by way of statements stating a positive or negative relationship. The goal is then to either prove or disprove that relationship. These research statements are defined as hypotheses, e.g. ‘factor A causes B’. All other factors that are not relevant to the hypothesis but might be present during the experiment are found, acknowledged, controlled and excluded. When only the relevant factors remain and observed data is significantly different a relationship is proven.

Experiments are, by definition, usually performed in a laboratory setting so that all other variables can be carefully controlled. In the case of information systems research, however, the social and contextual factors are quite relevant. Especially in the case of architecture design where discussion is prevalent. This causes a hit to true experimental design, which is why experiments of this nature are often called *quasi-* or *field experiments* (Wohlin et al., 2012). These types of experiments can never truly establish a cause and effect relationship due to uncontrolled variables possibly influencing results. For simplicity however, we will use the regular experiment term. The aim of the experiment is to only alter one factor and control all other variables. This one factor should then be the determining cause, if no significant variables all missed. The variable that is to be altered for a change in observation to take place is the independent variable. This variable is manipulated by the researcher. The variable that remains the same and demonstrates a change in observed data is called the dependent variable. This variable is observed by the researcher in order to determine an effect.

According to Wohlin et al. (2012), approaching the experiments in the form of case studies is recommended. A case study analyses the phenomenon in a single instantiation. This means research is done in a specific time and place in its own surroundings and context, which holds true for this experiment. Wohlin also mentions that case studies are very suitable for comparison of methods or the evaluation thereof, making it a good fit for this study. Case studies are easier to plan and are suitable for the evaluation of methods or tools in a specific situation. Because of their detail in a given context, results are harder to generalize to other situations.

Considering the reduced feasibility to randomize, the reduced scale of the experiment, the lack of resources for repeat experiments and the reduced ability to control all variables the experiments are technically analysed as case studies (Wohlin et al., 2012), in that they are seen as highly situational and contextual phenomena and are not easily generalizable. Since not all variables are completely controlled, this can lead to unpredicted findings. These findings, however, might shed light on new ideas or theories concerning the subject. Considering the exploratory nature of this research, the choice for a case study view is therefore quite fitting. Regarding the design and execution, however, an experimental approach adhering to Bhattacherjee (2012) is still utilized.

2.4.2 Experiment Design

The experiments are performed by way of static group comparison (Oates, 2005), through a Two-Group Experimental Design. This means that all participants are divided into two groups, a test- and control group. A test group receives the instrument that is theorized to have a determining effect (treatment) whilst the control group is not. The function of the control group is to form a base of data with which the observed data of test group is compared. In the case for this study, a post-test only control group design is chosen. This design is a simple version where the group of participants is split and tested once, see Figure 22. Post-test only control group design. This design omits the pre-test as participants are not available twice due to curriculum restraints.

R	X	O_1	(Test group)
R		O_2	(Control group)

Figure 22. Post-test only control group design

In the figure above, *R* represents a randomized assignment of the participants to either group. *X* represents the treatment which in this case is the utilization of the design reasoning approach. O_1 and O_2 represent the observations at the moment in time. This model explicitly shows two observations are performed concurrently for two randomly assigned groups. In order to determine the effect *E*, Bhattacherjee (2012) suggests the following formula:

$$E = O_1 - O_2$$

This formula simply measures the difference in post-test observed data between the test- and control group. In the case of the practitioners, however, another experimental design is chosen due to the reduced amount of participants. Adhering to the same notation, the research design is shown in Figure 23. Non-equivalent post-test only control group design.

N	X	O_1	(Test group)
N		O_2	(Control group)

Figure 23. Non-equivalent post-test only control group design

In this case, the symbol *R* for randomized assignment is replaced by the symbol *N* which represents a non-equivalent group design, i.e. a non-random assignment of participants. The reason for this is the low amount of professional practitioners from Sogeti that participate in the experiment and the widely varying experience levels. In order to obtain more significant results it is deemed that both the test- and control group should have equal levels of expertise concerning architecture design.

If this is not the case, and the test- or control group should possess significantly more experience, the observed differences cannot be attributed to the design reasoning approach. This problem does not present itself for the Enterprise Architecture students whom all have the same level of experience in architecture design. The formula still applies. This non-random design is defined as *quasi-experimental* (Bhattacherjee, 2012), which means that the experiment design lacks random assignment and is therefore less effective. The validity threats that arise due to this decision are addressed in paragraph 2.4.4, Validity.

The artefact elaborated on in 4.4 Artefact Design and Creation is the instrument or treatment (*X*) mentioned before and takes the form of a method or approach. In order for this instrument to be used by the participants, research material is needed. In this case, the participants are required to complete a case using the design reasoning approach. An approach itself does nothing without material to use it on. The essence of the experiment design involves having participants performing architecture design using the design reasoning approach versus participants whom have to do the same case but do not utilize the approach. The difference in observed performance is then analysed.

2.4.3 Experiment Process

The main aspect of experimental research that sets it apart from other research strategies is *treatment manipulation* (Bhattacherjee, 2012). The main goal of the research is to establish a cause and effect relationship between a design reasoning approach (cause) and consistently better design reasoning utilization in the design process (effect).

To put it differently, the cause is the independent variable and the effect is the dependent variable. The independent variable (cause) is altered by the researcher to observe what effect this has on the dependent variable (effect). The unique feature of treatment manipulation means to control the *cause* in this paradigm. In other words, the control group will be given the treatment (independent variable / cause) whilst design reasoning in the design process (dependent variable / effect) will be observed for remarkable differences. The specific variables are elaborated on in paragraph **Error! Reference ource not found. Error! Reference source not found.**.

Another key element to a successful experiment is randomized selection and assignment (Bhattacherjee, 2012). The key goal to this element is that each participant has an equal chance to receive the treatment. In other words, the participants are to be randomly split in test- and control groups in order to ensure the research groups are similar. This is necessary so that observed differences when the groups are compared can be attributed to the manipulation of the independent variable and not due to other influential aspects. The importance of this element is addressed in paragraph 2.4.4 Validity, as the aspect of randomized selection is closely related to the internal- and external validity of the study.

2.4.4 Validity

As mentioned before, the success of the experiment hinges on the ability to recognize and control influencing variables. If that is not the case, the observed changes in data cannot be solely attributed to the independent variable i.e. the design reasoning approach (Wohlin et al., 2012).

Internal validity is concerned with making sure the observed changes are in effect due to manipulation of the independent variable and not due to other uncontrolled influencing factors (Oates, 2005). Many threats to the internal validity of the research arise due to the difference between moments in time. Factors like maturation of participants due to boredom or experience or influencing factors arising in time. Students could, for instance, become better as the Enterprise Architecture course progresses. In this case, however, no pre-test is conducted and these types of threats to internal validity are not relevant. According to Bhattacherjee (2012), the post-test only control group design controls for maturation, testing, regression, selection and pre-test – post-test interaction. Instrument validity is another example of internal validity, which is concerned with validating the instrument used to measure the observed effect. This element, in fact, is relevant for this study. Therefore, the design and creation of the case is closely supervised by the research supervisors before the experiment starts.

Construct validity is concerned with validating our concepts and questions and whether or not they measure what is intended to measure. If a concept is not adequately defined or validated, definitions might vary due to differing interpretations (Bhattacherjee, 2012). In the case of this study, however, no new concepts are introduced and existing concepts are closely defined in line with existing related research

With regards to *external validity,* the results have to be generalizable to a broader population (Wohlin et al., 2012). The external validity of this study is not high due to several concerns. First, the experiment can only be conducted once due to limited availability of students and professionals. Only when an experiment can be conducted multiple, or many, times can generalizability be guaranteed (Oates, 2005). Also, the participants are not representative of a broader population. The professionals are handpicked by the host organization Sogeti whilst students enrol in the Enterprise Architecture course themselves. The research has thus no control over the selection of participants and is limited in the amount of people that can participate. The situation of the experiment is very specific and contextual and is therefore not suited to externalize its results. This is not a big issue, however, due to the exploratory nature and goal of this research. Generalizability is not its main aim.

Reliability is the extent in which the study would yield similar results if conducted multiple times. This is difficult to assess, however, since the experiment is conducted once in this study. If the experiment were to be conducted multiple times the participants would be more experienced and that could influence results.
If the participants were replaced, the difference in skill and expertise could introduce bias. Also, the case that participants have to solve has to be completely redesigned to prevent maturity. However, in such a complex case, it is difficult to replicate another case with similar difficulty. The differences in these cases can, again, introduce bias and influence the results. Striving for a high reliability is not in line with the exploratory nature of this study.

2.4.5 Quantitative analysis

The treatment effect is measured simply as the difference in the post-test scores between the two groups: *E = (O1 – O2)*. These patterns are defined in the literature study, in order to compare the means of two independent groups. According to Bhattacherjee (2012), the appropriate statistical analysis for a post-test only control group design is a student's t-test. Wohlin et al., (2012) agrees by suggesting an independent samples t-test to compare two groups but also recommends the inclusion of a Mann-Whitney test when parametric assumptions are not met.

The Mann-Whitney test is a non-parametric alternative to the t-test and is best suited when some statistical assumptions made by the parametric equivalent are uncertain. The t-test is two tailed and thus, non-directional due to the uncertainty of the direction of the results. The goal of the t-test is to statistically demonstrate a significant difference between group means. Therefore a significance of < 0.05 is sought. The statistical assumptions of a parametric t-test are as follows:

1. **Categorical independent variable:** the independent variable has to be of categorical nature.
2. **Continuous dependent variable:** the dependent variable has to be a continuous variable.
3. **Independent observation:** observations of both groups have to be unrelated.
4. **Normal distribution:** the dependent variable is normally distributed within each group. To test this, the Shapiro-Wilk test will be conducted. To be successful, the test result has to be insignificant (> 0.05).
5. **Equal variances:** Levene's test can be used to assess if the groups have homogeneous variances. To be successful, the rest result has to be insignificant (> 0.05).

2.4.6 Qualitative analysis

Besides quantitative analysis, which is mostly concerned with statistical analysis and large volumes of data, qualitative analysis will be performed concurrently. Qualitative analysis is context-driven and is largely dependent on analytical skills to understand a phenomenon instead of predicting or proving it (Bhattacherjee, 2012). Various methods to perform qualitative analysis exist. Grounded theory, for example, is an inductive approach to forming theories out of existing data (Oates, 2005). The theory is then 'grounded' in the data. This method is not applicable, however, considering this study has a deductive research approach i.e. the theories are already defined and evaluated with gathered data.

Content analysis refers to the systematic analysis of the content of textual data (Bhattacherjee, 2012), which is applicable to the needs of this study. Content analysis is deductive, as opposed to the inductive approach of the grounded theory. A type of content analysis is sentiment analysis, which refers to the capture of sentiment i.e. the opinion or attitude towards something. Bhattacherjee (2012) mentions that content analysis, albeit fitting for gaining highly detailed and specific information on textual data, lacks a consistent and structured approach to analysing said data.

Schilling (2006) does offer a structured approach. Schilling suggests a systematic and rule-based approach to the analysis of textual data. This approach consists of 5 consecutive steps, see Figure 24. Structured qualitative analysis approach (Schilling, 2006).

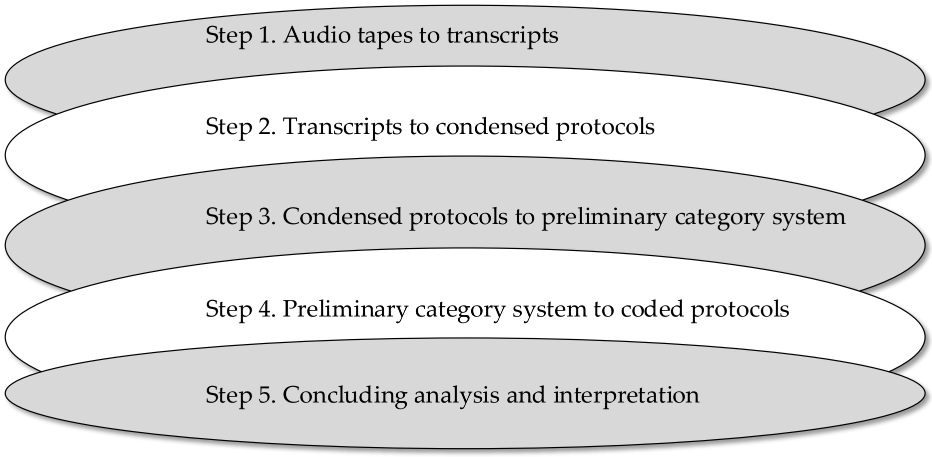


Figure 24. Structured qualitative analysis approach (Schilling, 2006)

Step 1. The first step is concerned with generating transcripts from audio tapes. In order for text analysis to be performed, audio data has to be made explicit. Even though this step sounds trivial, some rules have to be defined to ensure a systematic process (Schilling, 2006). During this study, transcription will adhere to a relevant-only text transcription. This means that slips of the tongue, stutters, coughing or drumming of fingers will not be transcribed. Only discussion that is relevant to the material at hand will be transcribed in order to further decrease transcription time. For example, phrases that are clearly not valuable to the research like: ''It is cold in here'' will not be transcribed. In the case of this research however, audio recording and transcription is not necessary as the architects provide the documentation themselves.

Step 2. The second step guides the transformation from transcript data to condensed protocols. These protocols contain descriptions of the situation of the text, so to have a clear context of the following data. The analysis will also be given a clear direction, which specifies the piece of data further. The goal of this step is to reduce the noise of raw data, also known as paraphrasing (Schilling, 2006). Paraphrasing is done by deleting all words that are deemed unnecessary or irrelevant. This is mostly done through the judgment of the researcher, but a clear strategy has to be defined in order to separate units of analysis. The unit of analysis for this study will be a half-sentence. A half-sentence represents a component of the full sentence that does not lose meaning, as opposed to having a single-word unit of analysis. The third option is a full-sentence paraphrasing strategy, which guarantees no loss of meaning or information can take place. This study opts against this strategy due to resource constraints.

Step 3. The third step guides the process of condensed protocols to a preliminary category system. This means that previously uncovered units will be categorized in categories that are defined by the researcher. These categories can be predetermined or defined according to the field data, where overarching and recurring elements form the categories. This study opts for the latter approach considering the relatively unknown direction the field data may take.

Step 4. The fourth step is concerned with generating coded protocols from the preliminary category system. Basically, generated units of analysis are assigned to categories. These categories can already be predetermined or surface inductively as analysis progresses. In order to code units to categories, a coding scheme will be defined. This scheme will determine definitions, terms and rules for assigning codes with examples to further clarify the categories.

Step 5. The fifth step handles the analysis and interpretation of the data and the eventual conclusion thereof. This step is mostly defined as the analysis progresses, however some key criteria are a given. Among these criteria are: the frequency of occurrence, representativeness of mention, weight and vividness. For example, the unit of analysis may mention something clearly explicit or implicitly refers to it.

3 LITERATURE REVIEW

This appendix chapter offers all theory and literature covered and used whilst executing the thesis project.

3.1 Architecture

This paragraph covers architecture as a concept and distinguishes the different types of architecture. Its main goal is to grasp the concept and impact of architecture and to lay a knowledge foundation with which we can relate design reasoning. First, a definition is sought, including different definitions for the various types of architecture. The different types of architecture are analysed and compared. The last paragraph deals with existing industry standards and is meant to demonstrate the lack of design reasoning support to confirm the problem statement.

3.1.1 Definition

Architecture has many definitions, including those specifically for enterprise- or software architecture. In the case of this study, however, a more general definition of architecture will be used. This definition does not solely focus on enterprise- or solution architecture, but tackles architecture as a whole. The definition, as stated by ISO / IEC / IEEE FDIS 42010:2011 is: *the fundamental concepts or properties of a system in its environment, embodied in its elements, relationships, and in the principles of its design and evolution* (ISO / IEC / IEEE, 2011). In short, this definition can be interpreted as a holistic perspective of a system in its context.

Some specific definitions exist regarding enterprise- or software architecture, e.g.: *a set of design artefacts, or descriptive representations that are relevant for describing an object that can be produced to requirements as well as maintained over the period of its useful life* by Zachman (1987) or *the software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both* (Bass, L., Clements, P., Kazman, R., 2003). A more general definition, used by ISO / IEC / IEEE (2011), is deemed more fitting considering the research does not necessarily apply to just enterprise- or software architecture. The main topic is architecture as a whole.

The main goal of architecture is to manage complexity of a system, software or otherwise, that is growing in intricacy and complexity (Lankhorst, 2009). As you need an architecture for designing a house, architecture is needed to maintain a(n) (enterprise) system. For example, details on the amount of windows, size of those windows, building materials, colours and how they are structured together is necessary in order to understand and interpret the house. Similarly, an overview of an organization, its business processes, their technical infrastructure, the people, the information streams, application landscape and their interdependencies need to be defined. With regards to software, interrelated software elements and functions have to be defined (Bass et al., 2003).

A notable difference between building and systems architecture, however, lies in the history. Building architecture goes back to thousands of years of practice, expertise and experience. Building architecture has a common framework in that dimensions, terms and languages have been standardized for millennia. Systems architecture still lacks that common language, and the shared culture of reference frames is dynamically changing. For example, every building architecture contains dimensions in the metric system which is understood by virtually every architect. Systems architecture, however, has widely varying languages and modelling techniques. For instance, a description language can be textual or graphical and informal or formal. Different domains utilize different methodologies, methods or tools. These tools might only support specific languages, and not interpret others. This intricacy, stemming from the young and dynamic nature of systems architecture, demands studies and projects that work towards standardization in methods and approaches.

3.1.2 Architecture Process

Architecture is not to be seen as a product which is delivered and finished. Architecture is a process. An important player in the architecture process is the stakeholder (Lankhorst, 2009). A stakeholder is, according to the ISO / IEC / IEEE standard (2011), a role that has some stake in concerns to a system. For example, a business manager probably has no stake in how the architecture is designed. He does, however, have a stake in the impact changes have on his or her (business) concerns. An architect has to be aware of these concerns, ranging from various different perspectives and stakeholders. A Chief Information Officer (CIO) probably views a Human Resource Management (HRM) system differently than a Chief Operations Officer (COO) does. This creates the need for a holistic perspective of a system that needs to be presented toward roles with various backgrounds. This process is simplified in Figure 25. Architecture process (Lankhorst, 2009).

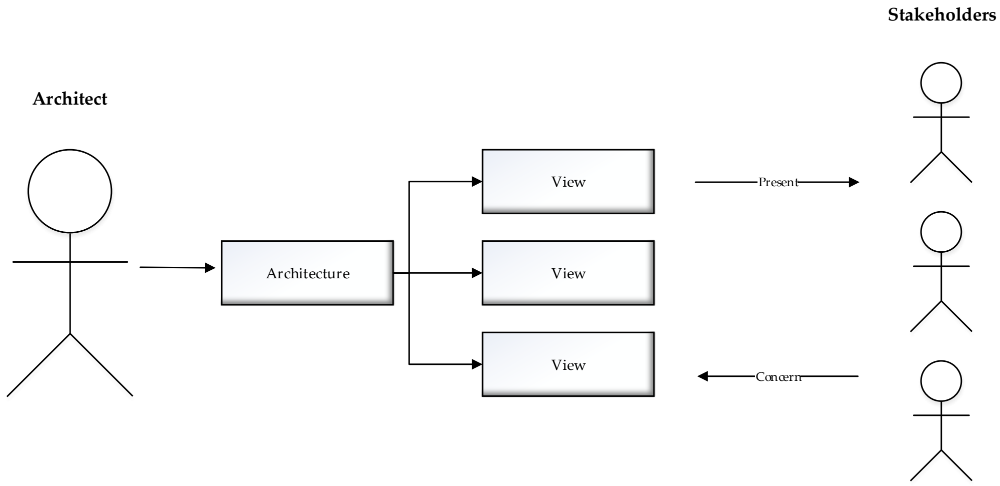


Figure 25. Architecture process (Lankhorst, 2009)

This process is in line with software architecture peers, who view stakeholders as the people for whom the systems are built (Rozanski & Woods, 2011). Views are pieces of architectural knowledge that address stakeholders concerns. They are separated on the nature of those concerns. The process above only encapsulates the high over process of communicating an architecture and to whom, however. In order to fully grasp systems architecture, the process of designing an architecture is very important.

3.1.3 Architecture Design

As shown in the previous figure, architectures are designed around views. Views are essentially snapshots of an architecture centres around a certain domain that addresses a stakeholder concern (Lankhorst, 2009). Views are necessary to deal with the architectural complexity and intricacy. Stakeholders have no stake in elements of the architecture that are not their areas of management or expertise. Thus, architectures do not need to be presented in full detail to each stakeholder. If a stakeholder is concerned with how the new system has to be supported through servers, the technical or hardware infrastructure view is relevant. There is no need to concern the stakeholder with a view of the architecture of human capital, as this is irrelevant for his concerns.

Considering these views are always necessary, standardized viewpoints have been predefined so as to design an architecture around. These viewpoints have standardized notations, definitions and agreed upon structures. Some examples of standardized viewpoints are the process view, which deals with how consecutive steps are executed, the application view, which handles which software applications support which parts of the business process and the technical infrastructure, which details how these software applications are physically supported and ran. Some frameworks that guides the development of these viewpoints are the Zachman Framework and Kruchten 4+1 View Model, both of which are tackled in 3.3, Industry Standards. This distinction can be seen in the IEEE Standard Framework, where the viewpoint and view relationship is shown (see Figure 26. IEEE STD 1471 highlighted (Hilliard, 2007). This industry standard is further elaborated on in a coming paragraph.

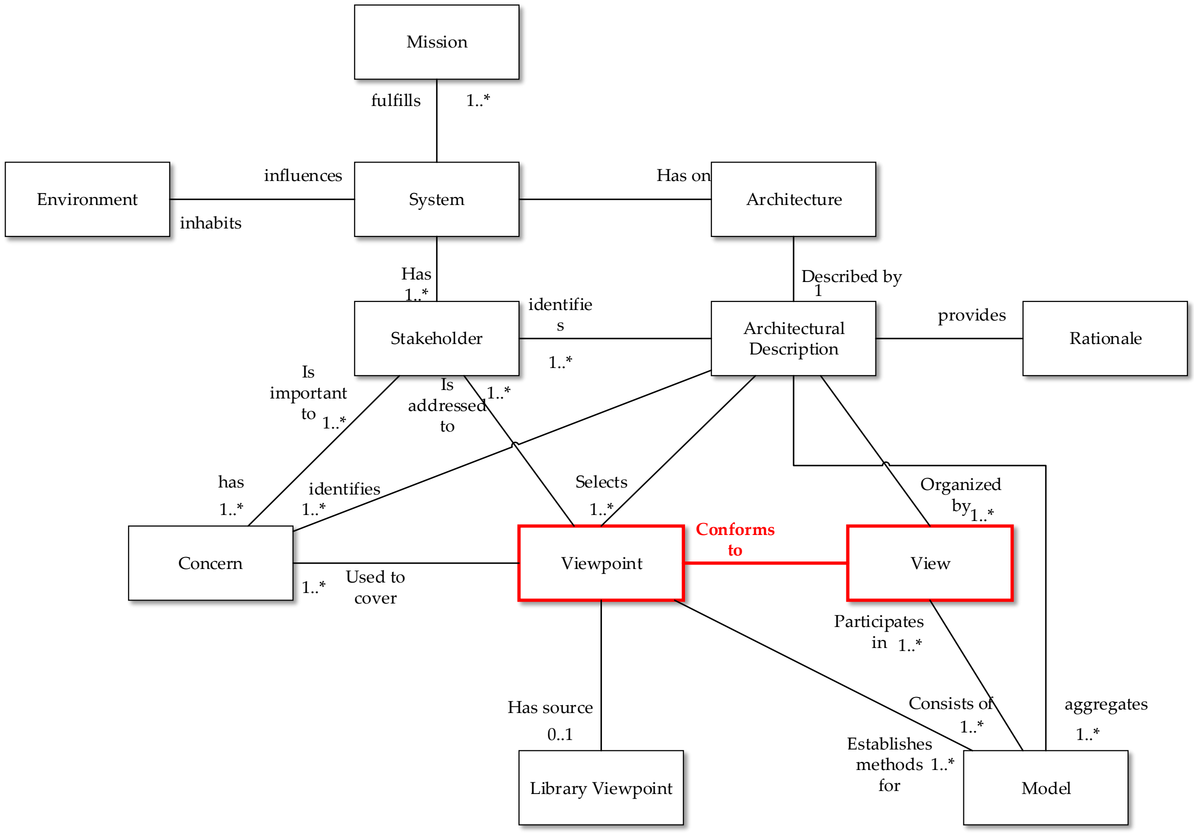


Figure 26. IEEE STD 1471 highlighted (Hilliard, 2007)

These industry frameworks offer architects guidance on how to approach designing an architecture. Some of the more well-known architecture development frameworks are identified in paragraph 3.3, including an analysis on their relationship to design reasoning support. The careful selection of correct views, along with the use of an architecture development framework forms the basis of architecture design. This design is still in theory, however, and is yet to be made explicit.

When the necessary views and stakeholder concerns are known architects have to start designing the architecture. There are various notations and languages in order to do so, depending on the type of architecture that is to be designed. ArchiMate is a well-known example of a modelling language that allows architects to visualize their enterprise architecture design (Lankhorst, 2009) whilst the Unified Modelling Language (UML) is an industry standard when modelling the architecture of software systems. These languages exist to support architects in the visualization and documentation of architectures, so that these designs may be clearly communicated to stakeholders and peers. The trick lies in the wide variety of available techniques, notations and languages. There is no one standard for systems architecture as each type of architecture utilizes their own supporting techniques and tools. In fact, each view can have a different technique and notation. The Business Process Modelling Notation (BPMN), for example, only supports the modelling of business processes. The process view is just a single layer in, for instance, the ArchiMate framework or Kruchten 4+1 View Model. A quality design is correctly distinguished into views that are based on stakeholder concerns, supported by a widely known language in order to visualize the design.

Another aspect, that is exclusive to software architecture, is the idea of concrete functional- and non-functional requirements. A stakeholder might, for example, say the ability to automatically send emails is a key feature to the to-be designed software system. The architecture has to be designed around this requirement, as it will represent a functional element of the system (Rozanski & Woods, 2011). Non-functional requirements, however, do not represent a key feature that is to be implemented. Non-functional requirements represent aspects of the system that cannot be addressed by implementing a software function. For example, the security, reliability and stability of the system can be requirements which have to be taken into account when designing the architecture. These cannot simply be addressed by introducing another element, however. Perhaps the non-functional requirement of stability can be achieved by introducing *less* elements, for example. Where enterprise architecture is concerned with a holistic perspective of the organization, including its structure, elements and relationships on each level, the software architecture handles the implementation of all key requirements whilst adhering to abstract constraints and limits. Architecture design has to account for all these concerns, whilst being aware of constraints, risks and impact on other layers of the system.

3.2 Types of Architecture

As mentioned before, there are various types of architecture. In order to grasp the vast topic of systems architecture, the most widely known types of systems architecture are studied. These are *enterprise architecture, solution architecture, software architecture* and *hardware architecture.* In these paragraphs, each individual domain is studied and differences are discussed. These types of architecture are similar in many respects and on the surface have subtle differences. Yet when applied and studied, notable differences emerge.

3.2.1 Enterprise Architecture

The most holistic architecture approach, enterprise architecture, is concerned with applying a broader scope than just technical or IT domains (Lankhorst, 2009). Enterprise architecture basically encompasses architecture at the organisation level. This entails not only hardware or IT but also refers to people, software applications, hardware infrastructure, information streams, business processes etc. The main goal of enterprise architecture is to capture the essential elements of the core business and all elements that support it, including IT and hardware.

All across the organisation, local domains have individual architectures. IT is a good example, as is the application architecture. Locally, these architectures may be sound. Problems may surface, however, when changes occur that impact multiple domains and, thus, architectures. For example: a heavy technical infrastructure, with lots of servers, may offer solid performance against a low cost. This infrastructure, however, may not be optimal when many business processes are of flexible nature and run of cloud-based applications. Enterprise architecture provides the insight required to take varying requirements into account (Lankhorst, 2009). A good enterprise architect recognizes concerns and requirements from different unrelated domains and maps the potential impacts and considers the trade-offs and risks.

In short, enterprise architecture can be seen as a more general perspective to systems architecture. An approach that provides specific principles, methods and tools in order to design an architecture that encompasses the entire organisation. This differs from more solution oriented architectures, which are more narrowly focused on a specific domain.

3.2.2 Solution Architecture

Solution architecture has, even though related to, notable differences from enterprise architecture. Greefhorst & Proper (2013) define solution architecture as: 'a*n architecture of a solution, where a solution is a system that offers a coherent set of functionalities to its environment. As such, it concerns those properties of a solution that are necessary and sufficient to meet its essential requirements.'* In other words, solution architecture is a more specific systems architecture. The solution architecture is only concerned with specific elements that are relevant to the solution. A new business process might not have impact on existing hardware infrastructure, for example. A solution architecture might only concern a new application and thus only have an architecture design on the organizational layers that are directly or indirectly impacted.

This distinction is supported by Gartner's IT glossary (2013) on their website which mentions that a solution architecture combines various enterprise architecture viewpoints for a specific solution as opposed to a complete organization. The Open Group (2011) refer to solution architecture as a *description of a discrete and focused business operation or activity and how IS/IT supports that operation. A Solution Architecture typically applies to a single project or project release, assisting in the translation of requirements into a solution vision, high-level business and/or IT system specifications, and a portfolio of implementation tasks.* This definition confirms the sentiment of solution architecture being a specific architecture release.

3.2.3 Software Architecture

Bass et al. (2003) refer to software architecture as *the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both*. This definition refers to both static structures and dynamic structures as distinct software pillars. Static structures are self-contained pieces of code that are fixed (static) and are designed elements that are programmed to have some sort of function (Rozanski & Woods, 2011). Dynamic structures define software elements that describe how the system actually performs when executed. These elements are not by design, but provide details on how it reacts to external stimuli.

Software architecture, as opposed to enterprise architecture, has a much narrower and detailed scope. It is concerned with the design of a piece of software, whereas enterprise architecture deals with a complete organization. Software architecture goes into more detail, however, also envisioning every aspect of interacting software elements and how they must behave (Rozanski & Woods, 2011). A software architecture has concrete functional requirements that are deemed necessary by the stakeholder. These functional requirements have to be implemented, as opposed to enterprise architecture, where a holistic perspective regarding strategy objectives and business – IT alignment are more important. This difference in goal and scope is significant when comparing the two, however both types of architecture design still feature architecture designs. This means that they need to be designed, represent a system and exist to address stakeholder concerns.

The difference between solution architecture, however, is more subtle. Solution architecture is not necessarily bound to a technical dimension (Greefhorst & Proper, 2013). A solution might cross boundaries into multiple domains, whereas software architecture practitioners are only concerned with the internal elements of a piece of software. For example, an application system might be a solution accompanied by a change in hardware infrastructure and corresponding business process to execute it. A software architecture does not handle such contexts as they are out of its scope.

3.2.4 Hardware Architecture

Rai & Kang (2008) refer to hardware architecture as the design of a system's physical components and how they relate. The comparison of enterprise architecture and hardware architecture is similar to the comparison with software architecture. Where enterprise architecture enjoys a holistic perspective, hardware architecture is concerned with the design of the physical hardware infrastructure. The similarity in the comparisons stem from the process of a broad to narrow perspective. Software architecture, similar to hardware architecture, is one specific layer of enterprise architecture.

3.3 Architecture Frameworks

This paragraph aims to grasp what industry methodologies and approaches exist to support practitioners whilst also confirming the lack of design reasoning support in industry standards. 8 frameworks and methodologies have been picked, explained and analysed. The selection criteria of these frameworks are frequency of mention, reference count and industry renown. The selection is in line with widely referenced books and papers on the topic.

3.3.1 Zachman Framework (ZF)

The Zachman Framework is an Enterprise Architecture framework which serves as a fundamental structure for viewing an enterprise system from a number of different perspectives (Zachman, 1987). Its key goal is to provide a two-dimensional schema that guides the organization of an enterprise system. The framework offers a comprehensive list on which elements to address when defining an enterprise system. View Figure 13. Zachman Framework (Sowa & Zachman, 1992).

As can be seen in the figure, the model is represented in two dimensions. The rows represent a certain perspective i.e. how a specific role would view an element of the system. The columns represent primitive questions in order to specify the system artefact. Each crossing of a row and column produces an artefact. This artefact should address a key element in the system on a particular issue (how, what, who).

	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
Objective/scope (Planner)	List of things important in the business	List of business processes	List of business locations	List of important organizations	List of events	List of goals & strategies
Enterprise Model (Owner)	Object Model	Business Process Model	Business Logistics System	Work Flow model	Master Schedule	Business Plan
System Model (Designer)	Logical Data Model	System Architecture Model	Distributed Systems Architecture	Human Interface Architecture	Processing Structure	Business Rule Model
Technology Model (Builder)	Physical data / Class Model	Technology Design Model	Technology Architecture	Presentation Architecture	Control Structure	Rule Design
Detailed Representation (Programmer)	Data Definition	Program	Network Architecture	Security Architecture	Timing Definition	Rule Speculation
Functioning Enterprise (User)	Usable Data	Working function	Usable Network	Functioning Organization	Implemented Schedule	Working Strategy

Figure 27. Zachman Framework (Sowa & Zachman, 1992)

The *why* in the Zachman framework represents the motivation or reasoning for a specific element of the architecture. For example, a *planner* would want business goals or a strategy in order to define the motivation and reasoning behind designing a system. This sort of motivation differs from design rationale, however, considering it is not a justification for why the design decision itself has been made but rather a generic description on what the motivation for the context of the entire system is. The Zachman Framework does not inherently provide instructions on what, why and how to do so.

3.3.2 The Open Group Architecture Framework (TOGAF)

The Open Group Architecture Framework (TOGAF) is a high-level framework and method that provides an approach to the development and management of an architecture (Lankhorst, 2009). Its key goal is to provide a method by which enterprise architects can produce and maintain a system architecture.

TOGAF consists of four main components:

- **The Architecture Capability Framework:** the capability framework addresses the required capabilities (processes, skills, roles etc.) in order to implement and maintain a system architecture.
- **The Architecture Development Method (ADM):** ADM provides a concrete, iterative method and approach on how to develop the system architecture. ADM is considered to be the core of TOGAF.
- **The Architecture Content Framework:** the content framework consists of four interrelated specialization domains. These are the business-, application-, data- and technical architecture. These domains together form the overall enterprise architecture composition.
- **The Enterprise Continuum:** the enterprise continuum consists of the architecture- and solution continuum and provides a way of classifying system architectures and illustrate how architecture are developed through various reference models.

The Architecture Development Method (ADM) is the most well-known element of TOGAF and provides architects with a stepwise approach how to develop an architecture. See Figure 28. TOGAF ADM version 9.1 (The Open Group, 2011).

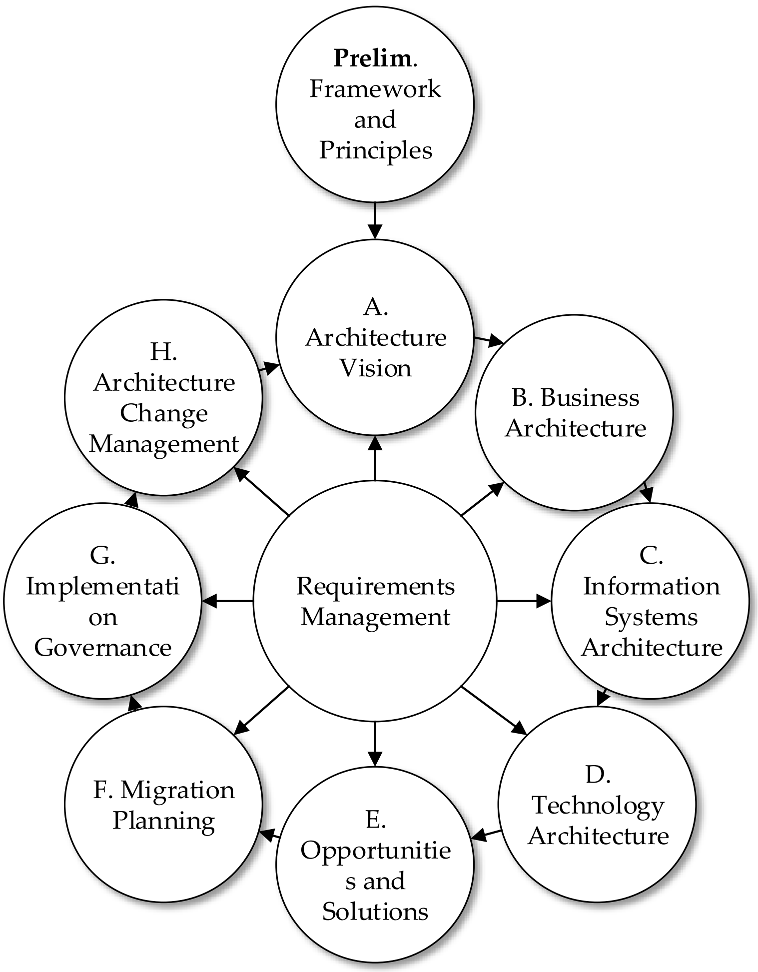


Figure 28. TOGAF ADM version 9.1 (The Open Group, 2011)

TOGAF provides the architect with comprehensive process from start to finish. However, TOGAF does not provide guidance on deciding how broad the scope should be or how specific the level of detail should be (Lankhorst, 2009). TOGAF is intended as a generic method, to be used in as many situations as possible. According to the architecture principles book published by The Open Group, the documentation of design rationale is recommended. However, TOGAF itself does not provide clear instructions on how to do so.

3.3.3 IEEE Std 1471 Conceptual Framework

The IEEE standard 1471 is an IEEE best practice for the architectural description of system architectures (Hilliard, 2007). It focuses mostly on software intensive systems, and provides a theoretical base for the definition, analysis and description of system architectures (Lankhorst, 2009). Its main goal is to provide a standard in terms of concepts, definitions and their relationships. See Figure 29. IEEE STD 1471 (Hilliard, 2007).

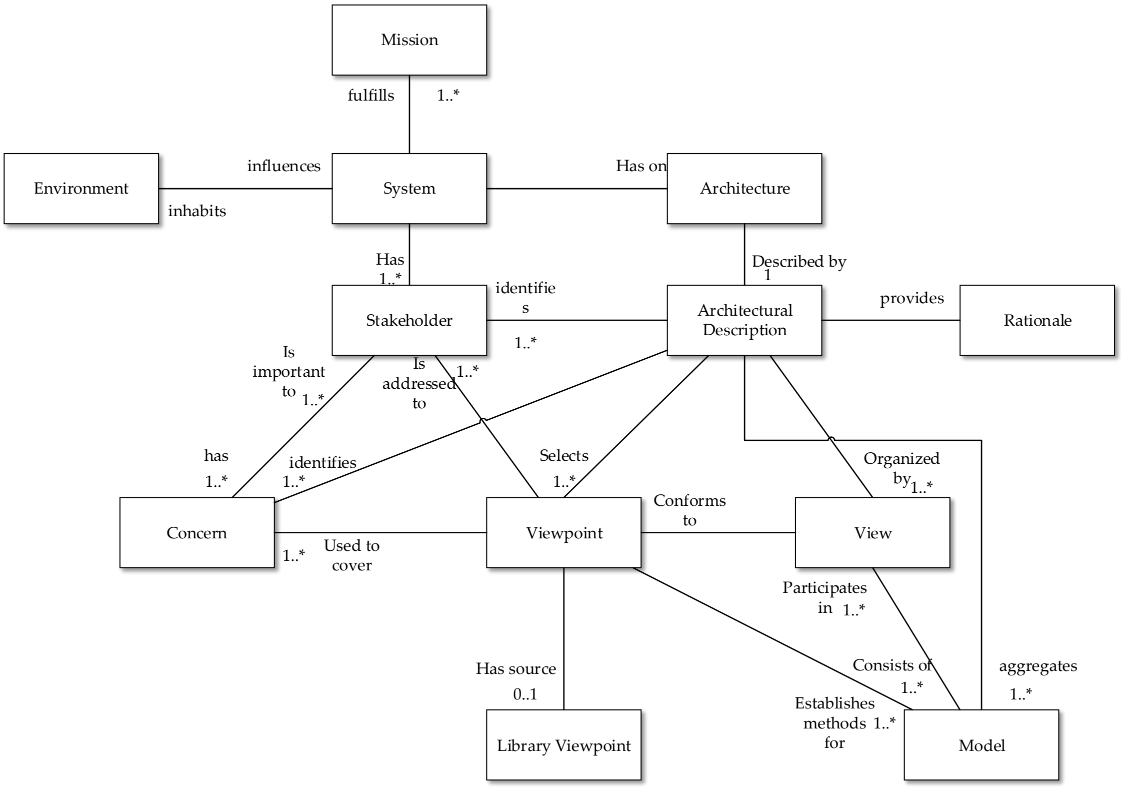


Figure 29. IEEE STD 1471 (Hilliard, 2007)

The conceptual model gives a clear idea which key concepts need to be present for describing a system and how they relate. The model does not, however, provide instructions on exactly how to do so nor does it suggest specific methods or approaches.

The 1471 model does explicitly mention a rationale is to be provided when defining the architectural description, however it does not provide a means or offer guidance on how to go about doing so.

3.3.4 Kruchten 4+1 View Model

The 4+1 model is an industry standard for describing the architecture based on multiple views (Kruchten, 1995). The model is used to describe the architecture of a system from the perspectives of multiple stakeholders. It is primarily used for the design of software systems, but can be used to represent any kind of architecture. The viewpoints are as follows (see Figure 30. Kruchten's 4+1 View Model (Kruchten, 1995):

- **Logical view:** the logical view is concerned with the requirements of the system and represents the functionality it should provide.
- **Development view:** the development view views the system from the perspective of a developer (programmer) and is concerned with the execution, implementation and management of the system.
- **Physical view:** the physical view represents the system from an engineer's perspective. It is concerned with software components and hardware elements and how they physically connect.
- **Process view:** the process view focuses on the dynamic aspects of the system and elaborates on how the system elements communicate and run. Non-functional requirements (scalability, stability, concurrency, performance) are addressed here.

- **Scenarios (use-case view):** the system is illustrated by a set of scenarios, or use-cases, that detail how exactly the system will run in a given situation.

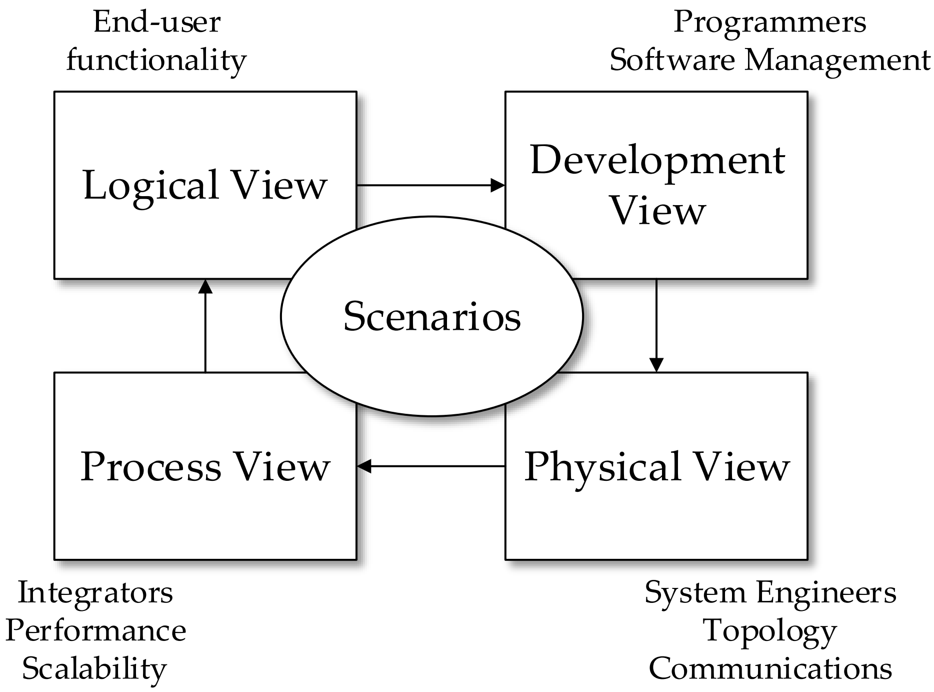


Figure 30. Kruchten's 4+1 View Model (Kruchten, 1995)

The 4+1 View Model provides a recurring, stepwise approach on how to design a system from multiple different perspectives. Although the accompanying documentation of the 4+1 View Model (Kruchten, 1995) recommends the capture of design rationale the model itself does not explicitly tell you do so nor does it give clear instructions on how to approach it.

3.3.5 Model Driven Architecture (MDA)

The Model Driven Architecture (MDA) is an approach to the design and development of software systems (Soley, 2000). It is developed by the Object Management Group (OMG) in 2000 and aims to provide guidelines on the creation of system designs and –models. Its main goal is to create computer-generated applications and software components from high level abstract models. This is done in three stages (see Figure 31. MDA Framework):

- **Computation-Independent Model (CIM):** in this stage the requirements are defined and a business model for the context of the system is created.
- **Platform-Independent Model (PIM):** in this stage the operation workflow of the system is created with little detail so as to illustrate platform independency
- **Platform-Specific Model (PSM):** in this stage application-specific technology is modelled and linked to the platform specifics it intends to use.

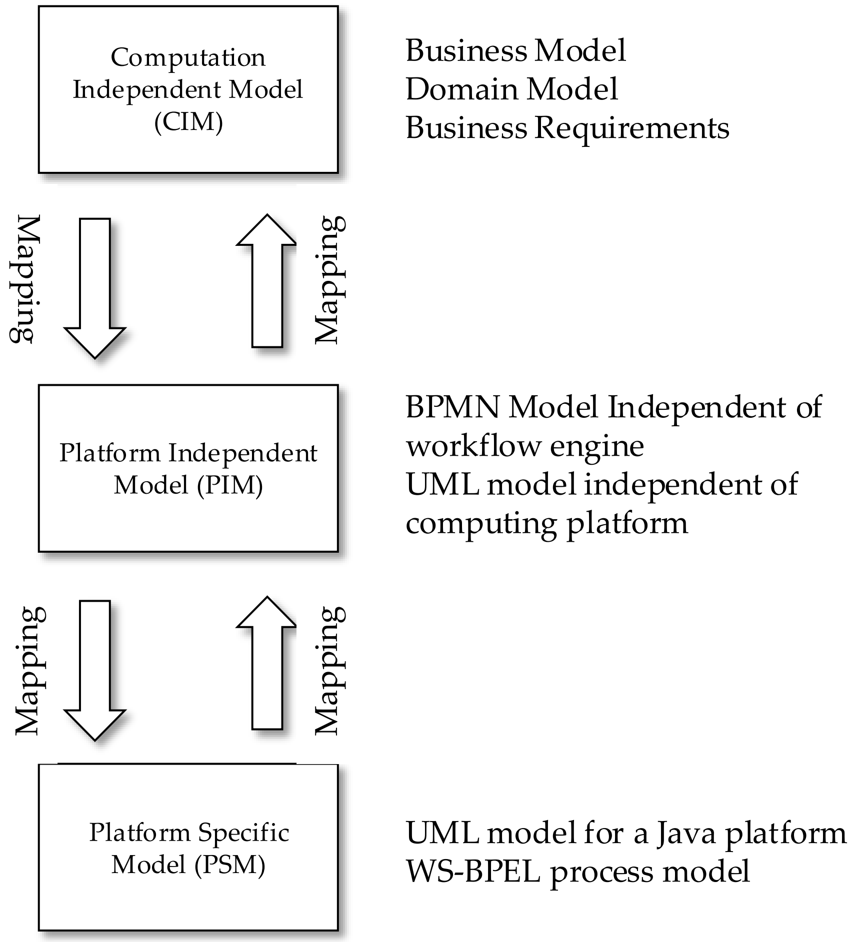


Figure 31. MDA Framework (Lankhorst, 2009)

The MDA uses the Unified Modelling Language (UML) as the modelling language of both PIMs and PSMs, which is an industry standard among notations. According to Lankhorst (2009), the MDA framework is ever so relevant for EA as it is for software architecture due to new languages being developed within the MDA framework. As MDA concentrates on automation and computer model generation there is no mention of capturing design rationale.

3.3.6 Architecture of Integrated Information Systems (ARIS)

ARIS (Architecture of Integrated Information Systems) is a well-known approach to enterprise modelling (Lankhorst, 2009) and aims to offer a holistic approach to the analysis of system processes (Scheer, 1998). ARIS provides a high-level framework and is supported by its business process modelling (BPM) tool (see Figure 32. ARIS Framework (Scheer, 1998)).

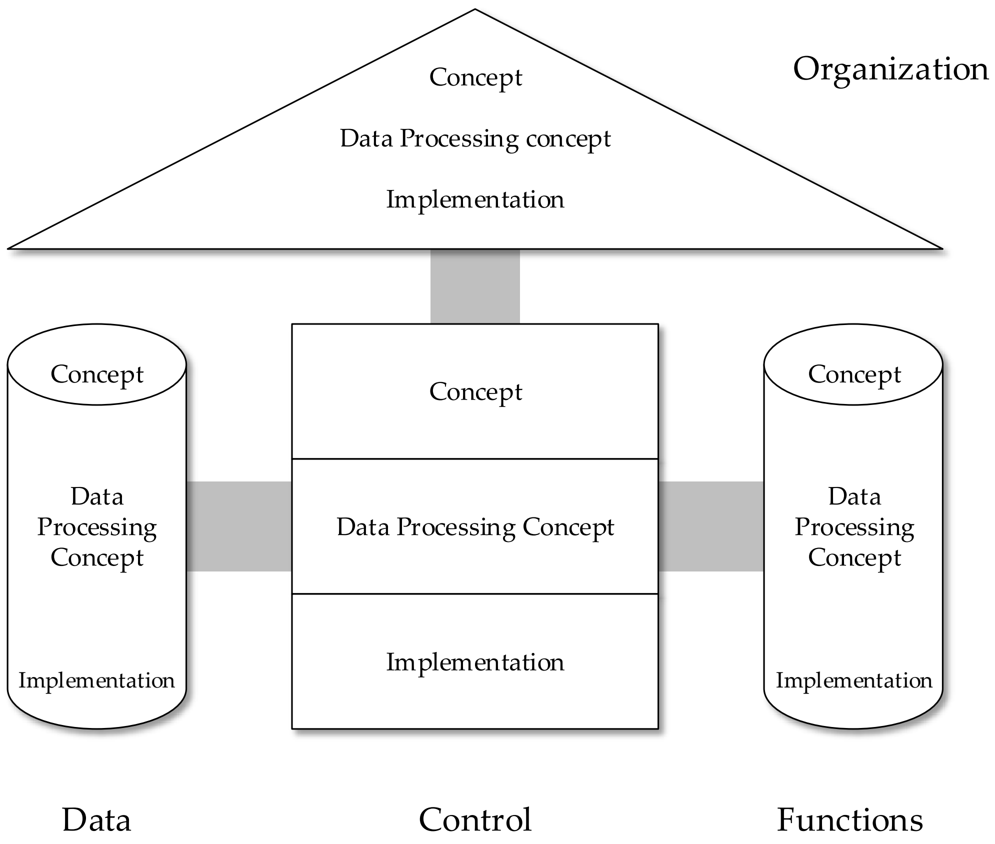


Figure 32. ARIS Framework (Scheer, 1998)

Even though ARIS is intended for system designers to use the framework is focused on business modelling, with an emphasis on process modelling. It is limited to its own syntax and semantics and is therefore not flexible (Lankhorst, 2009). It also does not mention-, nor provide instructions to, the capture of design rationale as it concentrates on process modelling and not the design process of an architecture.

3.3.7 Federal Enterprise Architecture Framework (FEAF)

The Federal Enterprise Architecture Framework (FEAF) was developed by the Federal Chief Information Officers (CIO) council. It suggests a common method for organization design and performance improvement. Its main goal is to provide guidance on how to implement EA and suggest a best practice approach for federal governments, see Figure 33. Federal Enterprise Architecture Framework (CIO Council, 1999).

An architecture can be partitioned into four distinct levels, according to the FEAF framework:

- **Business Architecture:** the business architecture elaborates on people, processes and relationships between them.
- **Data Architecture:** the data architecture details how information is won, shared and extracted through the organization.
- **Application Architecture:** the application architecture lists the software components that process the data and how these components relate.
- **Technology Architecture: in** the technology architecture the hardware components that support the above three layers are made explicit. Also their interdepending relationships are shown.

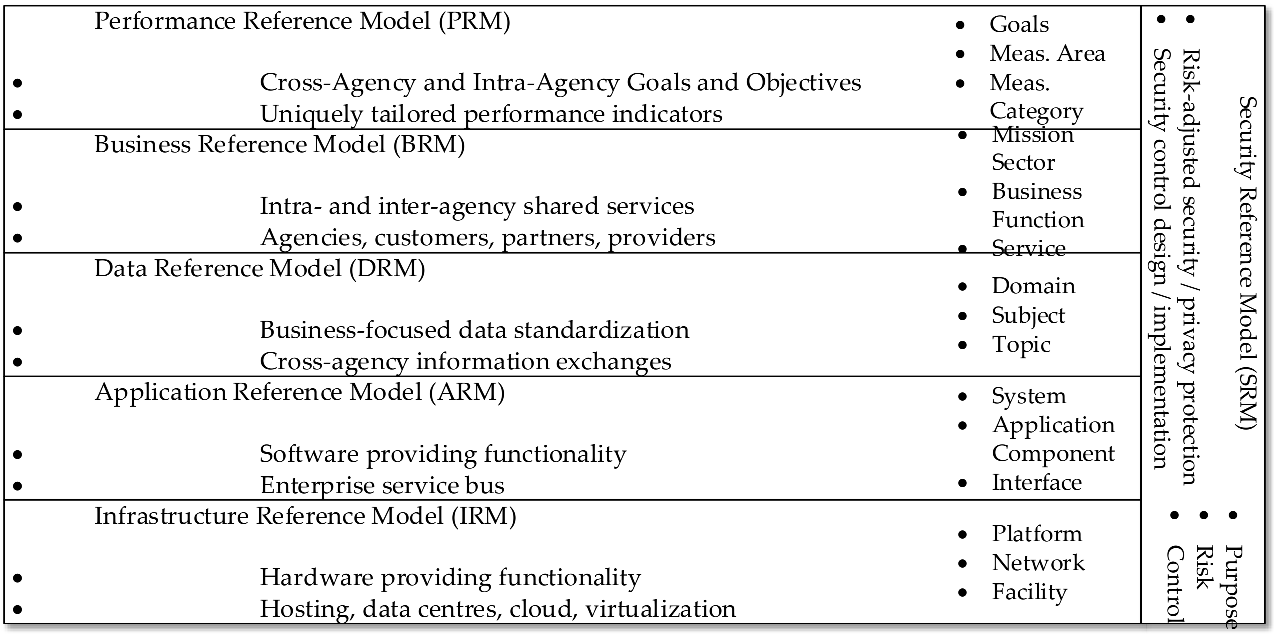


Figure 33. Federal Enterprise Architecture Framework (CIO Council, 1999)

Even though the FEAF framework provides a breakdown of an architecture and guides which steps to follow, how to approach development details on design reasoning are not present. The framework mention that design drivers should stimulate the development of the architecture, but does not guide on how to go about those decisions nor does it provide clear instructions on if and how you should provide design rationale.

3.3.8 Department of Defense Architecture Framework (DoDAF)

The Department of Defense Architecture Framework (DoDAF) is another example of an architecture framework developed for and by the US federal government (C4ISR Architecture Working Group, 2009). The framework provides guidance on how to address various stakeholder concerns through multiple viewpoints. Its main goal is to combat the complexity of large enterprise systems by creating a partitioned overview of the whole system from various stakeholder perspectives. The different viewpoints offer detailed specifics in order to address concerns from multiple different domains.

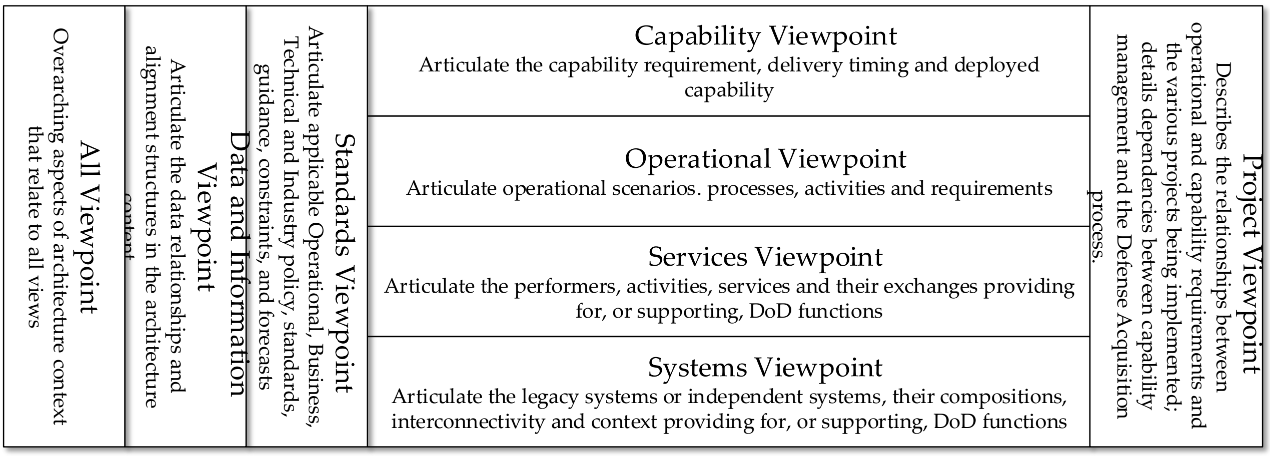


Figure 34. DoDAF v2.0 (C4ISR Architecture Working Group, 2009)

The DoDAF framework is designed to form a holistic system view for large and intricate systems, with an emphasis on its context and how the different perspectives relate. The accompanying documentation provided by the DoD does mention the writing of an Architectural Description (AD) which should include design rationale. However, instructions on how to do so are not present.

3.4 Architecture Description Languages

In the previous paragraph, the main industry architecture framework are analysed in the context of design reasoning and design rationale. As these frameworks often provide high over guidance on how to establish and design an architecture, they often omit provision of a language or notation that demonstrates how the architecture should be represented. In this paragraph, the goal is to determine to what extent the most used Architecture Description Languages (ADLs) in the industry specify or support design rationale and design reasoning. ADLs are defined as *any form of expression for the use in describing architectures* (ISO / IEC / IEEE, 2011). ADLs have a predefined syntax and clear semantics for representation and are suitable for the complete creation and communication of an architecture design. The ADLs chosen for analysis are based on their status in the industry.

3.4.1 ArchiMate (AM)

ArchiMate is the most used open source ADL in the context of enterprise modelling (The Open Group, 2012), and is based on the IEEE 1471 standard (ISO / IEC / IEEE, 2011). ArchiMate distinguishes itself from other notations and languages by its much wider scope compared to its more technical and detailed counterparts like AADL and SysML. An example can be seen in Figure 35. ArchiMate Layered View.

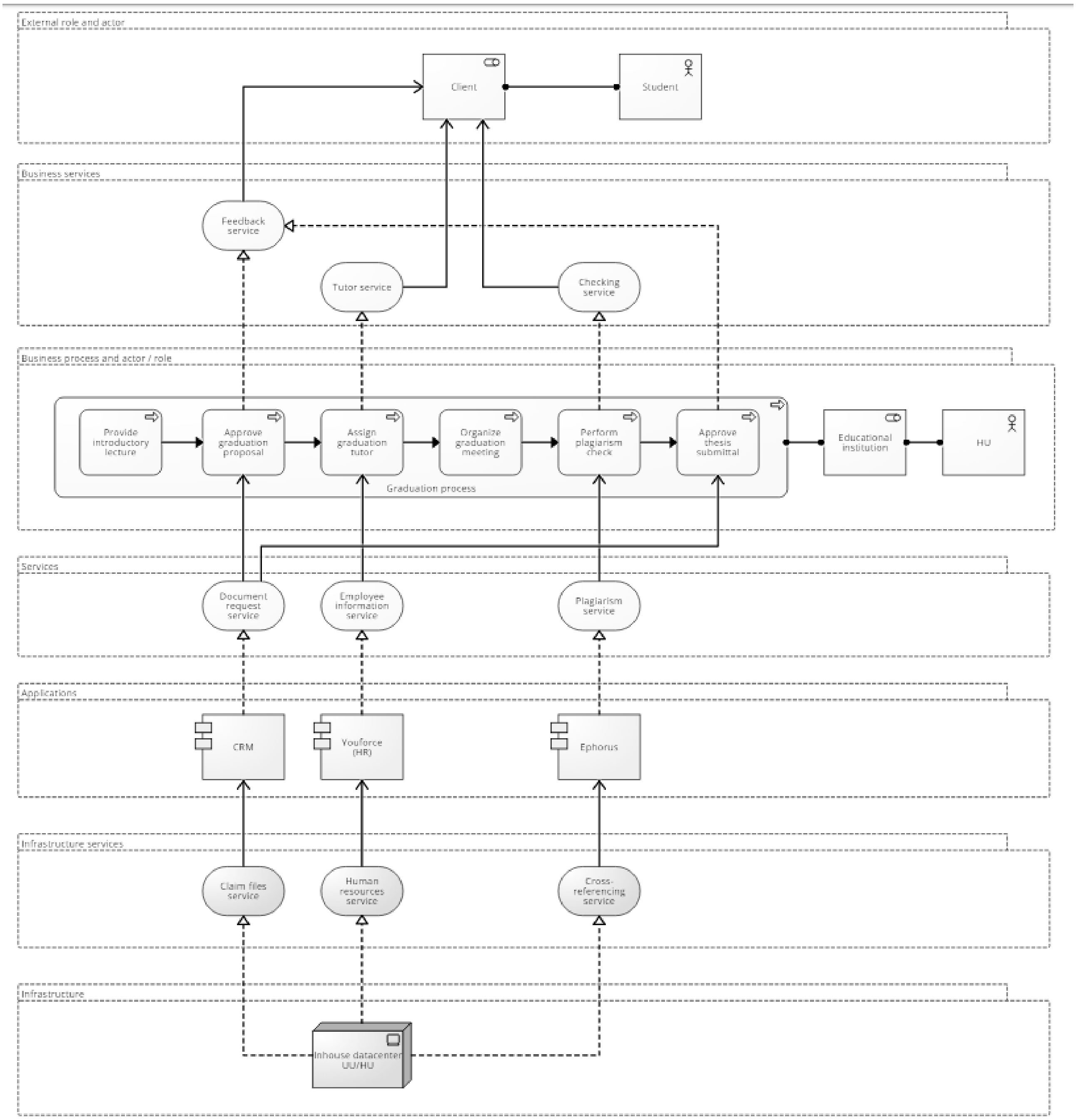


Figure 35. ArchiMate Layered View

In the figure above we can see a subset of the architecture of an organization that crosses multiple domains. This is called the layered view, which basically means the architecture design crosses multiple layers of architecture. From the bottom up, we can analyse the architecture in terms of its infrastructure, its executed services, applications that are ran because of it, realized services due to those applications, business process activities that call upon these application services and how a client may require the services realized by the business process.

In the context of design reasoning and -rationale, however, we notice a distinct lack of design reasoning support. Even though ArchiMate offers full support of completely designing enterprise architectures, ArchiMate does not offer an extension for, say, the design process itself. The ArchiMate 3.0 spec also does not mention the capture of design rationale, nor does the language itself provide instructions on how to do so.

ArchiMate does have a motivation extension, however, that allows designers to model intentions, drivers, requirements and goals of stakeholders. This extension is based on the Business Motivation Model (BMM) standard. Even though it may seem related, this only allows designers to model existing system concepts but is not related to design reasoning. Design reasoning is associated with motivation and argumentation *during* the architecture design process, *by* the architecture designers themselves. The motivation extension solely allows architects to model the motivational elements of the existing system architecture, which is not relevant to the topic (Plataniotis et al., 2014).

3.4.2 Business Process Model and Notation (BPMN)

Business Process Model and Notation (BPMN) is one the most used graphical notations for modelling business processes (Object Management Group, 2011). It is maintained by the Object Management Group (OMG), who released version 2.0 of the notation in 2011. BPMN relies on utilizing flowchart techniques to create Business Process Diagrams (BPDs) to represent business processes to both technical and business users. An example can be seen in Figure 36. BPMN diagram of a patent application.

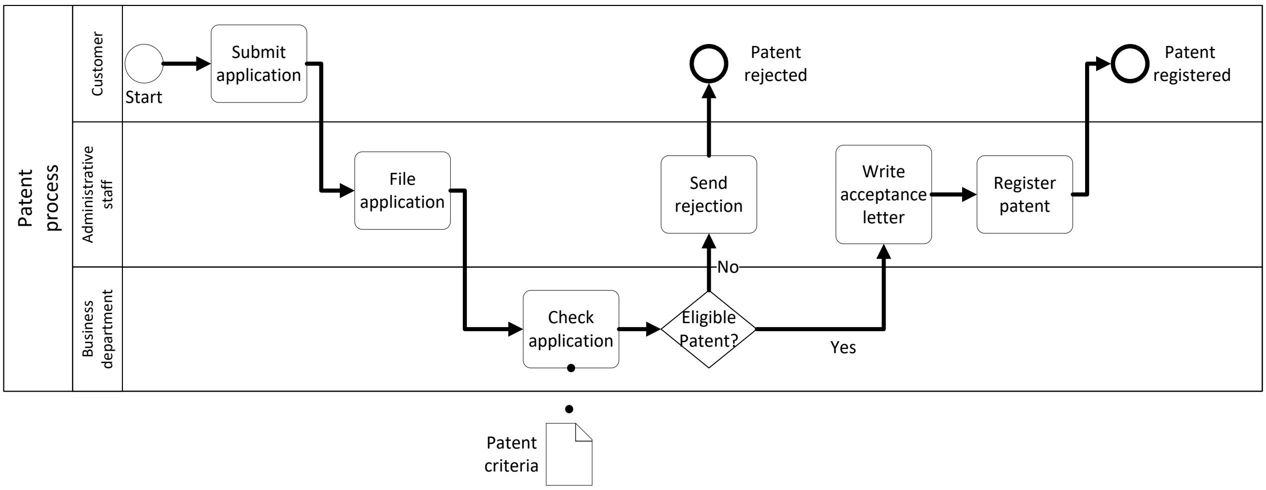


Figure 36. BPMN diagram of a patent application

In the context of design reasoning, however, no support for the reasoning and argumentation process is found. BPMN offers complete support of extensive modelling of business processes and even offers architecture designers guidance on the design process itself by demonstrating which BPMN elements to model first for simplicity. However, support for argumentation in the design process itself is not present. Also, neither the syntax nor the spec elaborates on the capture of design rationale.

3.4.3 Place / Transition net (Petri net)

A place / transition (PT) net (Petri net) is a well-known formal modelling language that represents system behaviour (Peterson, 1981). It was developed by Carl Adam Petri in 1939 and often used and maintained since then. Like BPMN, Petri nets offer a notation for the graphical representation of business processes. Petri nets distinguishes itself by offering a formal notation, unlike BPMN, UML and EPCs. This means Petri nets lend themselves to mathematical analysis and are suitable for demonstrating choice, iteration and concurrency in processes. An example can be seen in Figure 37. Petri nets Diagram.

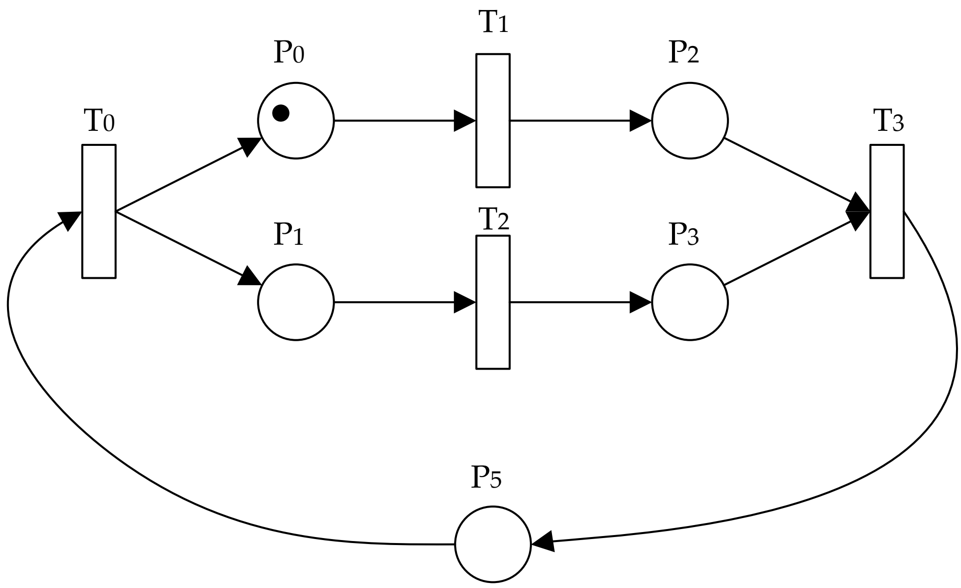


Figure 37. Petri nets Diagram

A petri net diagram consists of tokens, places, transitions and arcs. The black dot represents a token and allows the transition to fire if enough tokens are present. If the transition fires, the required tokens are consumed and appear as an output in the next place. Some transitions require multiple places to contain tokens, however. Considering the mathematical and binary nature of the diagram, the notation lends itself for modelling concurrent behaviour of systems and mathematical analysis. In the context of design reasoning, however, Petri nets do not inherently support argumentation on design decisions. Petri nets are suited for process analysis, mathematical iteration and definition and binary execution behaviour. Petri nets are not designed to include qualitative aspects of design decision argumentation, nor does it support any text based rationale capture.

3.4.4 Unified Modelling Language (UML)

The Unified Modelling Language (UML) is arguably the most used ADL due to its general nature. UML was developed to support visualization of system designs with a general purpose mind set (Rumbaugh, Jacobsen & Booch, 2004). It was adopted as a standard by the OMG in 1997, who have been maintaining the ADL. UML supports the representation of activities, system components, system behaviour and software interaction, among others. An example can be seen in Figure 38. UML diagram of BPMN.

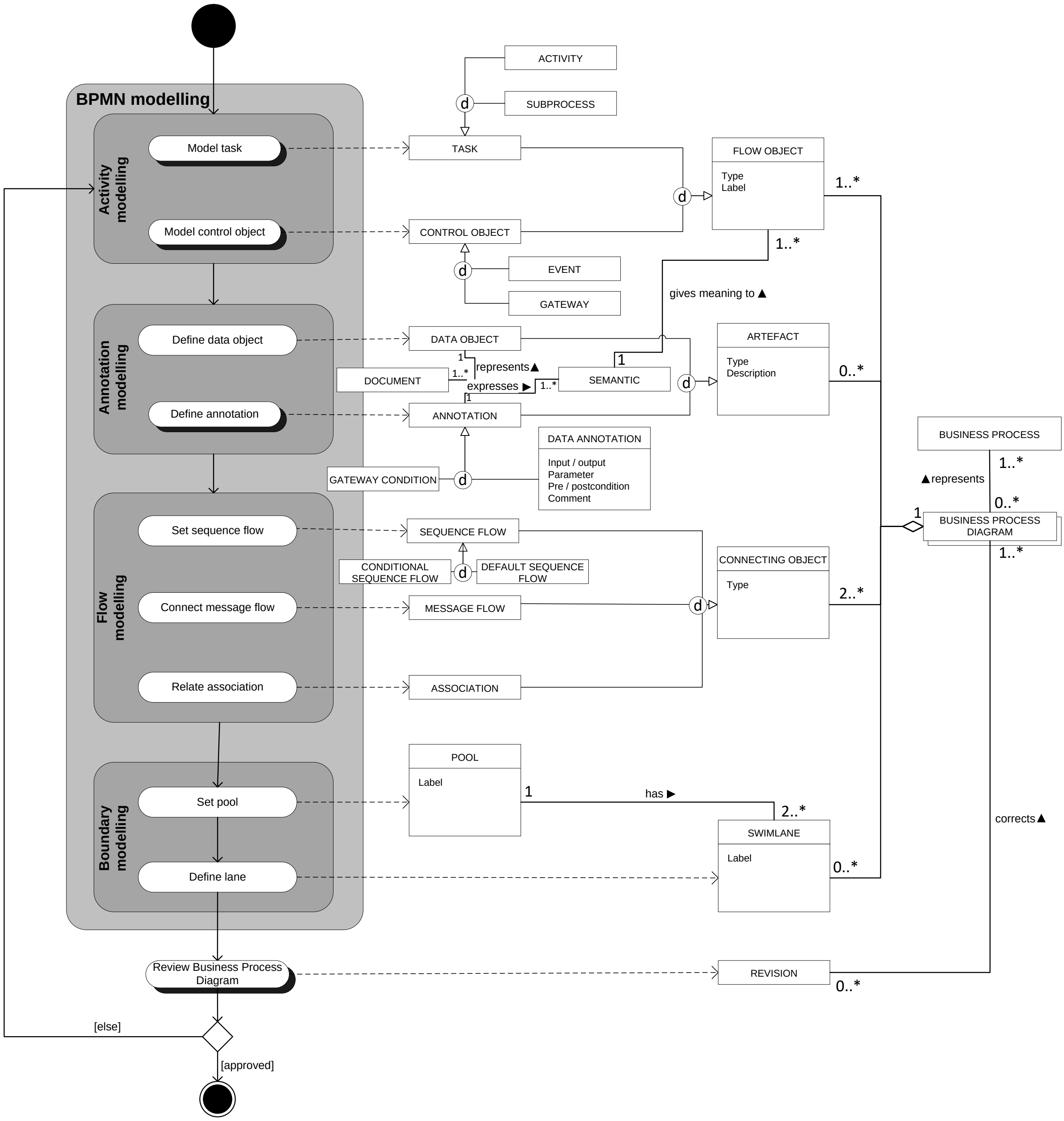


Figure 38. UML diagram of BPMN

Considering the wide scope of UML, various different diagrams exist in the UML standard. In the example above we can see a UML Activity Diagram (left hand side), combined with a UML Class Diagram (right hand side). The left hand side elaborates on the process that is to be executed whilst the right hand side details the concepts and elements that are produced by the specific activities of the process. On top of that, UML also contains component diagrams, use case diagrams, sequence diagrams, communication diagrams and more.

In the context of design reasoning, we notice there is very little explicit support for design reasoning, not in the immediate diagram structures nor in the UML specification. Even though UML has a wide range of diagrams to support modelling, no specific diagram exists for the capture of design rationale. Also, UML offers no guidance on *how* to approach making design decisions or if argumentation and reasoning should be employed. The argument can be made that UML Diagrams are very suitable for a design reasoning method. On top of being an industry standard, UML diagrams can comprehensively depict both guidance in the design reasoning process through UML Activity Diagrams and demonstrate which design rationale elements should be made explicit through UML Class Diagrams. On top of visual clarity, UML diagrams can comprehensively demonstrate relationships and dependencies. The Architecture Rationale and Elements Linkage (AREL) method by Tang et al. (2007) is a design rationale approach based on UML for these reasons.

3.4.5 Architecture Analysis & Design Language (AADL)

The Architecture Analysis & Design Language (AADL) is a relatively well known ADL used to design software and hardware architecture elements specifically. It was standardized by the Society of Automative Engineers (SAE) (Feiler, Gluch & Hudak, 2006), and mostly concentrates on systems currently in use. Therefore, the AADL has an emphasis on runtime elements in the execution platform. AADL supports both the design and analysis of architectures and is roughly comparable to the Component-Connector diagram of UML. A simple example can be seen in Figure 39. AADL Diagram.

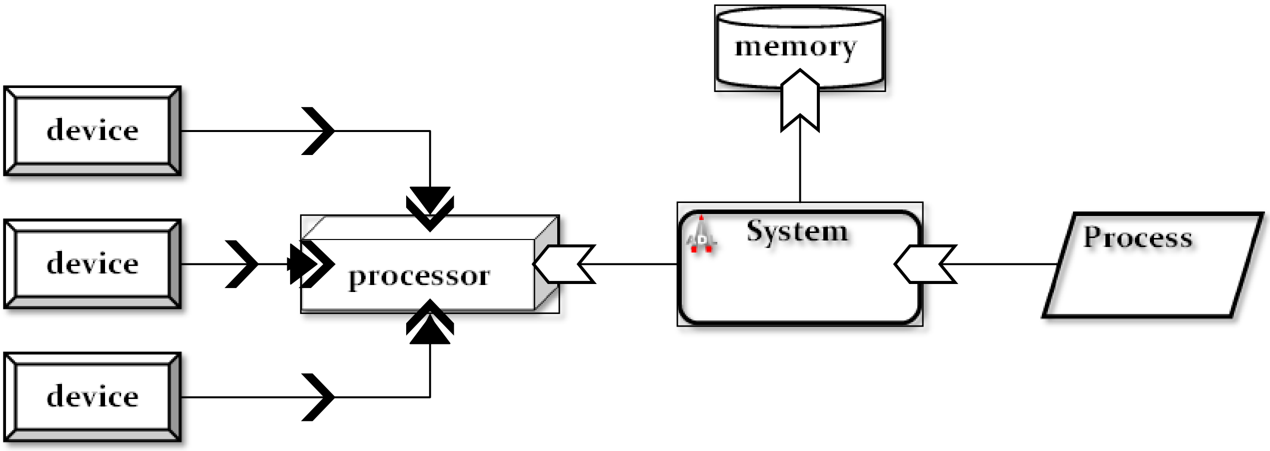


Figure 39. AADL Diagram

The example shows a simple system, where three hardware devices generate data that is aggregated by a single processor. The system then accesses data from this processor, which combines it with memory data which is then utilized by a process.

In the context of design reasoning, we notice very little immediate evidence for design reasoning support. For the most part, AADL provides a language and syntax in order to design both software and hardware architectures and their relationships. AADL can also be used to analyse system qualities such as performance and reliability through various AADL supported software tools. However, AADL concentrates on designing in the execution platform and runtime aspect of the system. Hence, its focus is not in providing designers with a guide on reasoning and argumentation in the design process itself. Also, no mention of the capture of design rationale is mentioned in the AADL specification.

3.4.6 Systems Modelling Language (SysML)

The Systems Modelling Language (SysML) is, like UML, a general purpose ADL for the design and analysis of systems architectures. SysML also offers verification and validation options. Similar to UML, SysML has a wide scope that allows for a broad support spectrum (Friedenthal, 2014). SysML can be seen as a subset of UML's wider range of supported objects, with an emphasis on software oriented elements. SysML offers a variety of diagrams, not unlike UML, like the package diagram, use case diagram, requirements diagram, state machine diagram etc. An example can be seen in Figure 40. SysML Requirements Diagram.

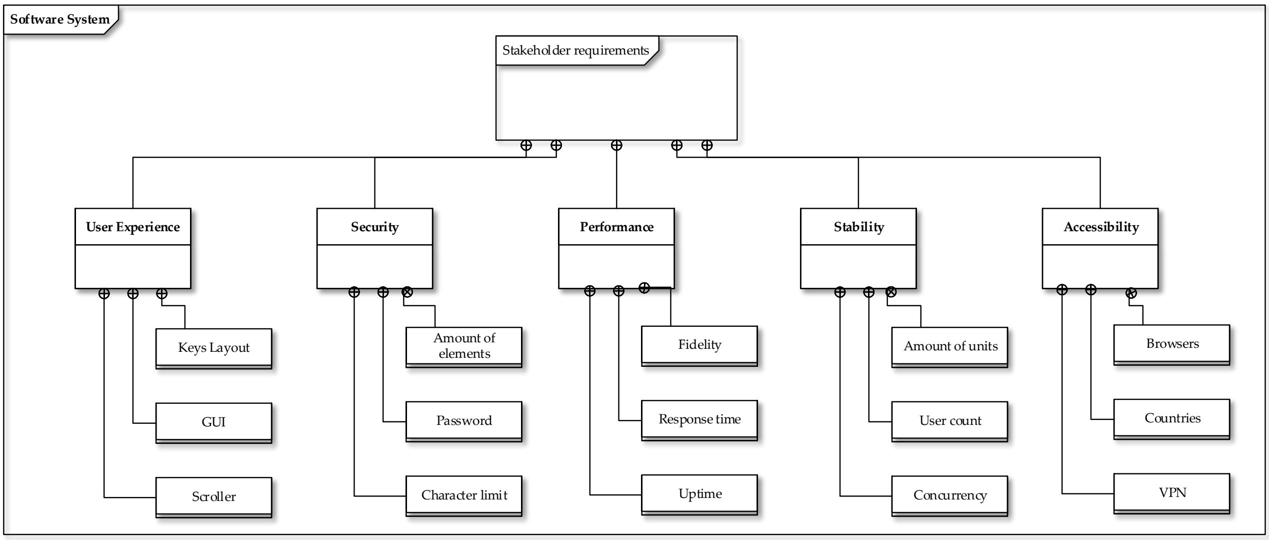


Figure 40. SysML Requirements Diagram

The SysML Requirements Diagram represents a software system that contains 5 defined categories of stakeholder requirements; user experience, security, performance, stability and accessibility. Each category consists of concrete requirements measures. In the context of design reasoning, we notice limited support for design reasoning as a concept. Overall, the SysML is not designed with the design process itself in mind. SysML concentrates on offering a modelling language for designing systems, with an emphasis on software systems. The language itself does not inherently offer guidance on how design reasoning should be applied, nor does it mention how the design process itself should take place.

However, the SysML 1.4 specification does mention that SysML supports the notation of design rationale. It mentions that rationale documents the justification for the design, requirements and design decisions. The specification describes the capture of design rationale by explicitly mentioning users can attach a rationale to any model element in the language. For example, a user may specify a relationship between requirements and attach a number that references back to more detailed justification documentation. The SysML specification does not, however, elaborate on how this detailed documentation should be made. Therefore, considering rationale is only uttered once in a 4 sentence paragraph, the support for design reasoning and design rationale is very limited.

3.4.7 System Architecture Description Language (SysADL)

System Architecture Description Language (SysADL) is a SysML profile for expressing architecture descriptions (Leite, 2013). SysADL uses well-known elements from the ADL community to effectively describe architectures. Like SysML, it focuses on documenting systems architectures. SysADL can be seen as different from SysML, however, due to its emphasis and focus on describing software architectures specifically. An example can be seen in Figure 41. SysADL example of Temperature Sensor.

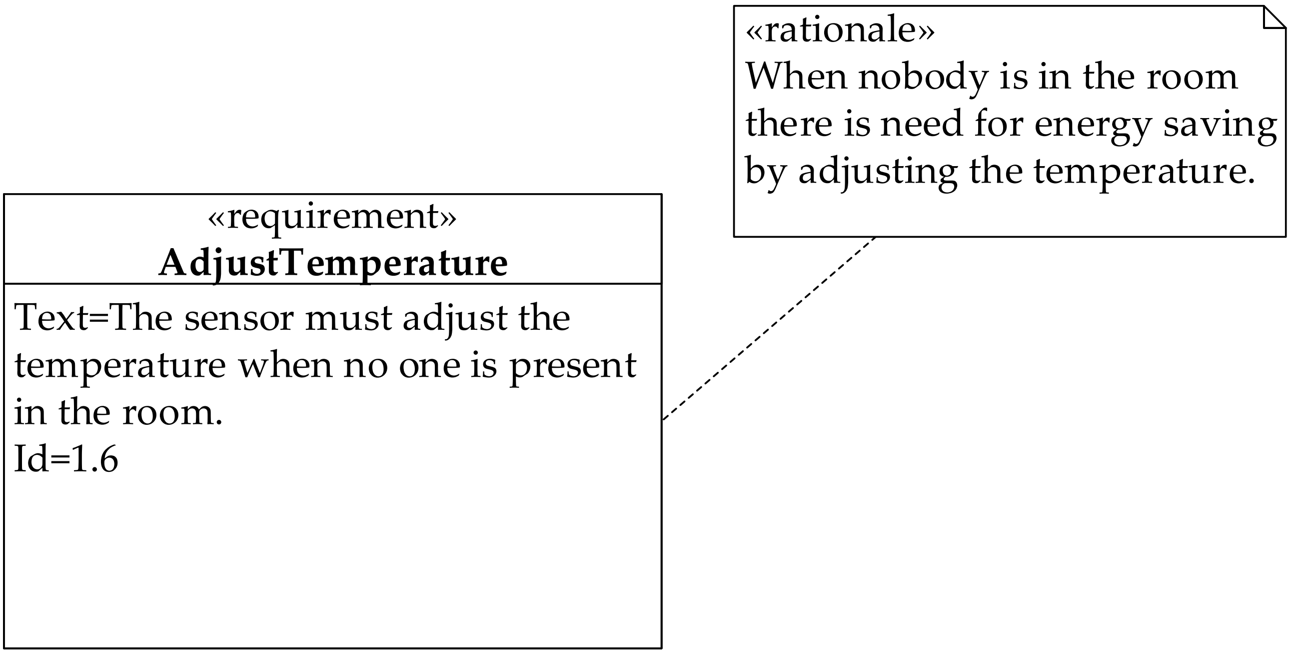


Figure 41. SysADL example of Temperature Sensor

The SysADL language has obvious rationale support included in the model elements itself. As SysADL focuses on the description of software architectures, specifically, the SysADL notation has a separate element for specifying the rationale for a requirement or design decision. The accompanying paper and documentation also mention the description of design rationale is supported by this element. However, no concrete guidance is given on what the rationale should contain and how it should be documented.

3.4.8 Generalised Rationale-aware Architecture Specification (GRASP)

Generalised Rationale-Aware Architecture Specification is an ADL that is designed to specifically express software architectures. As the name suggests, GRASP puts an emphasis on the support for design rationale. The aim of GRASP is to support architecture rationale expression according to the definition by Perry & Wolf (1992). This definition contains elements, form and rationale specifically. GRASP aims to support all three. An example can be seen in Figure 42. GRASP example of Temperature Sensor Rationale.

```
rationale Temp_Rationale {
        reason #'Ability to sensor temperature in order to save energy';
        reason #'Allows for temperature data and analytics';
}
template TempTemplate () because Temp_Rationale {
        requires Tempsensor
        provides Tempdata
        check Tempsensor->measureTempData = 1;
```

Figure 42. GRASP example of Temperature Sensor Rationale

GRASP is different from other ADL's as it is a complete formal notation. The ADL's syntax itself is in code-form. The main benefit is the improved ability to be processed by a computer but lacks readability by the users themselves, especially when inexperienced or unfamiliar with the notation. However, the provision of design rationale is most definitely supported by the ADL itself.

The accompanying documentation also mentions how design rationale should be used in terms of structure. However, no mention is made on how the rationale should be documented nor does it explicitly guide to what extent rationale should be captured.

3.5 Design Reasoning

As the foundation of architecture is established and the lack of design reasoning and rationale made distinct, a thorough understanding of design reasoning as an entity is required. It is assumed that design reasoning is currently underutilized, yet beneficial when used. These assumptions are tested and a grasp of design reasoning's definition and impact is sought. First, a definition is sought. Secondly, design reasoning utilization in the industry is explored, including its main benefits. The last paragraph covers the main theoretical benefits of design reasoning utilization.

3.5.1 Definition

Design reasoning as an entity is uttered often enough, but often in different contexts. Design reasoning can occur in architecture design in various different backgrounds or domains, e.g. from a technical or non-technical perspective. This makes finding a concrete definition more difficult.

It is important to note that when design reasoning is mentioned in this project, the relationship with systems architecture is assumed. Design reasoning can be seen as separate from systems architecture and solely refer to the reasoning process to design a drawing, for example. Do & Gross (1996) mention design reasoning refers to making decisions, expressing ideas, evaluating alternatives and taking action. Rittel (1987) mentions reasoning refers to all aware mental operations and communicate to others. Deliberation, consideration, arguing, pondering, debating and presenting these ideas and thoughts comprise the process of reasoning. Essentially, design reasoning constitutes a rational process where logical reasoning is utilized in order to come to a decision.

Parnas & Clements (1986), although being software oriented, mention that a complete rational design process is unobtainable. They suggest that humans will always remain somewhat biased due to personal experience and limited comprehension of facts. However, they also mention striving towards a complete rational process is still fruitful. Even though a rational design process is an ideal, mimicking it as closely as possible is still useful for a variety of reasons. Designs should still be as rational and structurally sound as possible to obtain a better quality design.

Also, designers still require guidance when designing complex systems. The overarching idea is that a nearly rational design process is still better than a biased and instinctive architecture design process.

Design reasoning is established to be the process of deliberation, argumentation and making decisions. Design reasoning is applied when, during the architecture design process, a design problem occurs. Tang et al. (2006) define design rationale as the knowledge captured by reasoning on the design decision. Therefore we can establish a process – product relationship between these two concepts. If design reasoning refers to the argumentation process, design *rationale* can be seen as the knowledge that is produced by this argumentation and deliberation when a design decision has been carefully considered and a decision has been made (see Figure 43. Figure 43. Process – Product relationship of DR).

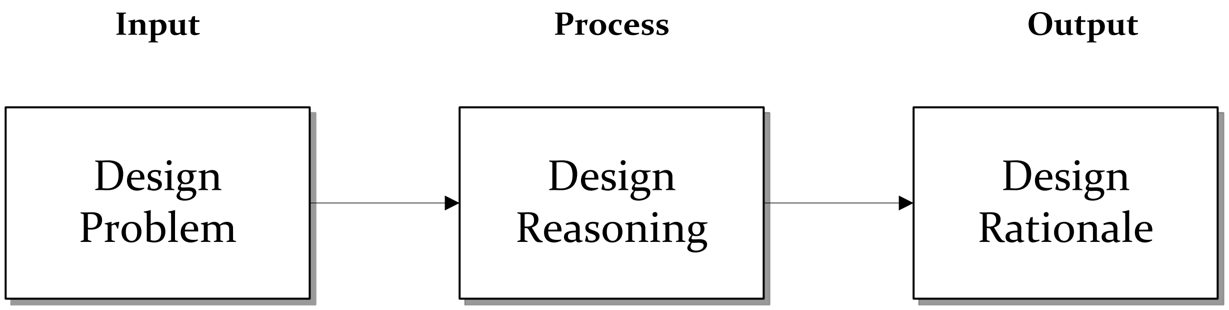


Figure 43. Process – Product relationship of DR

3.5.2 Principles

In order for the experiment observation to be successful, design reasoning needs to be dissected into observable parts. If these are not identified, there is no way in which the design reasoning utilization (dependent variable) can be measured. Therefore, the goal of this paragraph is to identify and define specific elements that comprise design reasoning so that they may be observed during the experiment phase. Also, this forms the basis for which elements need to be present when designing and creating a design reasoning approach.

As previously established, design reasoning can be seen as the process of deliberation, argumentation and reasoning on architecture design decisions. Tang (2011) identifies key principles to adhere to if design reasoning is to be utilized whilst moving away from naturalistic decision making. They are summarized as follows:

- *Reasoning and Inferences* – architecture designers should identify the situation, reason and argument and logically conclude.
- *Problem Structuring* – architecture designers should be able to form a logical overview of the problem, requirement identification and design planning.
- *Assumption Analysis* – architecture designers should identify and recognize any assumptions behind design decisions and know whether or not these assumptions may have an effect on the design and system.
- *Constraint Analysis* – architecture designers should recognize constraints that are formed by stakeholder requirements, regulatory constraints or project resource limitations.
- *Option Analysis* – architecture designers should always consider alternate options equally throughout the design process and not anchor their thoughts on a singular option.
- *Trade-off Analysis* – architecture designers should always consider trade-offs and compare priorities and evaluate requirements when faced with a design that cannot satisfy all stakeholder requirements or constraints.
- *Risk Analysis* – architecture designers have to consider the possibility of negative effects of design decisions. These risks have to be weighed and taken into account when design decisions are made, including the risks of alternatives.

Even though these principles seem clear, in order to embed these principles into the architecture design process they cannot be simply told to architects. There is an element of awareness that needs time as architects don't simply pick up a new method of working. Also, the principles have to be transformed into an intuitive and approachable manner so that it can be easily picked up by architects and applied effectively. The artefact design & creation phase will tackle these adoption issues.

3.5.3 Industry Adoption

In order to understand the concept of design reasoning any further and add context with regards to its relationship to architecture, it is imperative to grasp to what extent design reasoning is currently utilized in the industry. Questions regarding its place and right to exist are central in this paragraph. Grasping this information adds further context and information to the conclusions and findings of this study. Also, identifying its place in the industry can support or debunk the problem statement in that design reasoning awareness is lacking and naturalistic decision making is, in fact, an issue.

Tang et al. (2006) conducted a survey in which the aim was to gauge the use of design rationale among industry practitioners. These industry practitioners consist of architects and architecture designers with various levels of experience. Overall, the study found that architects do employ design reasoning elements and do capture design rationale. However, the extent to which they do so is limited.
Tang et al. found that 88.9% of practitioners nearly always do, in fact, reason about their design decisions and 85.1% find design rationale to be an important factor in design justification. Also, 80.3% of practitioners nearly always consider design alternatives.

When practitioners are asked more specific questions, however, those numbers differ. In this case, questions are asked that forces architects to reflect on whether their opinion matches their behaviour. In other words, do architects actually utilize design reasoning or document design rationale? The same study by Tang et al. (2006) found that practitioners tend to utilize specific design rationales and not others. For instance, 87.7% of practitioners do use *design constraints* to reason about their design. Similarly, 85.1% do, in fact, mention the *benefits* of their design decisions. Practitioners tend to use specific design rationales that demonstrate the negative side of their designs less. For example, only 55.6% nearly always mention *design weaknesses* in the rationale of their design and 69.1% nearly always use *cost* as a reasoning factor.

Participants were also asked to elaborate on the extent in which they document design rationale and whether or not this knowledge is captured. Overall, 44% of practitioners document discarded design decisions very often. 36% of practitioners almost never do so.

Specifically, a tendency towards certain specific rationales is clear whilst others are overlooked. For example, practitioners often or always document *constraints* (82.7%), *assumptions* (79.1%) and *benefits* (69.1%), yet utilize *complexity* (50.6%), *costs* (45.7%) or *weaknesses* (35.8%) as captured design rationale less.

It appears that design reasoning principles are used in the industry and practitioners do reason about their design decisions. However, not all aspects of design reasoning are utilized (Tang et al., 2006). Also, some design rationale is often captured whilst other design rationales are not. Alternative decisions and their argumentations are also not documented. An important point is that even though the percentages seem quite high, the survey does not provide insight into the *extent* in which practitioners actually use design reasoning nor does it verify if rationale documentation is comprehensive or, in fact, quite limited.

3.6 Design Rationale

As previously established, design rationale refers to the actual knowledge that is captured when design reasoning is utilized. The first paragraph is concerned with correctly defining design rationale as a concept. Then, the main characteristics and types of rationale will be distinguished. The last paragraph identifies the most well-known design reasoning methods and analyses them. This analysis forms the basis for the Method Association Approach elaborated on in paragraph 2.3.3, Method Engineering.

3.6.1 Definition

As mentioned before, if design reasoning refers to the deliberation and argumentation process that leads to a decision, design rationale can be considered the product of that process (see Figure 43.).

The Cambridge dictionary defines rationale as a reason or intention for a specific action. If we take this definition and apply it to architecture design, design rationale can be seen as the reason and justification an architecture designer has for making a certain design decision. This definition is similar to a definition by Lee (1997), who considers design rationale as *''the reasons behind a design decision, the justifications for it, the alternatives considered, trade-offs evaluated and argumentation that led to the decision''*. This definition encapsulates all products of arguing, deliberating, pondering and debating a design problem.

This definition is shared by peers in this research domain. Tang (2007) defines design rationale as the reasons and intentions in the act of designing. This definition, albeit similar, is less detailed on what that rationale should contain. Tang defines it simply as 'all reasons and intentions' that come to design decisions.

3.6.2 Why use Design Rationale?

In order to grasp the contribution of using design rationale, the benefits of using design reasoning and capturing design rationale is theoretically identified in this paragraph.

3.6.2.1 Applications

Design Rationale can be deployed in many useful ways. Burge & Brown (1998) define the set of design reasoning uses as follows:

- *Design verification* – to apply rationale towards the verification whether or not the design meets the designer's intent and, thus, stakeholder goals.
- *Design evaluation* – to verify whether or not the design itself is correct and to compare design choices to each other.
- *Design maintenance* – to keep track of made choices and discarded alternatives. The architect has a backlog of knowledge to utilize when new changes or modifications have to be made to prevent the architect from making a similar mistake.
- *Design reuse* – to determine what elements of the architecture design can be reused in new projects or when faced with new changes. The design rationale suggests which elements are critical or not when new requirements arise due to the justifications for previous design decisions.
- *Design teaching* – to teach and inform new personnel about a current system. The design rationale not only features insight into how the system works but details on why important decisions were made. This provides elaborate information on the system and its relevant context and conveys more information on the system as a whole.

- *Design communication* – to communicate relevant system information both during and after the design process. The design rationale contains critical information to all parties that are subject to or affected by the design. Providing design decisions and the justifications behind them conveys useful information to others. Other parties can in turn provide proactive feedback and support when presented with contextual information on other systems.
- *Design assistance* – to assist and support architects during the design process. Design rationale makes sure design choices are made explicit, including their justifications. These choices can be iteratively verified and evaluated and retrieved for validation. It forces architects to clearly document their thought process and argumentation in a logical fashion.
- *Design documentation* – to provide a holistic view of a system that includes the dimension of time. Design rationale documentation can assist architects if the final system is completely dissected. However, if a history of the system is also in place a richer body of knowledge regarding the system and its behaviour is available. Should problems arise, including those with other systems, the ability to trace back design decisions can simplify the understanding.

3.6.2.2 Architectural Degradation

Systems that have to be designed, implemented, tested, maintained, managed are subject to the dimension of time. As architectures mature, they need to evolve in order to meet a new set of stakeholder requirements. Over time, architectures become increasingly brittle – i.e. the system is less adaptable and becomes resistant to change (Perry & Wolf, 1992). This makes sense, considering the design decisions that have been made previously and their justifications are not kept. If changes occur, new problems may surface that could have been prevented with this knowledge of the system.

Take, for example, an architecture of a system design that consists of servers that handle web traffic and applications that run on them. Say in 2012 the decision has been made to not update the HR application due to its inability to connect to old servers. In 2016, however, new servers are introduced that replace the old ones and the applications can be updated company-wide.

The board still wants to use the old servers for another department, however, and the servers are installed in a different building. However, the other department now has updated applications that do not run on those servers. This causes the organization to revert back a hardware change or divest into the application update, causing a multitude of unintended problems that may have major costs that were not envisioned. This phenomenon is called *architectural degradation* and consists of two elements:

Architectural drift leads to an increasingly brittle architecture and, thus, system by introducing design decisions into the descriptive architecture that were not included by the prescriptive architecture. In other words, design decisions have been made in the as-implemented architecture that were not previously envisioned in the as-designed architecture.

Architectural erosion leads to an increasingly brittle architecture and, thus, system by introducing design decisions into the descriptive architecture that violate design decisions in the prescriptive architecture. In other words, design decisions have been made in the as-implemented architecture that are in direct conflict with the as-designed architecture.
If design rationale is kept, however, the board could have traced back design decisions as to why the applications were not updated, preventing a financial setback and major conflict in business operations and technology support. Not every problem is as disastrous, obviously. However, architectural degradation, i.e. the architecture's resistance to change, increases. This increase causes a lack of coherence in the systems and their relations, clarity in design and purpose and overall increase in system obscurity (Perry & Wolf, 1992). Design Rationale reduces the cost of evolving systems and architectures and allow them to more simply and gracefully adapt to changes.

3.6.2.3 Knowledge Management

Another crucial benefit to using design rationale is knowledge management. As opposed to the architectural argument, documenting design rationale guarantees the retention of knowledge. Considered alternatives, made decisions and assumptions, constraints, options, arguments, implications, related systems, considered risks and trade-offs all contribute the body of organizational knowledge. If design rationale is not kept, however, this information is left implicit in the minds of architects. When personnel changes occur organizational knowledge, over time, decreases (Tang & van Vliet, 2008). This knowledge could be retained if made explicit at the time. Plataniotis et al. (2014) concur with this premise using the term *architectural knowledge vaporization.*

The researchers suggest that the industry learned from the field of software architecture that architectural knowledge vaporizes if design rationales are left implicit and that this consequence also applies for the field of enterprise architecture.

Also, as mentioned previously in the applications of design rationale, this knowledge can be used to teach younger architects who do not possess previous experience regarding the specific architecture or domain. The documentation of design rationale provides organizations with bodies of knowledge that can be used to train and teach new architects.
If not done, architects have to repeatedly learn specific architectures and system which costs an organization resources in terms of man hours.

3.6.3 Types of Rationale

In order for the main goal of this research to be achieved, design rationale needs to be dissected into distinguishable parts. If these are not identified, there is no way in which the design rationale capture can be implemented in the design reasoning approach. Therefore, the goal of this paragraph is to identify and define specific elements that comprise design rationale so that they may be defined and incorporated during the artefact design & creation phase.

There are various types of design rationale. Burge & Brown (1998) identify the following five types:

- *Argumentation based* – the design rationale presents arguments that define the design. These arguments present pros and cons for each alternative, option and issue.
- *History based* – the design rationale represents the design history of a system. This type of rationale includes the dimension of time so design decisions are included chronologically.
- *Device based* – the design rationale is based on a model of the device itself. The model can simulate the behaviour of the device, where the design rationale can be deduced from.
- *Process based* – the design rationale is completely interwoven with the design process itself and therefore guides the format of the rationale.
- *Active document based* – the design rationale is predefined and already saved in the system. When an architect designs an architecture, the design rationale system automatically generates rationale for the elements designed by the architect.

These types are not exhaustive and may simultaneously be true. For example, design rationale can be argumentation based and history based at the same time. The design rationale intended in this study is primarily *argumentation based*, as the goal of the research is to provide architects with an approach that supports reasoning and the process of arguing and deliberation on design decisions. However, the aspect of time in history based rationale is very important.

Also, process based rationale is an ideal to strive since practitioners consider design rationale capture as disrupting and resource intensive (Tang et al., 2006). Ideally, the approach should be argumentation based, interwoven in the architecture design process whilst taking the dimension of time into account.

Tang et al. (2006) define a set of generic design rationales that further specify argumentation based design rationales:

- *Constraints* – limitations placed on design decisions based on external factors like organizational or lawful regulations or project constraints.
- *Assumptions* – unknown factors that have a direct influence on the design.
- *Weaknesses* – the elements that cannot be achieved by the design decision.
- *Costs* – both explicit and implicit cost of a design decision implication.
- *Complexities* – the relative and subjective measure of a design's complexity, in terms of maintenance and implementation.
- *Certainty of design* – the measure of the design's effectiveness, in terms of its effectiveness in meeting stakeholder requirements.
- *Certainty of implementation* – the measure of a design's implementability, in terms of schedule and cost, and risk the organization takes by implementation.
- *Trade-offs* – the consideration and comparison of alternatives and its pros and cons.

However, not all industry researchers agree upon a standard set of rationale elements that need to be captured. Tyree & Akerman (2005) for example, whilst sharing some of the rationale types above, also offer alternative specific rationale:

- *Positions* – A comprehensive list of alternative considerations or viable options that have been considered. These options have to be detailed enough to demonstrate that option has been carefully considered.
- *Arguments* – Additional factors that may constrain the design decision.
- *Implications* – All identified eventual implications the design decision may pose. Things like increased cost, reduced risk etc.
- *Related requirements* – The design decision has to be accountable and transparent. In this table the relevant and associated requirements that drove that decision have to be made explicit.

In order to agree upon a set of specific design rationales that are to be incorporated during the artefact design and creation phase, a few sets of rationale types will be compared. The most used and agreed upon design rationale types will comprise the final set of design rationale elements. If a researcher suggests a more specific design rationale as opposed to a more generalized statement from their peers, the more specific rationale will be chosen. Together, this will form the basis for the elements that need to be incorporated in the design rationale part of the artefact.

3.6.4 Rationale Capture

Rationale can be produced in various different ways. Lee (1997) identifies 5 major methods in rationale capture:

- *Reconstruction* – the reconstruction approach constitutes the production of rationale by reasoning from existing knowledge (introspection). This existing knowledge can be current architecture models, video material, audio or through transcription text analysis. Reconstruction is completely separate from the design process, so it does not disrupt any design activities. It also guarantees a more careful and detailed production of rationale, considering the architect solely focuses on producing rationale. However, it has to be a complete new process after design has taken place. This is less efficient and takes up more time and resources. Also, an element of knowledge loss can occur considering the time difference between the design phase and rationale phase.
- *Record and Replay* – this approach produces rationale *during* the design process. Design problems are identified, alternatives are considered and criteria and claims are defined whilst design is taking place. This can occur through digital means (forums, videoconferencing) or through regular face to face meetings. Informal or semiformal rationale representations are suitable for this approach considering they least distract from the design process itself.
- *Methodological By-product* – in this approach the rationale is produced as a logical by-product of following a method in the architecture design process. This method then constitutes the capture of rationale. This approach is ideal because it allows the architect to simply follow a method that generates the rationale in the design process. The designer does not have to divide attention to multiple elements. The trick is finding a method that produces good rationale whilst not distracting from the design process.
- *Apprentice* – the apprentice approach consists of the interaction between a designer and a computer system. The system verified design decisions made by the designer and asks questions whenever an action is made it does not understand. In other words, whenever a designer makes a decision that differs from the system's expectations it asks the user to justify that difference. This justification produces the rationale. Every time the user enters a rationale for a decision, the system becomes smarter with an added rationale. In reality, however, such a system is difficult to implement. The uniqueness of every architecture makes it hard to produce a rationale. And a fully-fledged AI that can recognize whenever a decision requires a rationale or not is difficult to fully realize.
- *Automatic Generation* – the automatic generation approach constitutes a system that produces rationale from an existing rationale base. The system analyses a complete history of designs and defines the how's and why's of the performed actions. This approach is ideal for organizations with large, already existing bodies of designs. In reality, however, it is difficult to find such a consistent design base. Also, the designs have to be in the correct format to be processed by such a system.

Currently, only the reconstruction, record and replay and methodological by-product approaches are used. These are user initiative approaches, where the user determines the definition of design rationale and how it should be documented. The user also decides when rationale retrieval happens from the rationale documentation. In a system initiative approach, the system determines the production of rationale and guides the capture process. It can also automatically update, maintain and retrieve rationale as needed. In reality, however, much more research is needed before system initiative approaches are feasible (Lee, 1997).

3.6.5 Rationale Representations

In order to use design rationale effectively, rationale representation is essential. There are multiple approaches to capture design rationale currently in use. In this paragraph the most used approaches are analysed. Below are examples of semiformal, argumentation based representations. According to Lee (1997), semiformal representations are best to help architects archive, retrieve and examine the reasons for their decisions.

The approaches are picked on the basis of their status as industry standards. The outcome of this analysis forms the basis for the eventual development of the approach. As analysis takes place, strengths and weaknesses are identified that serve as input for the design reasoning approach suggested by this project. The analysis done follows to an inductive approach. This means that the design reasoning approach takes shape as analysis progresses. The analysed data drives and forms the artefact.

3.6.5.1 Questions, Options and Criteria (QOC)

The Questions, Options and Criteria approach (QOC) is a semiformal notation to capture design rationale specifically (Maclean, Young, Bellotti & Moran, 1991). It is designed to support the argumentation process of discussing and evaluating options and alternatives in the architecture design process. An example can be seen in Figure 44. QOC diagram A (Maclean et al., 1991).

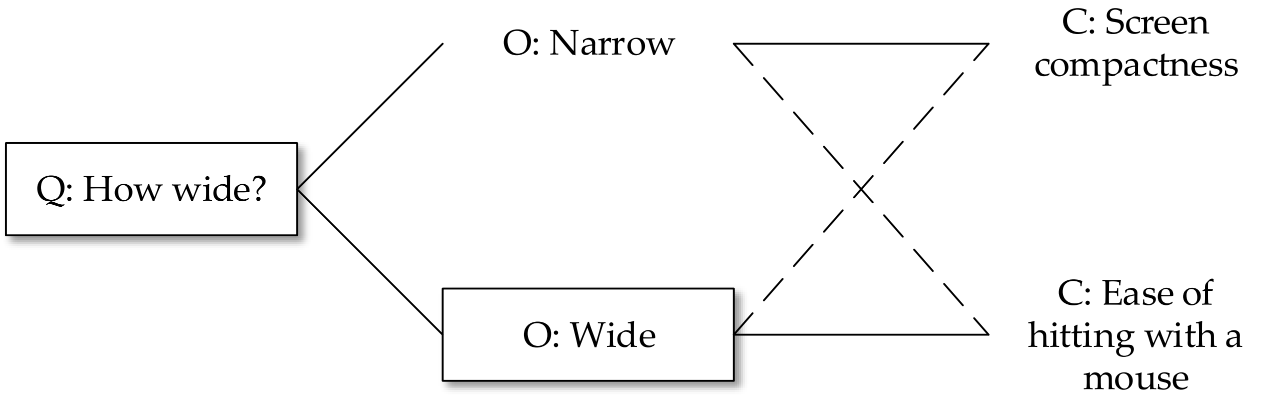


Figure 44. QOC diagram A (Maclean et al., 1991)

The Q, O and C indicate *questions*, *options* and *criteria*. *Questions* represent the key design issues where debate is needed, *options* represent the possible alternatives and *criteria* represent the measures or characteristics that define an alternative. The QOC diagram features small descriptions that detail the QOC elements for further clarification. The solid lines represent a positive assessment between options and criteria, whilst dotted lines represent a negative relationship. If an option is boxed, it means that that decision was final.

The QOC approach is suitable for simple, argumentation based design where decisions have clear options and criteria and elements can be generalized. It is designed to provide architects with a simple method that stimulates reasoning. However, it is not as specific as intricate designs can be. Architecture design is quite complex, commonly involving many options with their own criteria and assessments. Simple design decisions can easily become cluttered and unclear (see Figure 45. QOC diagram B (Maclean et al., 1991).

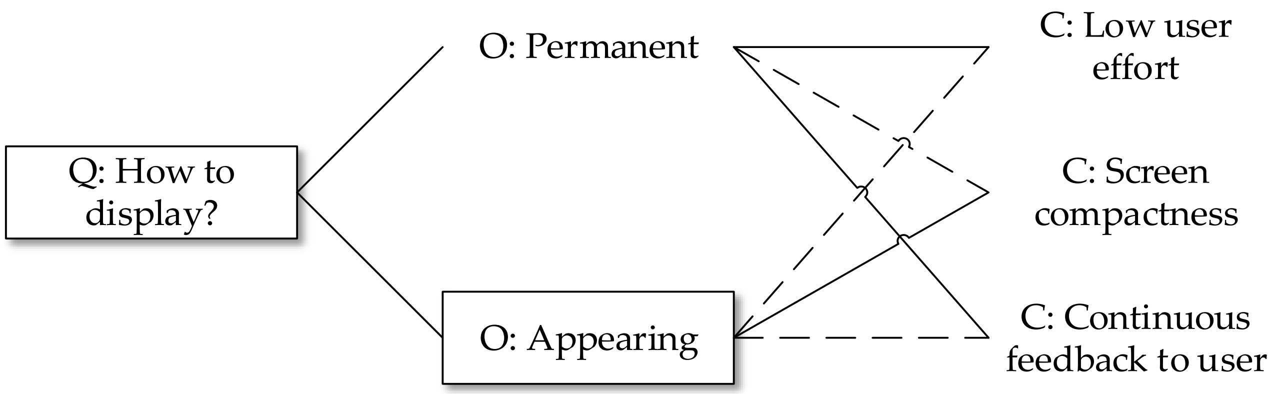


Figure 45. QOC diagram B (Maclean et al., 1991)

3.6.5.2 Issue-Based Information System (IBIS)

The Issue-Based Information System (IBIS) is an approach to utilize argumentation to deal with wicked problems (Werner & Rittel, 1970). As mentioned previously in the document, a wicked problem can be defined as an intricate issue where no standardized solutions exist. IBIS is designed to guide the issue solving process by way of identifying and arguing issues. An example IBIS graph can be seen in Figure 46. IBIS diagram.

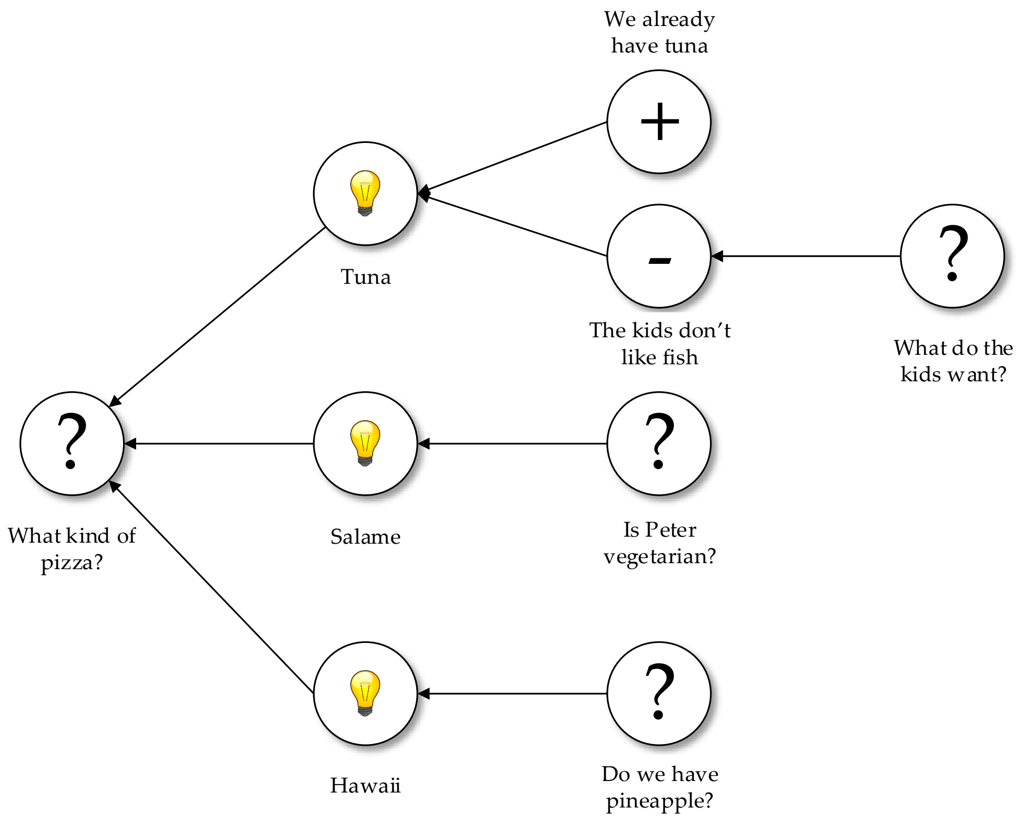


Figure 46. IBIS diagram

Similar to the QOC approach, IBIS features a tree graph of sorts where *questions* are central in the content of the graph. These questions are represented by the question mark icon. Each question has *ideas*; ideas represent possible solutions to the issue raised with the question. Each idea can feature more questions (that require more ideas subsequently) or can immediately feature a plus or minus icon. These icons represent the actual argument (rationale) that is either in favour (+) or against (-) the idea.

Similar to the QOC approach, IBIS is suitable for simple, argumentation based design yet can get cluttered when designs become increasingly complex and cluttered. The difference between IBIS and QOC is that QOC explicitly mentions options as design decisions whilst IBIS abstracts the process by its *idea* element. QOC options are explicit design options whilst IBIS ideas are not.

3.6.5.3 Decision Representation Language (DRL)

The Decision Representation Language (DRL) is a language for representing design rationale in the decision making process. It is developed by Lee (1989), and aims to represent design decisions, alternative options, goals and rationale for these decisions. An example can be seen in Figure 47. DRL Decision Graph Example

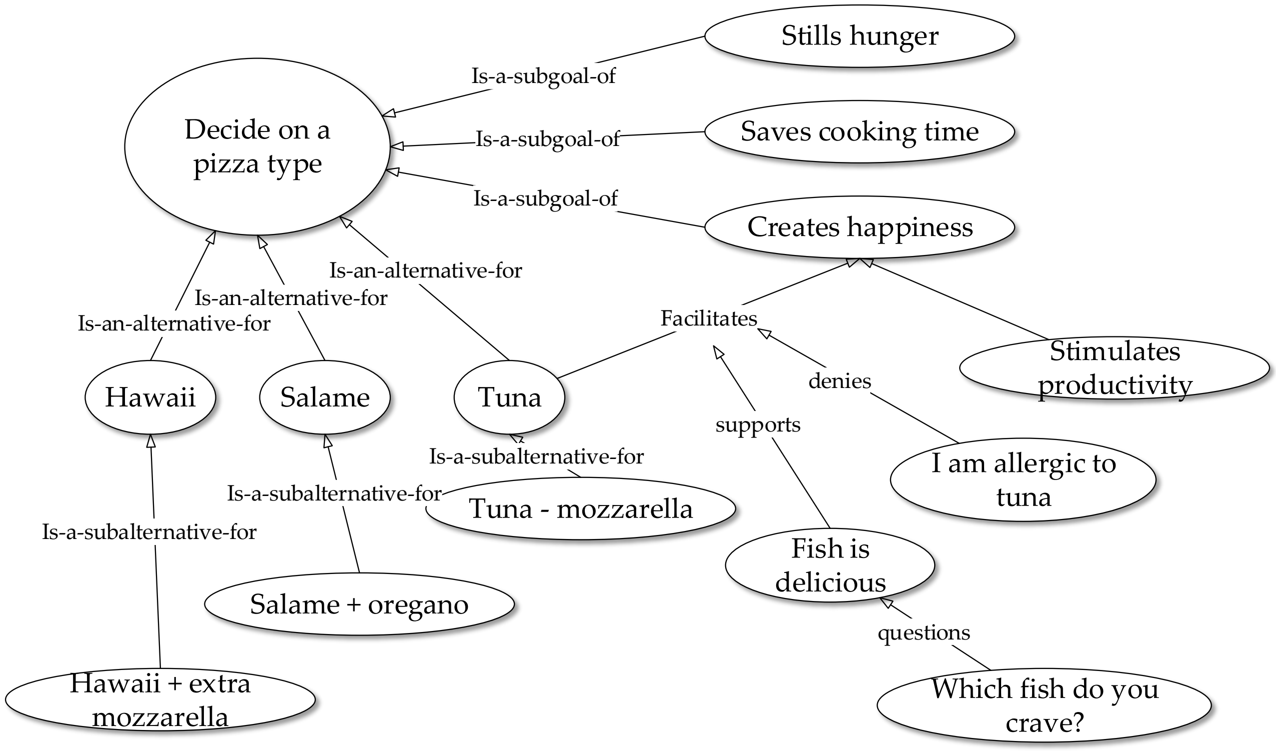


Figure 47. DRL Decision Graph Example

The DRL features a few elements that share commonalities with IBIS and QOC. The first element of the DRL is a *goal*, which represents the key aim for the design decision. A goal has certain related elements that provide context in how to achieve the goal, what *sub goals* must first be achieved and what *alternatives* are present. All these elements are supported by *claims*, which can either *support* or *deny* a goal. A claim represents a statement that has to be considered. A claim can also *answer* a question or *provide* an alternative.

Similar to IBIS and QOC, DRL is designed to provide architects with a model and vocabulary that stimulates design deliberation. The big difference, as opposed to IBIS and QOC, is the fact that DRL is designed to be used *asynchronously*. In other words, DRL is to be used after the design process itself has taken place. It is used to construct a rationale from historical records of design sessions. This is evident when analysing the model, as it gets complex quickly as the intricacy of designs increase.

3.6.5.4 Architecture Decision Description Template (ADDT)

The Architecture Decision Description Template (ADDT) is an approach to capturing design rationale by concentrating on important design decisions, first, and let the decisions drive the architecture. It was designed by Tyree & Akerman (2005) and is basically a template document that details what elements must be provided and documented when making architecture design decisions. The key elements that must be documented according to the ADDT are listed below, including their explanations, see Table 63. Architecture Design Description Template (Tyree & Akerman, 2005).

Architecture Decision Description Template	
• *Issue*	A description of the issue that is addressed by the design decision.
• *Decision*	The final decision that has been made in order to solve the issue.
• *Status*	The current state of the design decision. They can be pending, decided or approved.
• *Grouping*	Grouping of sets of design decisions in order to organize architecture decisions and combat complexity. Grouping can create order in the template.
• *Assumptions*	A list of assumptions that are present while making the design decision. These can be factors like cost, technology and people.
• *Constraints*	Additional factors that may constrain the design decision.
• *Positions*	A comprehensive list of alternative considerations or viable options that have been considered. These options have to be detailed enough to demonstrate that option has been carefully considered.
• *Argument*	The actual argumentation behind why a certain design decision has been made.
• *Implications*	All identified eventual implications the design decision may pose. Things like increased cost, reduced risk etc.
• *Related decisions*	Any related decisions that are relevant and associated with this design decision.
• *Related requirements*	The design decision has to be accountable and transparent. In this table the relevant and associated requirements that drove that decision have to be made explicit.
• *Related artefacts*	Any related IT artefacts, designs, systems, applications or documents that are relevant to the decision.
• *Related principles*	Here, any relevant principles related to the organization (or law) are to be listed. This shows transparency and adherence to additional regulations.
• *Notes*	Any additional relevant notes to the decision process.

Table 63. Architecture Design Description Template (Tyree & Akerman, 2005)

ADDT does not provide a design reasoning process, nor does it offer a comprehensive guide on how argumentation should be done. As opposed to IBIS and QOC, the ADDT provides a comprehensive list on the capture of design rationale knowledge. This information is often omitted or partially available through other approaches. Utilizing the ADDT, architects can clearly trace back what decisions has been made, why they have been made and any additional information that is relevant to the decision. The main advantage of the ADDT is the comprehensiveness and completeness of the design rationale that is captured using the template. However, the ADDT offers no reasoning or argumentation method that accompanies the template. Another disadvantage might be the resource intensive task of completing the whole template, as not every decision has to be completely filled in due to the simplicity of the decision.

3.6.5.5 Views and Beyond (VB)

The Views and Beyond (VB) approach is a collection of techniques in order to document software architecture (Clements, Garlan, Bass, Stafford, Nord, Ivers & Little, 2002). The VB approach emphasizes the use of views for documentation. Specifically, 3 views are important: the Module Viewpoint, Component and Connector Viewpoint and the Allocation viewpoint. Additionally, the VB approach demonstrates how the document for design rationale should be built. The main techniques elaborated on are categorized as follows:

- Finding out what stakeholders need – documentation has to be produced that is relevant to someone.
- Providing information to satisfy needs. Record design decisions according to specific views that are selected based on stakeholder needs.
- Checking the resulting documentation whether stakeholder's needs are satisfied.
- Packaging the information in an effective manner to stakeholders.

Similarly to the ADDT, the VB approach concentrates on providing designers with a complete template and guide on how the software architecture should be documented. A comprehensive list is available that elaborates on what elements comprise the design rationale, in what detail it should be captured and how these elements relate.

3.6.5.6 Architecture Rationale and Elements Linkage (AREL)

The Architecture Rationale and Elements Linkage (AREL) is a model that supports the capture of design rationale in the architecture design process. It is designed by Tang et al. (2007) and aims to provide architects with a system that allows for both capture and retrieval of rationale. The conceptual model can be seen in Figure 48. AREL Conceptual Model.

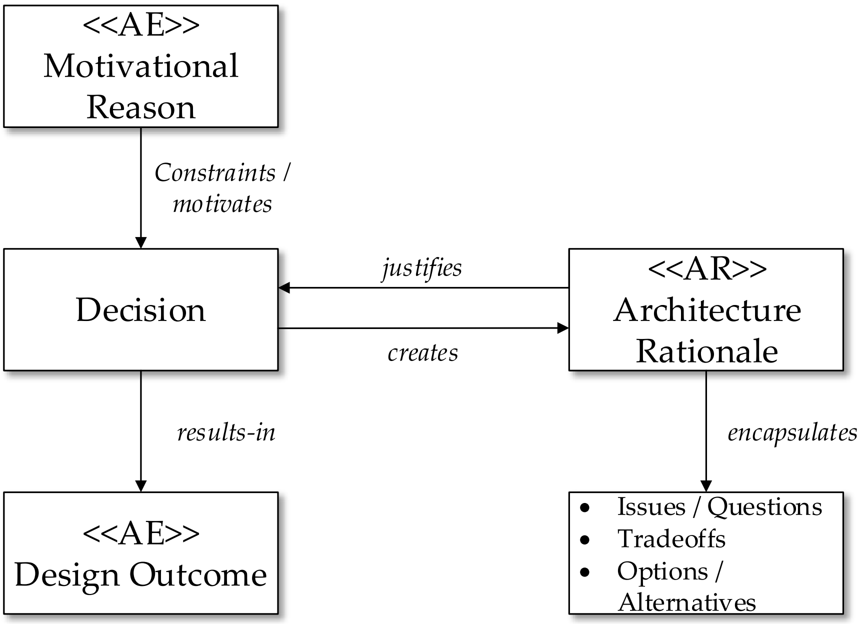


Figure 48. AREL Conceptual Model (Tang et al., 2007)

The conceptual model features elements and their relationships. The elements represent the information that needs to be present whilst the relationships define how the design decisions are made. The key distinction between AREL and other systems is that AREL is both a qualitative and quantitative rationale capture method with a retrieval method as opposed to IBIS, QOC and DRL which provide the designer with a language that helps the deliberation process and allows for textual rationale capture.

3.6.6 Rationale Methods in Practice

When viewing the current state of the art it seems evident there is no standard in the industry, both in terms of design reasoning as a deliberation process and design rationale capture. However, there is consensus that these concepts are of importance in the industry (Tang et al., 2006) and that practitioners are beginning to recognize it (Tang & van Vliet, 2009). However, researchers agree that, despite the growing recognition, the concepts of design reasoning and rationale capture both are underutilized in the current state of the art (Regli et al., 2000; Tang et al., 2006; Verries et al., 2008).

Regli et al. (2000) argue that there are three types of challenges that prevent the widespread use of design reasoning and rationale capture: *representational* challenges, *capture* challenges and *retrieval* challenges. In terms of representation, designers need to find the best method that allows for easy input, effective view and activeness. Also, the capture process needs to contain the least amount of overhead possible so that it minimizes the interference with the design process. Lastly, efficient retrieval of design knowledge needs to be available without much resources spent of navigation.

Tang (2007) mentions design rationale methods have laid a solid foundation for the use of design reasoning in the industry. However, there are still certain elements that prevent design reasoning from widespread use. Tang provides 6 key criteria that contribute towards the successful implementation of design rationale in the industry that should be strived towards in order for the implementation and industry adoption to be achieved:

- *Effective Capture of Design Rationale* – argumentation-based models should capture both the reasoning and the rationale argumentation structure. This argumentation structure is time consuming and difficult to trace and should be simplified without losing design rationale depth and context.

- *Effective Communication of Design Rationale* – design rationale has tendencies to be overrepresented and cluttered with information where that is not necessary and underrepresented when more information should be present. The trick is guiding the documentation of design rationale in a way that is effective and efficient with the least amount of overhead as possible.
- *Design Artefact Focus* – design rationale should specify design elements and artefacts for improved evaluation, maintenance and rationale effectiveness. Rationale should not only consist of design decisions and their justifications,
- Traceability and Impact Analysis – design rationale should be structured in such a way that supports traceability and impact analysis. One of the major reasons for design rationale is its usefulness when maintenance or change activities take place. Rationale should point towards related elements for impact analysis to be feasible.
- Comprehensive Design Rationale – design rationale should be comprehensive in terms of flexibility to support different types of rationale.
- Common Tool Support – design rationale should be supported by software tools as much as possible to reduce the amount of overhead.

4 METHOD ASSOCIATION APPROACH

This appendix chapter offers the Method Association Approach in its entirety. The approach was used to design and construct the artefact that would become the Rationale Capture Cycle.

4.1 Project Situations

The MAA method prescribes identifying project situations in order to the construction of the situational method to be tuned to a specific situation. In terms of architecture design, 3 distinct project situations have been identified during the practitioner interviews. These situations are: creating a Project Starting Architecture (PSA), Solution Architecture (SA), Goal Architecture (GA) and Agile Architecture (AA), as named by Sogeti practitioners. Ultimately, however, these project situations all contain design decisions that require rationale and do not differ too wildly from each other. Therefore, the resulting artefact does not vary on a project basis. For the sake of completeness, however, the project situations are still included and defined.

4.1.1 Project Starting Architecture (PSA)

The PSA is created at the beginning of a project in order to identify risks, guide development and define borders of design and creation. This architecture design does not in any way represent the final architecture for design and implementation but is meant to facilitate change, decision-making, identify principles and regulations, analyse potential implications and risks and define project borders and constraints. In some situations, this architecture is known as the Architecture Definition Document (ADD).

4.1.2 Solution Architecture (SA)

The SA, however, represents the architecture that details the final design. It differs from a PSA due to its different scope. The PSA defines project constraints and borders whilst the SA describes the final solution design and is an end result of a specific solution or system in its context.

4.1.3 Goal Architecture (GA)

The GA describes the wished end result, or SOLL-situation. It also describes the required steps that need to be performed in order to get there. A SA can be a GA, if the final goal is to provide a system description to solve an issue. However, usually a GA has a wider scope. In some occasions the GA also contains an IST-situation architecture to show contrast and comparison.

4.1.4 Agile Architecture (AA)

The last project situations stems mostly from the new way of working that is currently heavily adopted. The Agile methodology and scrumming causes architecture to be seen as more of a process instead of a product. An architecture is not necessarily an end result of a process. In this project situation, the architecture is a consistent document and model that is continuously updated.

4.2 Feature Groupings

Luinenburg et al. (2008) defines feature groups as a set of design requirements that are grouped together. These requirements are necessary as they provide the criteria with which existing methods can be compared. In other words, the feature groups are sets of business requirements that influence design and form the means by which suitable existing methods can be selected and compared. The relevant requirements are identified during the interview phase by Sogeti experts and through literature study. The feature groups consist of Sogeti's professionals' expressed desires of a design reasoning / rationale artefact and best practices from relevant scientific studies.

The generic design rationales suggested by Tang et al. (2006) are taken as a baseline due to its comprehensiveness. The rationales are compared to suggestions from other researchers like Lee & Kai (1991), Burge & Brown (1998), Bass et al. (2003), Tyree & Akerman (2005) and others to get a sense of best practice and uniformity in terminology. These design requirements are then validated and discussed during validation sessions with architecture practitioners and with the thesis supervisor. Some of the design requirements are coupled together due to simplicity and overlap in definition. Other design requirements are named differently to enhance readability.

The resulting design requirements that serve as selection and comparison concepts are the following: *problem identification, constraint identification, assumption recognition, option listing, benefit listing, weakness listing, trade-off analysis* and *risk analysis. Reflection & evaluation* is added to improve the reflection and evaluation aspect of the process. As these concepts all comprise the overarching feature of design rationale capture, no further divisions are made in terms of feature groupings.

These concepts are mapped to existing candidate methods in the association table in paragraph 4.5, Association. In the next paragraph the potential candidate methods are identified, as the MAA approach prescribes.

4.3 Candidate Methods

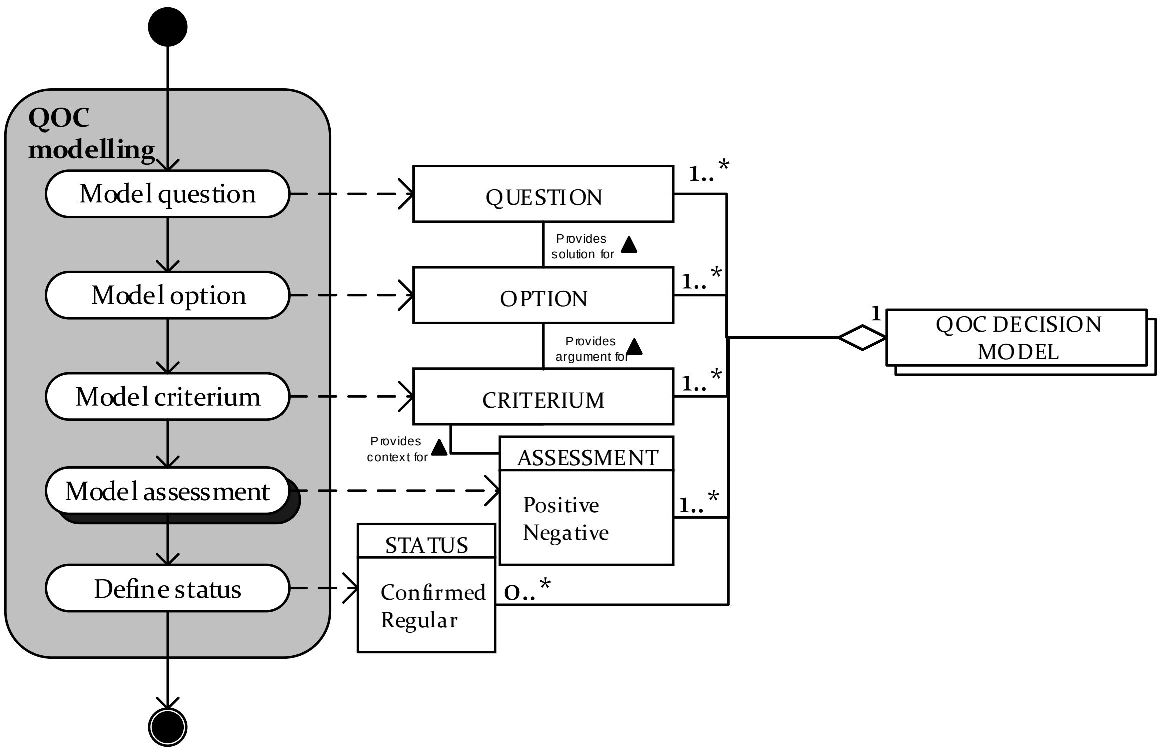
The existing candidate methods for rationale capture were already identified during the literature review and are as follows: Questions, Options & Criteria (QOC), Issue Based Information System (IBIS), Decision Representation Language (DRL), Architecture Decision Description Template (ADDT), Views & Beyond (VB) and Architecture Rationale & Elements Linkage (AREL). Also, considering the research field is new, some of the candidate methods are not methods at all. Rather, they are suggestions offered by researchers on how to tackle the capture of rationale. These methods are chosen due to their well-known status in the industry and are verified by the thesis supervisor.

For an elaboration on these methods, their history, uses, goals and a visual model refer to paragraph 3.6.5, Rationale Representations in appendix 3, Literature Review. These models are translated into the PDD notation for insertion into the method base, as described in the next step.

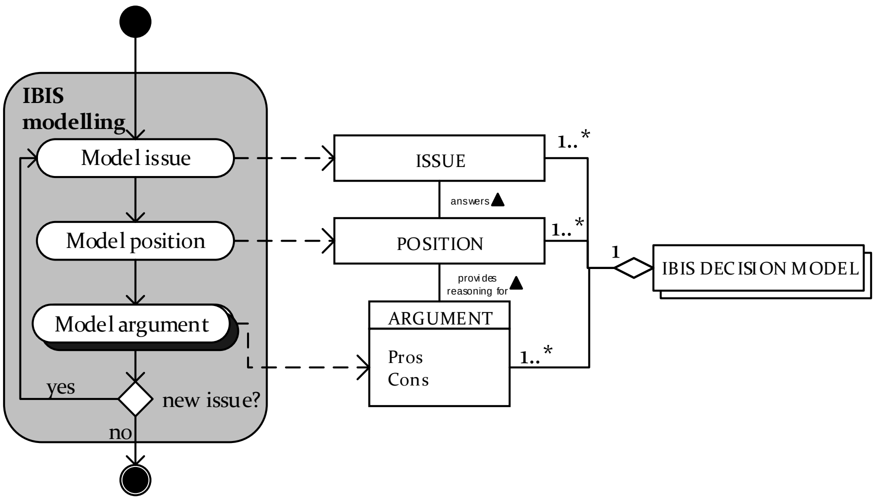
4.4 Method Base

The next step of the MAA method prescribes that the relevant method fragments of the selected candidate methods are modelled. The modelling technique used to model these methods is the Process-Deliverable Diagram (PDD), a notation based on UML standards that describes both the process and product side of a method. This notation is prescribed by the MAA method. For each candidate method a meta-model is created using the PDD notation. All PDD models together comprise the Method Base. The full PDD models can be found in the following paragraphs.

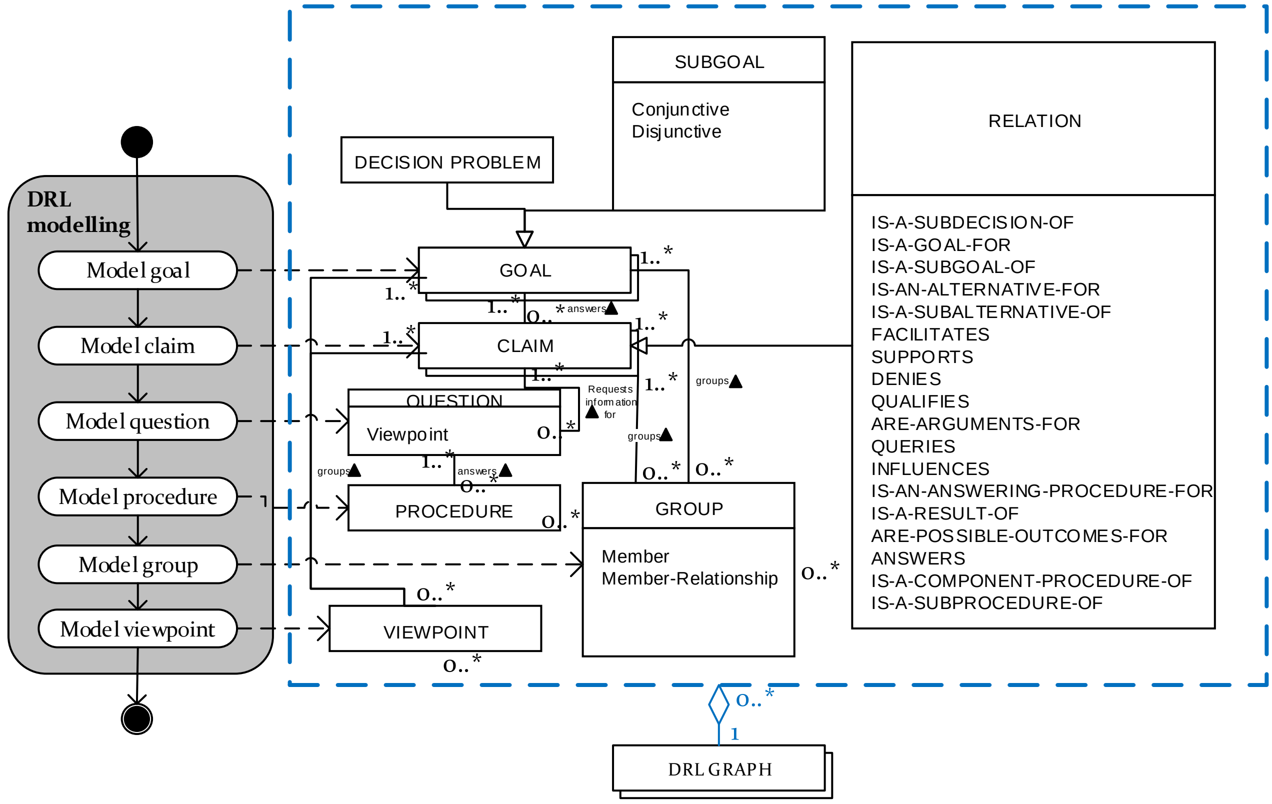
4.4.1 QOC



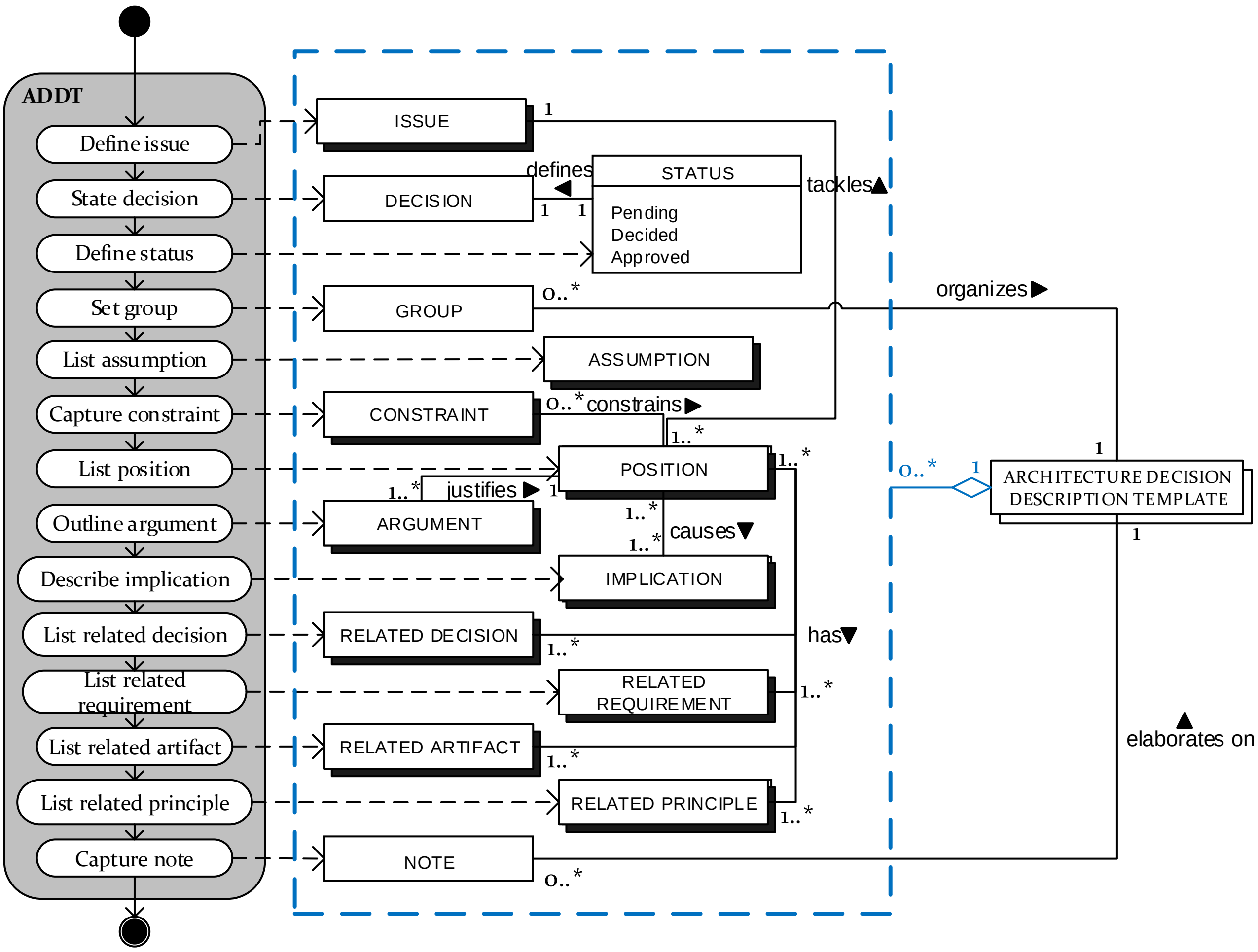
4.4.2 IBIS



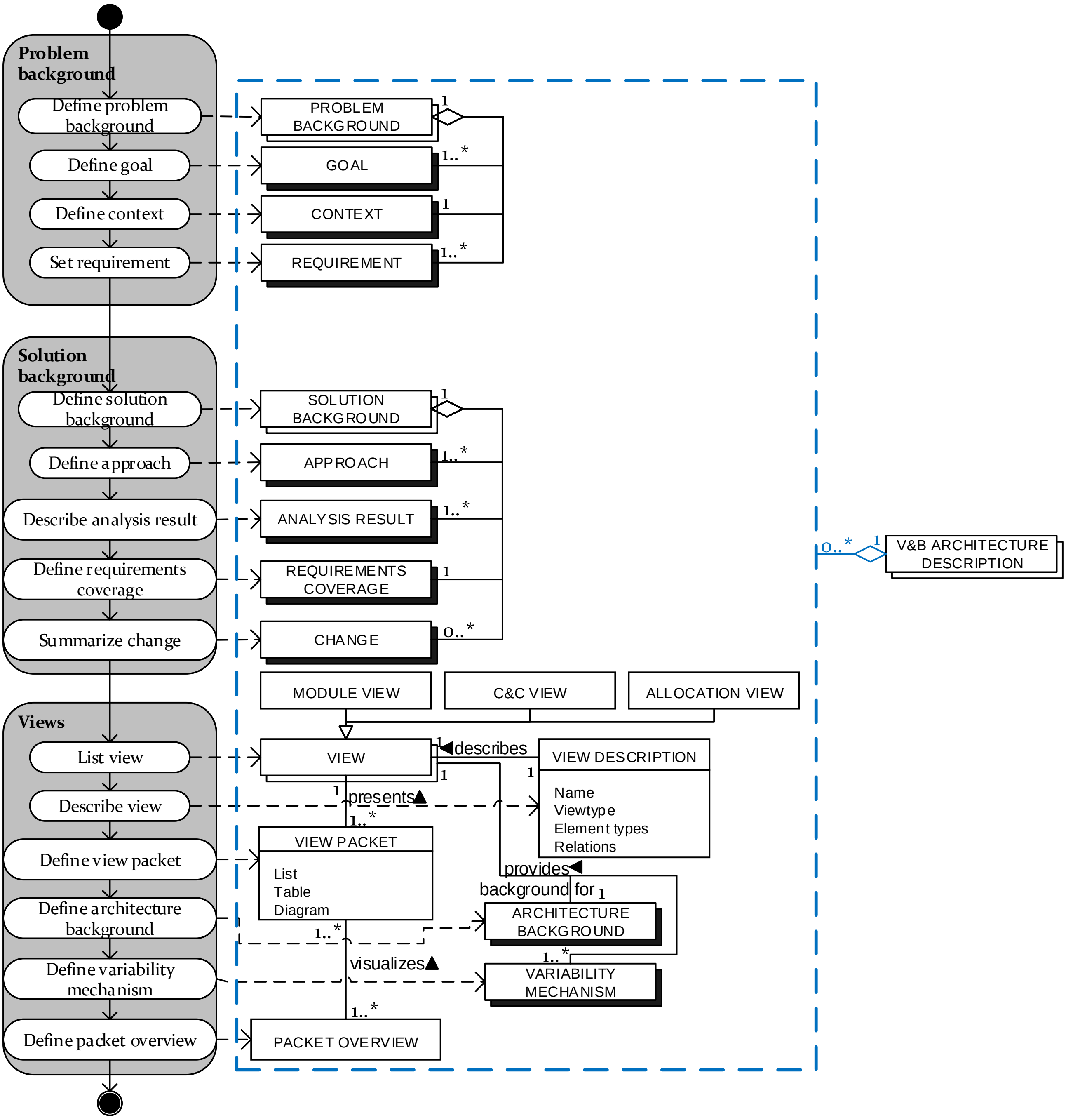
4.4.3 DRL



4.4.4 ADDT

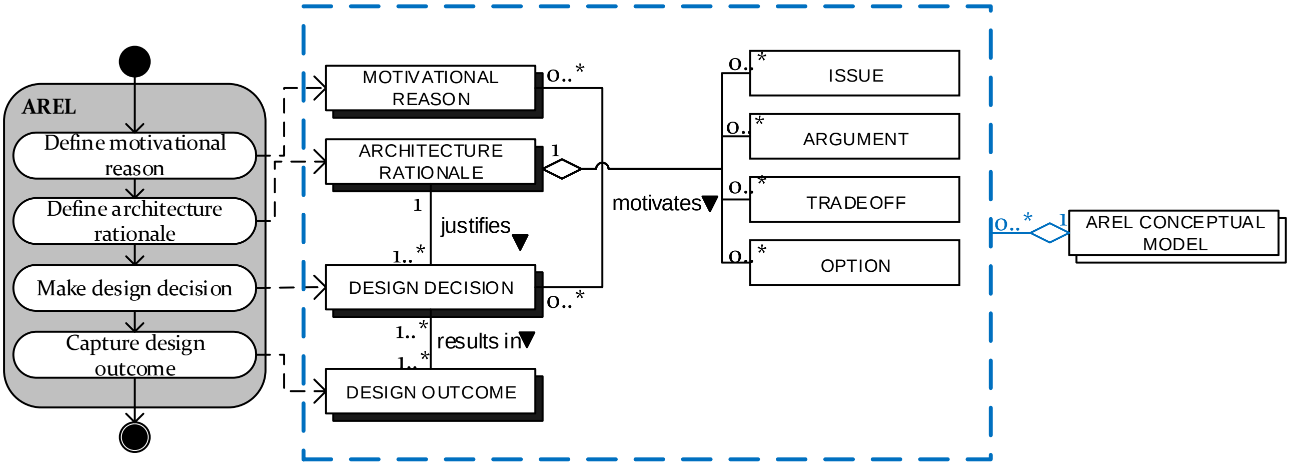


4.4.5 VB



Problem background
Define problem background
Define goal
Define context
Set requirement
Solution background
Define solution background
Define approach
Describe analysis result
Define requirements coverage
Summarize change
Views
List view
Describe view
Define view packet
Define architecture background
Define variability mechanism
Define packet overview
PROBLEM BACKGROUND
GOAL
CONTEXT
REQUIREMENT
SOLUTION BACKGROUND
APPROACH
ANALYSIS RESULT
REQUIREMENTS COVERAGE
CHANGE
MODULE VIEW
C&C VIEW
ALLOCATION VIEW
VIEW
describes
VIEW DESCRIPTION
Name
Viewtype
Element types
Relations
presents
VIEW PACKET
List
Table
Diagram
provides background for
ARCHITECTURE BACKGROUND
visualizes
VARIABILITY MECHANISM
PACKET OVERVIEW
V&B ARCHITECTURE DESCRIPTION
1
1..*
0..*

4.4.6 AREL



4.5 Association Table

In Figure 49. MAA Association Table the design requirements are mapped against the method fragments of the method base. The method fragments are derived from modelling the PDD's of the methods in the previous chapter. The 'X' in the association table represents that that specific method fragment is needed in order to incorporate a design requirement. In other words, the table allows an overview of the desired design requirements and shows to what extent they are addressed by existing method fragments. This way, the existing method fragments from the method base can be incorporated and assembled into a situational method that is based on the desired design requirements.

Association table		QOC					IBIS				DRL					
Methods		Questions	Options	Criteria	Assessment	Status	Questions	Ideas	Pros	Cons	Goal	Claim	Relation	Question	Procedure	Group
Design Requirements – Rationale capture		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Problem identification	1	X					X				X					
Constraint identification	2															
Assumption recognition	3											X				
Option listing	4		X					X				X				
Benefit listing	5				X				X				X			
Weakness listing	6				X					X			X			
Trade-off analysis	7															
Risk analysis	8				X					X			X			
Reflection & evaluation	9															

Association table		ADDT														V&B									AREL					
Methods		Issue	Decision	Status	Group	Assumptions	Constraints	Positions	Argument	Implications	Related decisions	Related requirements	Related artifacts	Related principles	Notes	Problem	System overview	Goals and context	Driving Requirements	Solution	Architectural approaches	Analysis results	Requirements converage	Changes	Decision	Outcome	Reason	Issues / questions	Tradeoffs	Options / alternatives
Design Requirements – Rationale capture		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Problem identification	1	X														X												X		
Constraint identification	2						X												X											
Assumption recognition	3					X													X				X							
Option listing	4							X												X					X					X
Benefit listing	5								X													X				X	X			
Weakness listing	6								X													X				X	X			
Trade-off analysis	7										X																		X	
Risk analysis	8									X												X				X				
Reflection & evaluation	9							X																X						X

Figure 49. MAA Association Table

4.6 Method Assembly

In order to assemble the preliminary artefact, an association strategy is needed. In this case the feature group strategy is chosen. This strategy starts from the design requirements and looks at the mapping and coverage by the methods from the method base. Here, insight is gained into which relevant method fragments cover the design requirements. Also, potential differences and similarities in method fragments can be analysed. Based on the coverage of design requirements a situational method will be assembled.

4.7 Preliminary Artefact and Design Philosophy

The method is assembled using the relevant method fragments from the method base, producing the following preliminary artefact:

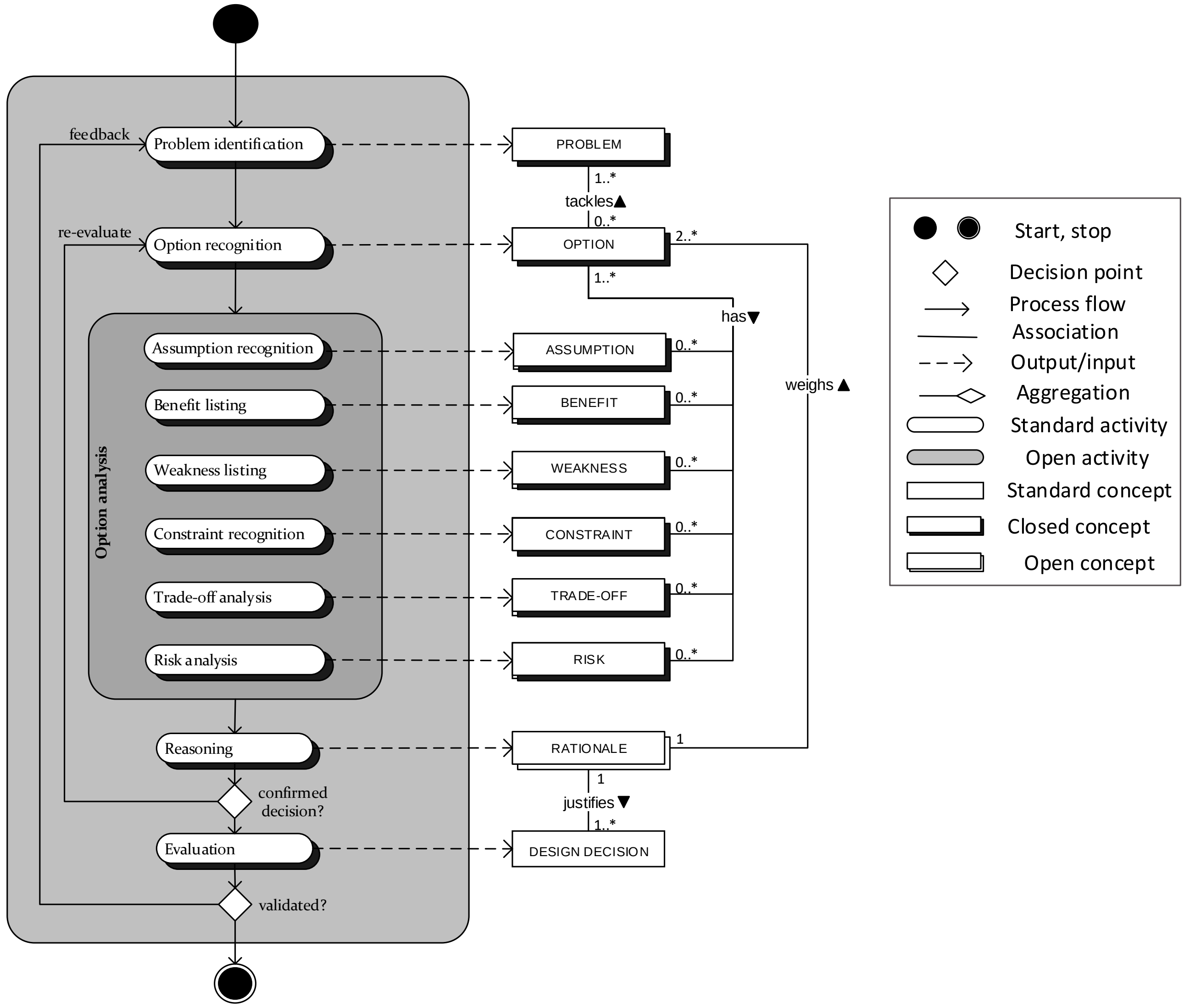


Figure 50. Rationale capture PDD

The artefact adheres to the PDD notation, as per MAA prescription. The artefact is consciously designed around a high abstraction level model. The reason for this is that architecture design is very context-specific (Poort, 2014). Fitting exact design processes can result in problems when various different domains and projects arise. Therefore the artefact should not be seen as a product but rather as a process. The artefact does not attempt to define what rationale should be, but rather provide handles that serve as a best practice. Its elements can be cherry picked and applied ad-hoc, as this provides the best basis for use and application in as many scenarios as possible. During literature review and the interview phase it was found that in order to warrant widespread use and guarantee simplicity, such a design philosophy should be chosen. It is nigh impossible to define and restrict what rationale should be due to the dynamic nature of design rationale itself and the young domain of architecture design. Also, there are too many different project and problem areas that force the artefact to remain abstract.

4.8 Artefact Validation

The main validation will take place during the experiment phase, where in 3 separate moments the participants will be able to express their feedback regarding the artefact. Two times, during the sessions themselves, the experiment will feature a survey where the participants can communicate their feedback and input. The third moment will be a discussion session where additional validation will take place. Artefact validation will be done by 10 architects from Sogeti, all currently practicing various forms of architecture (enterprise, information, software) with varying levels of expertise and age at various clients in the Netherlands. The questions asked in the survey can be seen in the next appendices where the case will be elaborated on. The end of the case features the survey.

5 EXPERIMENT CASE MOMENT 1

The following pages represent the experiment cases for both moment and moment 2. The test group and control group variants are similar, however the control group variant does not have an instruction for the Rationale Capture Cycle in moment 1. Also, in moment 2 the control group variant does not have rationale present. These were the independent variables that were manipulated. Due to restriction of page sizes, however, only the test group cases are added as they contain all the information.

Gebruik van de 'Rationale Capture Cycle'

Op de volgende pagina staat een casus waarbij je een aantal architectuurmodellen dient te ontwerpen. Het is van belang om tijdens het ontwerpen je ontwerpbeslissingen van rationale (verantwoording) te voorzien. Dit zal op de volgende pagina's verder worden toegelicht. Hiervoor krijg je een aparte documentatie template digitaal opgestuurd.

Een van de doelen van het onderzoek is het in de praktijk testen van het ontworpen instrument. Het instrument neemt de vorm aan van een model ten behoeve van het gebruiksgemak. Tijdens deze sessie zal het model worden getest. Geen zorgen, hiervoor hoef je vrijwel geen extra input te leveren. Sterker nog, het is ontworpen om je tijdens het ontwerpproces te helpen!

Om je in het ontwerpproces te ondersteunen heb je een A4 met de '*Rationale Capture Cycle*' als model gekregen. Je moet dit zien als een instrument dat je helpt bij de totstandkoming van je ontwerpbeslissing. Het model geeft je inzicht in welke elementen belangrijk zijn wanneer je een beslissingspunt tegenkomt in je ontwerpproces. Het model is in essentie een visuele weergave van de belangrijkste beredeneringsaspecten die samen de kern van je beredeneringsproces vormen.

Het model is dus ontworpen om je handvaten te geven bij het maken en evalueren van ontwerpbeslissingen, maar geeft je ook richting in de documentatie hiervan. Belangrijk is dat je het model niet als vaste procedure ziet, maar als hulpmiddel tijdens je reguliere ontwerpproces. Aan het einde van de casus krijg je de gelegenheid om het model van terugkoppeling te voorzien.

Hieronder volgt eerst de casus. Lees deze eerst goed door. Daarna zal er een set aan wensen duidelijk worden gemaakt. Ook zal de casus duidelijk maken wat je uiteindelijk dient op te leveren, en in welke format.

Om praktische redenen heb je **2 uur** de tijd om de casus te voltooien. Wanneer je, volgens jou, geen bijdrage meer kan leveren of waarde kan toevoegen ben je natuurlijk vrij om je werk eerder in te leveren.

Succes!

De Utrecht Bank casus

Introductie

In de komende twee uur word je gevraagd architectuurmodellen te ontwerpen op basis van een aantal variabelen. Lees eerst goed de casus door. Op de volgende pagina staan verdere instructies ten behoeve van de output. Belangrijk: vul de **starttijd** bovenin bij de vorige pagina voor het lezen van de casus. Vergeet ook niet de **eindtijd** in te vullen wanneer je klaar bent.

Modernisering van de dienstverlening van de Utrecht Bank

De Utrecht Bank is oorspronkelijk begonnen als exclusieve bank voor welgestelde particulieren. Deze klanten komen uit families die al generaties lang bij de bank voor hun geldzaken komen. De klanten weten wat ze aan de bank hebben en waarderen de bank dan ook voornamelijk om haar voorspelbaarheid en betrouwbaarheid. Het doorgroeien naar de zakelijke markt en het verruimen van het dienstenpakket naar verzekeren heeft de bank-tak tot nu toe redelijk ongemoeid gelaten. De bank heeft zijn exclusieve imago door slim label-management weten te behouden. Er begint echter een omslag te komen in het klantenbestand, doordat steeds meer ''nieuwe rijken'' zich tot de bank wenden. Een significant deel van deze nieuwe klanten zijn aan de jongere kant en hebben met start-ups in de app-industrie vermogen weten te ontwikkelen. Ze kiezen voor de Utrecht Bank vanwege de goede naam en reputatie. Deze nieuwe klanten brengen duidelijk nieuwe wensen met zich mee ten opzichte van de modernisering van de dienstverlening. De Utrecht Bank heeft zelf niet veel ervaring met moderne technologie want haar 'core business' draait om ouderwets bankieren.
In het kader van deze nieuwe eisen aan de Utrecht Bank, besluit de directeur dat het tijd is voor een goed informatieplan. De CIO krijgt hiertoe opdracht vanuit het bestuur. Hij formeert meteen een projectteam van de vijf beste mensen uit zijn afdeling en geeft hen de opdracht een informatieplan op te stellen. Het projectteam gaat direct aan de slag. Ze beginnen met het uitpluizen van de beleidsstukken en jaarplannen van de verschillende onderdelen van de bank. Vervolgens brengen ze de bestaande situatie op het gebied van de informatievoorziening in kaart: welke informatiesystemen zijn er in gebruik, wat doen die informatiesystemen en hoe hangen ze met elkaar samen? Als derde stap proberen ze in kaart te brengen wat de nieuwe gebruikers nou precies voor behoeften hebben. Met behulp van deze informatie bepalen ze een ruwe richting en visie om aan de evoluerende behoeften te voldoen.

Opdracht

Jij bent een van de aangewezen architecten in het projectteam en dient het informatieplan van een aantal doelarchitectuurmodellen te voorzien. Van jou wordt verwacht dat je de vertaalslag maakt naar een werkbare architectuur. De CIO benadrukt dat er niet te veel gekeken moet worden naar het huidige landschap, maar dat er meer een ideaalscenario moet worden geschetst. Het bestuur is namelijk bereid forse veranderingen door te voeren in haar organisatie en is benieuwd naar je creatieve visie en schets van een ideaalscenario. De volgende stap zal dan bestaan uit het vergelijken van de as-is en jouw toekomstmodellen om zodanig werkbare projecten te onderkennen.

Requirements
De volgende algemene requirements zijn je opgelegd vanuit het bestuur:

1. De toekomstige omgeving dient de overgang van decentraal naar centraal duidelijk te waarborgen. Momenteel staan de verscheidene databases (particulier, zakelijk, hypotheek, verzekeringen etc.) decentraal op lokale servers. Wanneer er data wordt opgevraagd door de web portal krijgt de hoofdserver een verzoek voor informatie die vervolgens één voor één het verzoek door vertaald naar de desbetreffende databases. Wanneer dit proces is voltooid aggregeert de hoofdserver de data en creëert deze een view op basis van alle opgegeven velden. Het bestuur krijgt al jaren klachten over de snelheid van de web portal en is overtuigd dat dit beter kan. Je wordt vrijgelaten in je oplossing, maar security is uiteraard een primaire zorg.

2. In het verleden konden gebruikers via een internetportal enkel hun rekening inzien en bedragen overschrijven, maar wij willen dit uitbreiden. Gebruikers moeten via één centrale portal nu ook hun hypotheek en verzekeringen kunnen inzien, bijvoorbeeld. De Utrecht Bank levert naast hypotheken en verzekeringen ook andere particuliere leningen, beleggingsmogelijkheden en spaarrekeningen. Daarnaast heeft de Utrecht Bank voor al deze particuliere diensten ook de 'zakelijke' variant. Gebruikers moeten nu naadloos tussen hun particuliere en zakelijke rekeningen kunnen 'switchen'.

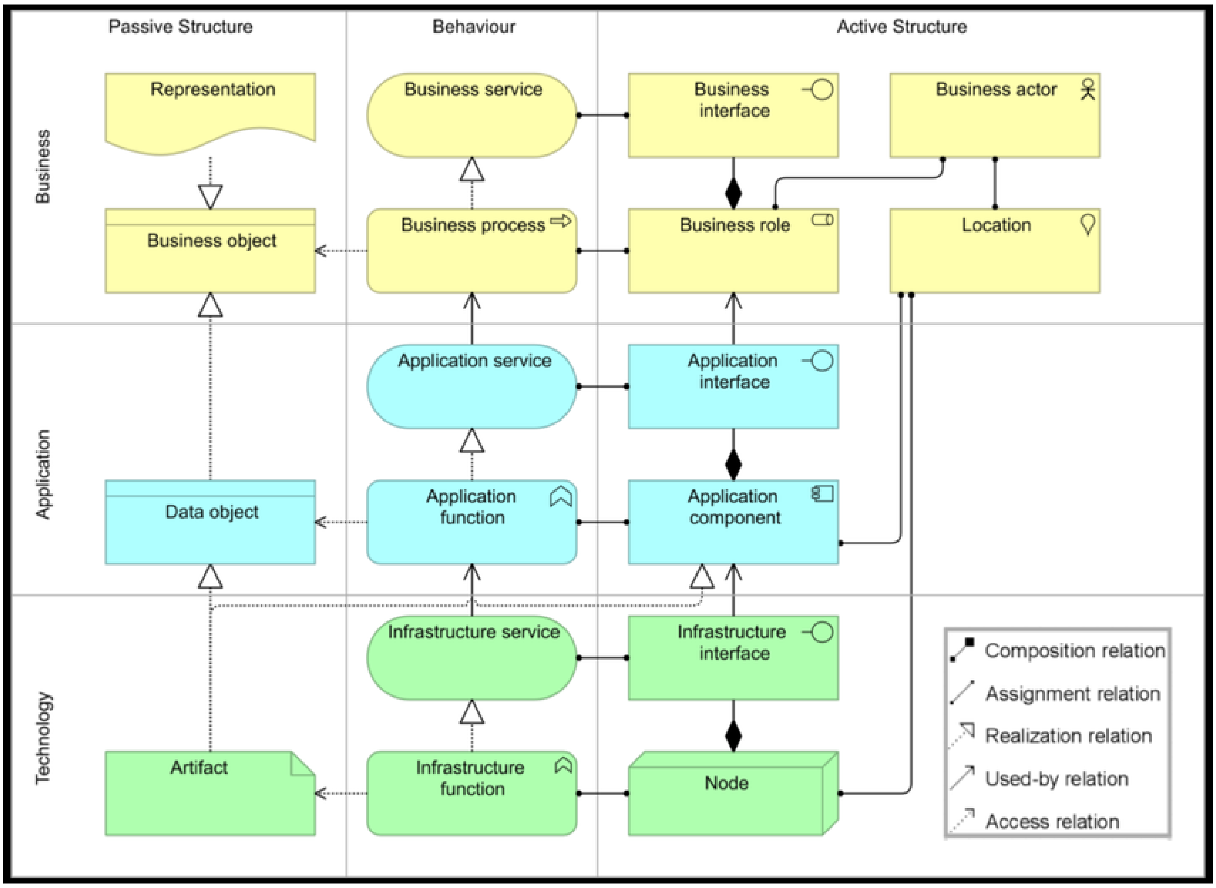
3. Gebruikers moeten via de centrale portal nu ook administratieve verzoeken en aanvragen (creditcards, kapotte betaalpassen, beheerwijzigingen) kunnen voltooien zonder dat wij daar handmatige handelingen voor hoeven te verrichten. Vanuit onze dienstverlening moet, waar nodig, automatische brieven met eventuele producten naar de klanten thuis worden gestuurd.

4. Wij willen graag dat de nieuwe gebruikers in een moderne omgeving toegang hebben tot hun bankzaken. De hiervoor genoemde portal dient via de smartphone en tablet ook toegankelijk te zijn, ook wanneer deze niet via Wi-Fi is verbonden. Daarnaast willen we dat gebruikers betaalverzoeken kunnen creëren en deze digitaal kunnen doorsturen naar derden.

Het bestuur is open voor nieuwe ideeën en geeft je de ruimte om met oplossingen te komen die afwijken van de reguliere gang van zaken.

Output
Om overeenstemming omtrent de gewenste producten te garanderen verwacht de CIO dat je je aan de volgende globale **principes** houdt:

- Je dient verduidelijking op te leveren omtrent de wensen van het bestuur. Je wordt vrijgelaten in de keuze voor oplossingen, zelfs als deze (gedeeltelijk) buitenshuis moeten worden gerealiseerd. Het rekening houden met de Utrecht Bank en haar strengths en weaknesses is wel van belang, evenals het kostenplaatje. De CIO laat de inschatting daarvan aan jou over.
- Je hoeft niet in (technisch) detail te treden, als de benodigde informatiestromen en –behoeften maar in kaart worden gebracht.
- De CIO ziet graag dat je conform de standaarden in de organisatie werkt en vraagt je ArchiMate te hanteren. Hij geeft je het volgende metamodel van ArchiMate 3.0 als handvat:



Figuur 1. ArchiMate 3.0 metamodel

Er wordt verwacht dat je de volgende **producten** oplevert:

1. Kies **2 bedrijfsprocessen** uit de casus en modelleer de volgende elementen per gekozen proces:
    a. Een model van het bedrijfsproces zelf en welke diensten deze levert.
    b. Een applicatiefunctiemodel en diens relatie naar het bedrijfsproces met betrekking tot de geleverde diensten.
    c. Een infrastructuurfunctiemodel, inclusief welke diensten deze moet leveren.
    d. Uiteindelijk dienen er dus **2 processen** opgeleverd te worden. Inzichtelijk moet worden gemaakt hoe de processen werken, welke diensten ze leveren, welke applicatiediensten er nodig zijn om het proces te bewerkstelligen en welke infrastructuur ondersteuning vereist is.
2. De CIO vindt het **cruciaal** dat je je gemaakte ontwerpbeslissingen goed **documenteert**. De CIO is van mening dat een architectuurmodel zonder context veel minder waarde heeft. Daarom wil hij wil dat je je beslissingen van rationale voorziet. Daarvoor heeft hij een **architectuur documentatie template** opgesteld. Deze is te vinden in je **mailbox** en dient bijgehouden te worden terwijl je je modellen ontwerpt. Uiteindelijk wil hij deze ook in zijn mailbox weer ontvangen.

Je bent vrij om te kiezen welke 2 processen je van belang acht, mits je zo volledig mogelijk de wensen van het bestuur aanpakt. Daarnaast ben je ook vrij om te kiezen welke elementen je precies modelleert, mits je conform de **ArchiMate** standaard werkt.
De Utrecht Bank wil uniformiteit onder haar werknemers en het werk dat ze leveren en verwacht dat je producten oplevert in de **ArchiMate** notatie. De Utrecht Bank heeft hier zelf geen licentie voor beschikbaar, maar gebruikt **Archi** als gratis tool. Daarnaast ontvangt de Utrecht Bank graag de digitale modellen en de documentatietemplate via mail naar het onderstaande adres. Vergeet ook niet de eindtijd in te vullen.

Hoogachtend,
Pim de Jong
p.s.h.dejong@students.uu.nl / pim.de.jong@sogeti.com
De Utrecht Bank

Mocht je **Archi** nog niet ter voorbereiding hebben geïnstalleerd, volg de volgende instructies:

1. Ga naar https://www.archimatetool.com/download.
2. Klik op 'Windows Installer' en download de tool.
3. Open het installatiebestand en installeer de tool.
4. Start het programma.

Survey

De volgende vragen gaan over het gebruik maken van het model tijdens het maken van de casus. De vragen gaan specifiek over in hoeverre het model gebruiksvriendelijk, laagdrempelig, effectief en behulpzaam was tijdens het architectuurontwerpproces.

1. Was het model van toegevoegde waarde? Hielp het bij het bedenken van oplossingen, risico's, aannames etc. waar je normaliter wellicht niet aan zou denken?

____
____
____

2. Was het model effectief? Hielp het model bij de totstandkoming van een ontwerpbeslissing of bij het documenteren van je architectuur?

____
____
____

3. Was het model leesbaar? Was het direct duidelijk wat er met de termen werd bedoeld? Leidde de syntax van het model af van de inhoud?

____
____
____

4. Was het model eenvoudig in het gebruik? Was het direct duidelijk hoe het model gebruikt kon worden? Was het model intuïtief?

____

____
____

5. Voelde het model natuurlijk? Kon je het model eenvoudig gebruiken tijdens je ontwerpproces? Voelde het model als ontwrichtend of storend aan?

____
____
____

6. Ik wil het volgende nog graag kwijt over het model, de casus of het onderzoek:

____
____
____

7. Ik heb de volgende activiteiten uitgevoerd:
   a. Mijn **naam**, **starttijd** en **eindtijd** zijn ingevuld.
   b. Ik heb de modellen **geëxporteerd** en **opgestuurd**.
   c. Ik heb de architectuur documentatie template **bijgehouden** tijdens het ontwerpproces en deze **opgestuurd**.

Einde sessie 1

Bedankt voor de participatie! Je levert onmisbare input voor het vorderen van het onderzoek. Nadat ik alle resultaten binnen heb kunnen jullie een mail van mij verwachten met daarin de vraag de sessie af te ronden door middel van een assessment. Hier krijgen jullie een **drietal** oplossingen, gemaakt door jullie collega's, en dienen jullie deze van een **waardering** te voorzien. Dit kan simpelweg digitaal via mail. Hiervoor heb je maximaal **30** minuten de tijd, en kan gedurende de **gehele volgende week**. Verdere instructies kan je aan het begin van volgende week verwachten.

Nogmaals dank en fijne avond,

Pim de Jong
Universiteit Utrecht
06-29409111
p.s.h.dejong@students.uu.nl
pim.de.jong@sogeti.com

6 EXPERIMENT CASE MOMENT 2

Introductie

Experiment sessie #2

Op de volgende pagina's kom je een opdracht tegen over de Utrecht Bank. De Utrecht Bank heeft in het verleden een informatieplan opgesteld, met een aantal architectuurmodellen als potentiële oplossingen. Hier dien je veranderingen in de bestaande architectuur door te voeren. Hiervoor moet je ook (gedeeltelijk) de architectuur herontwerpen. De architectuurmodellen van het informatieplan zijn ook digitaal in je mailbox te vinden. Het is van belang om tijdens het ontwerpen je ontwerpbeslissingen van rationale (verantwoording) te voorzien. Dit zal op de volgende pagina's verder worden toegelicht. Hiervoor krijg je een aparte documentatie template digitaal opgestuurd.

De Rationale Capture Cycle

Een van de doelen van het onderzoek is het in de praktijk testen van de *Rationale Capture Cycle*, het ontworpen instrument om rationale te vangen tijdens architectuurontwerp. Het instrument neemt de vorm aan van een visueel model, ten behoeve van het gebruiksgemak. Tijdens deze sessie zal het model weer in gebruik worden genomen. Geen zorgen, hiervoor hoef je vrijwel geen extra input te leveren. Sterker nog, het is ontworpen om je tijdens het ontwerpproces te helpen!

Om je in het ontwerpproces te ondersteunen heb je een A4 met de '*Rationale Capture Cycle*' gekregen. Je moet dit zien als een hulpmiddel bij de totstandkoming van je ontwerpbeslissing. Het model geeft je inzicht in welke elementen belangrijk zijn wanneer je een beslissingspunt tegenkomt in je ontwerpproces. Het model is in essentie een visuele weergave van de belangrijkste beredeneringsaspecten die samen de kern van je beredeneringsproces vormen. Je kan het gebruiken om structuur te brengen in het proces van voldoen aan requirements of aanpakken van issues. Daarnaast is het waardevol als checklist bij het structureren van je rationale in de documentatie van je architectuur.

Het model is dus ontworpen om je handvaten te geven bij het maken en evalueren van ontwerpbeslissingen, maar geeft je ook richting in de documentatie hiervan. Belangrijk is dat je het model niet als vaste procedure ziet, maar als hulpmiddel tijdens je reguliere ontwerpproces. Aan het einde van de casus krijg je de gelegenheid om het model van terugkoppeling te voorzien.

Op de volgende pagina begint de casus. Lees deze eerst goed door. Daarna zal de casus een bestaande architectuur tonen, het zal dan aan jou zijn om de as-is architectuur dusdanig aan te passen zodat deze aan nieuwe wensen voldoet. Deze wensen zullen tijdens de casus naar voren komen. Ook zal de casus duidelijk maken wat je uiteindelijk dient op te leveren, en in welke format. Tijdens het ontwerpen dien je je gedachtegang te documenteren, inclusief de verantwoording van je ontwerpbeslissingen.

Om praktische redenen heb je **2 uur** de tijd om de casus te voltooien. Wanneer je, volgens jou, geen bijdrage meer kan leveren of waarde kan toevoegen ben je natuurlijk vrij om je werk eerder in te leveren. Belangrijk: vul de **starttijd** bovenin deze pagina voor het lezen van de casus. Vergeet ook niet de **eindtijd** in te vullen wanneer je klaar bent.

Succes!

De Utrecht Bank casus
In de komende twee uur dien je architectuurmodellen te herontwerpen op basis van een aantal criteria. Lees eerst goed de opdracht door. Op de volgende pagina staan verdere instructies ten behoeve van de output.
Opdracht
In een vorig stadium is een informatieplan opgesteld om de database structuur te centraliseren. Hier is de vertaalslag gemaakt naar een werkbare architectuur om dit te bewerkstelligen. Het bestuur heeft het informatieplan uitgebreid overwogen maar kan de keuze tussen de verschillende opties niet maken en heeft een verzoek voor nadere uitwerking uitgezet. De CIO benadrukt dat het bestuur wel erg tevreden was over de creativiteit waarmee de architecten te werk zijn gegaan.

Op de volgende pagina's staat het opgeleverde informatieplan. Het bestuur overweegt de verschillende opties die geleverd zijn in het informatieplan, maar is nog onzeker omtrent een aantal factoren. De elementen waar het bestuur onzeker over is zijn als volgt:

1. **Optie A: Datakwaliteit** – In optie A wordt een caching applicatiefunctie geïntroduceerd ten behoeve van de snelheid van de database infrastructuur. Het bestuur maakt zich zorgen dat het introduceren van een dergelijke technologie een negatief effect heeft op de datakwaliteit. Dit negatieve effect kan zich potentieel manifesteren in de vorm van conflicterende databases, synchronisatie issues en missing data.
2. **Optie B: Kosten / haalbaarheid** – In optie B is er verkozen voor een Enterprise Service Bus als patroon. Hoewel de CIO overtuigd is van de ontwikkeling richting een SOA patroon, maakt het bestuur zich zorgen om de haalbaarheid en kosten van een dergelijke oplossing. Het bestuur kan daar momenteel niet de middelen voor vrijmaken en is benieuwd naar een goedkoper alternatief.
3. **Optie C: Database performance** – In optie C adviseert het informatieplan alle databases te centraliseren tot één enkele data warehouse. Het bestuur is bang dat deze oplossing zorgt voor performance problemen als alle applicatieservices nu aan één enkele bron worden gehaakt. De CIO maakt zich zorgen over het aantal dataverzoeken dat nu richting een single point of failure gaat.

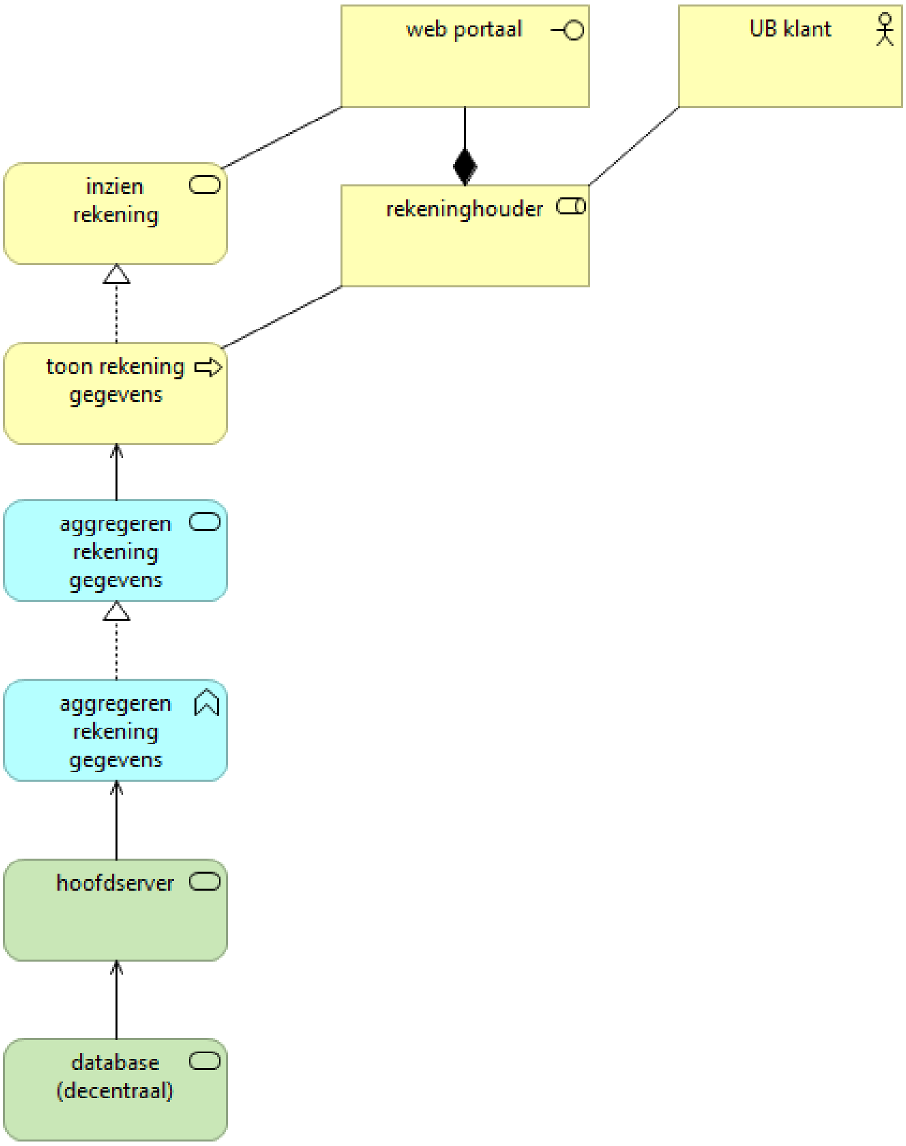
Het bestuur vraagt jou als architect de verschillende opties uit te werken en, indien nodig, aanpassingen door te voeren. Uiteindelijk verwacht het bestuur dat alle opties worden uitgewerkt / aangepast zodat er de bovenstaande zorgen worden verlicht of verholpen.

Het bestuur benadrukt dat het beredeneringsproces achter de aanpassingen helder moet zijn. Zo weet de organisatie waarom en hoe de veranderingen aan dit informatieplan uiteindelijk tot haar doelen kan leiden. Daarom wil het bestuur graag dat je je volledige gedachtegang beschrijft en documenteert tijdens het ontwerpen. Het bestuur leest graag terug wat de gedachtegang achter je ontwerpbeslissingen was.

Rationale
Het informatieplan op de volgende pagina bestaat uit **twee** onderdelen: de **architectuurmodellen** en de bijbehorende **rationale**. Het bestuur spoort je aan goed gebruik te maken van de rationale in het informatieplan. Deze vertegenwoordigt namelijk de gedachtegang van de originele architecten en diens architectuurmodellen uit het informatieplan. De CIO zou het zonde vinden als je deze niet mee zou nemen bij het doorvoeren van de beoogde veranderingen en het maken van je ontwerpbeslissingen.

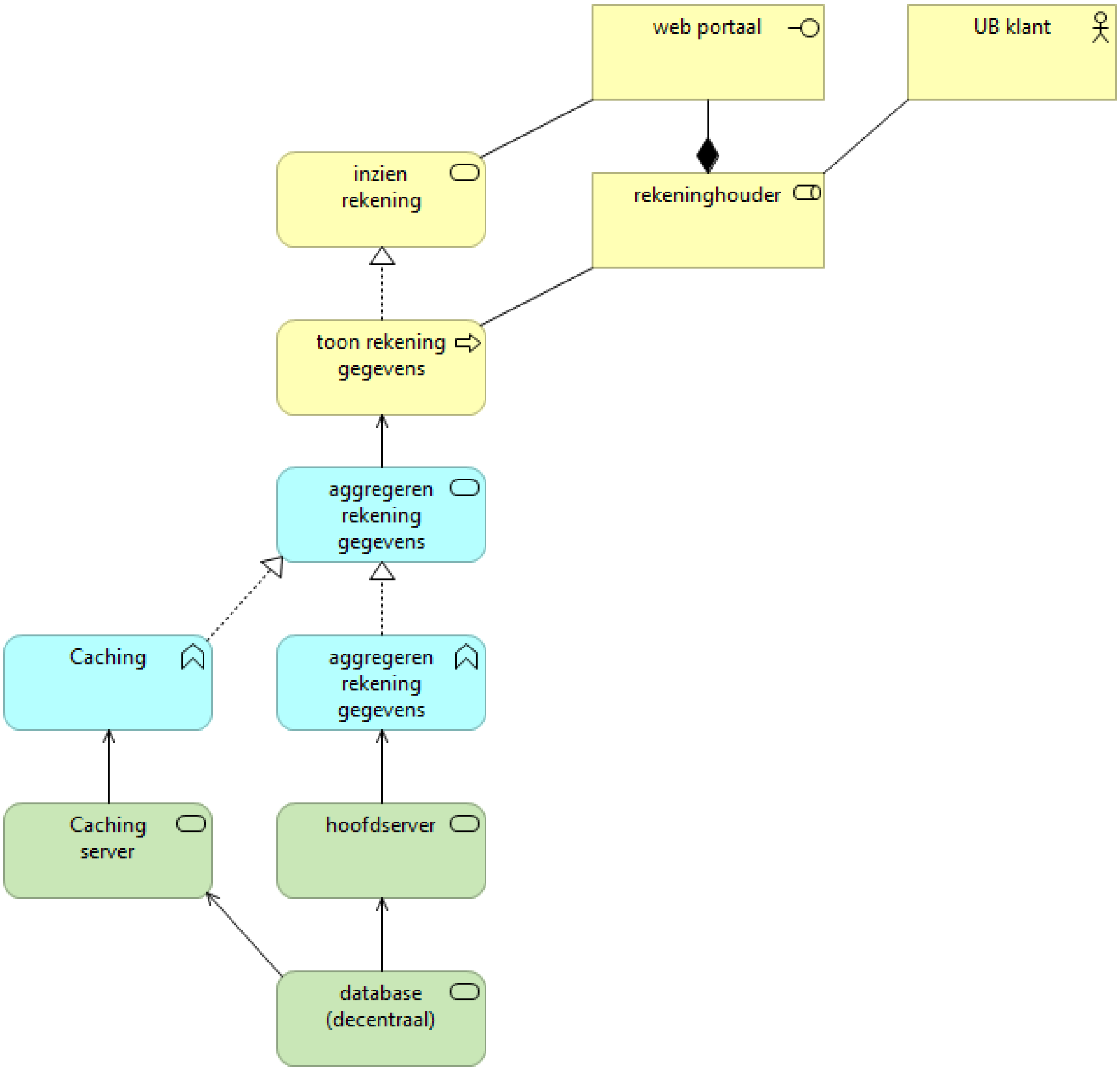
Het bestuur ziet het gebruik van de bijbehorende rationale ook graag terug in je advies en documentatie. Er zal **extra** gelet worden op of de mening en gedachtegang van het originele informatieplan **expliciet** meegenomen is in je documentatie en uitwerking. De rationale zal duidelijk worden aangegeven in het informatieplan. Aan het eind van de opdracht zal er ook een aantal **vragen** gesteld worden over de aanwezigheid van de rationale van het informatieplan en hoe deze je je ontwerpproces en ervaring beïnvloedde.

Het Informatieplan
As-is architectuur
Het volgende model geeft de as-is decentrale database landschap weer van de Utrecht Bank. De daaropvolgende 3 modellen, op de volgende pagina's, geven de verschillende opties weer die de architecten hebben geopperd tijdens het opstellen van het informatieplan. Daar staat ook het criterium die centraal staat bij het doorvoeren van een verandering rechts bovenin.



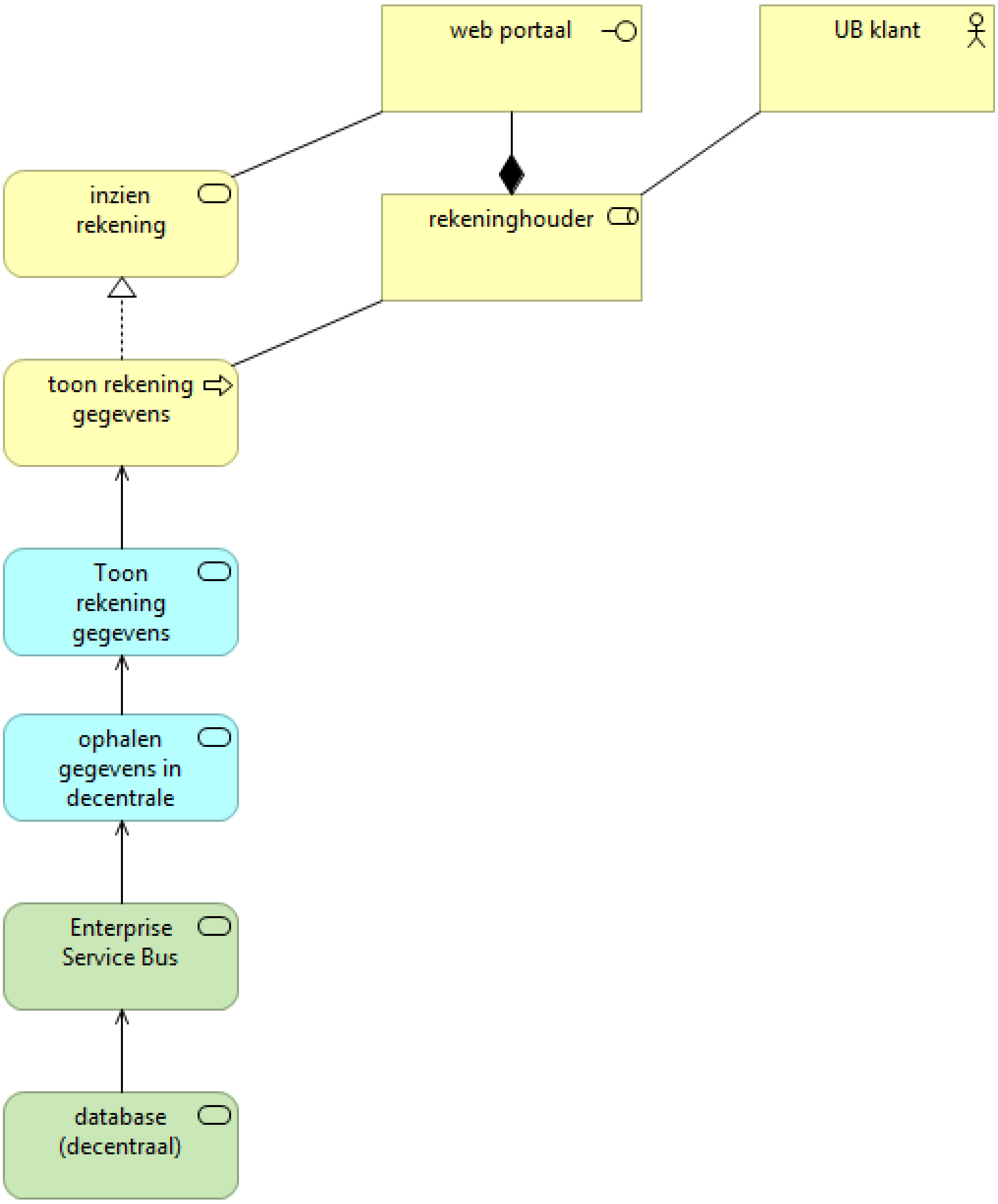
Figuur 1. Baseline model (as-is architecture)

Optie A - **Datakwaliteit**



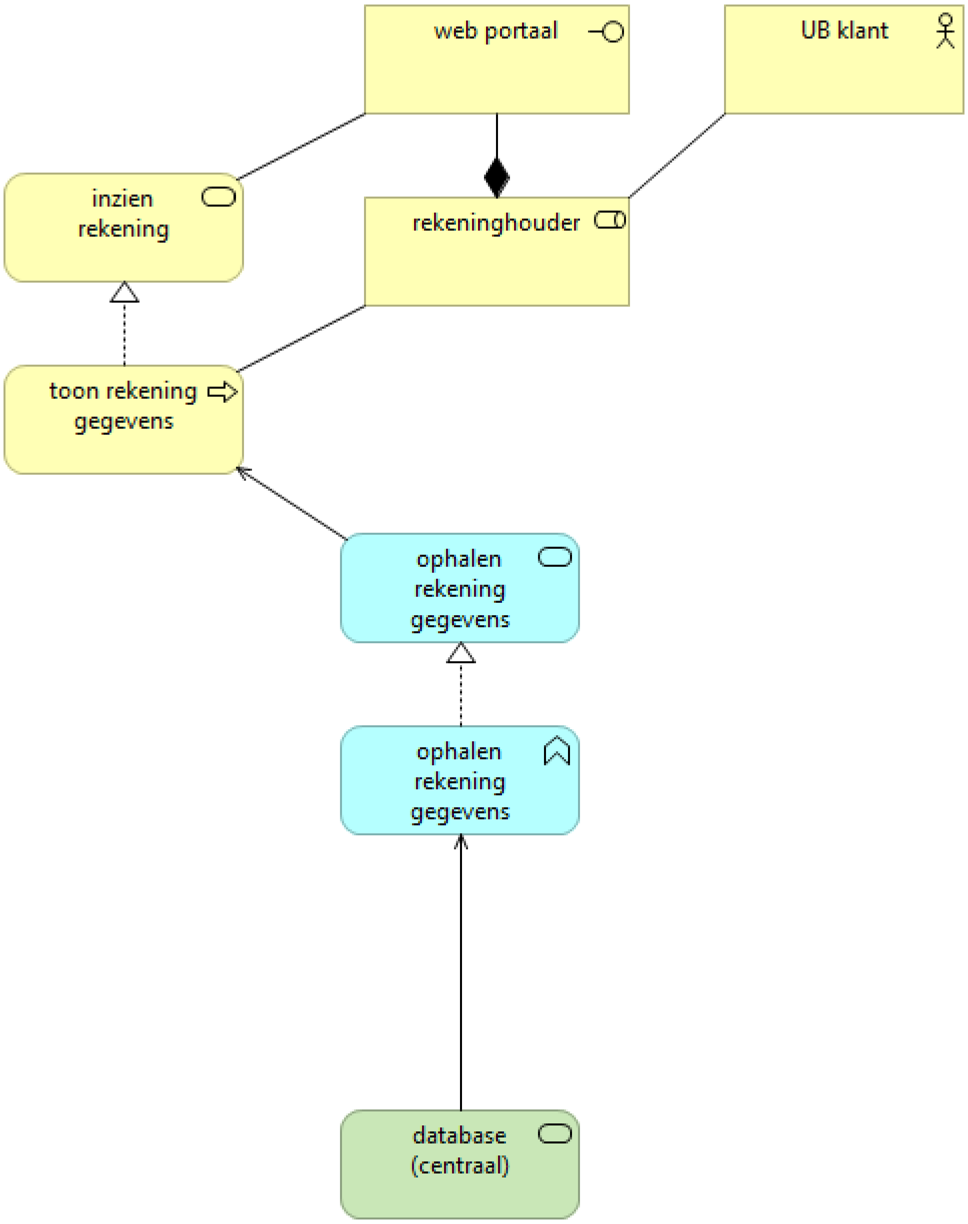
Figuur 2. Optie A

Optie B - **Kostenreductie**



Figuur 3. Optie B

Optie C – **Performance**



Figuur 4. Optie C

Architectuur beschrijving

- **Baseline model:** Momenteel staan de verscheidene databases van de Utrecht Bank (particulier, zakelijk, hypotheek, verzekeringen etc.) decentraal op lokale servers. Wanneer er data wordt opgevraagd door de web portal krijgt de hoofdserver een verzoek voor informatie die vervolgens één voor één het verzoek door vertaalt naar de desbetreffende databases. Wanneer dit proces is voltooid aggregeert de hoofdserver de data en creëert deze een view op basis van alle opgegeven velden.
- **Optie A**: Het versnellen van gegevens verzameling door middel van de introductie van een caching applicatiefunctie.
- **Optie B**: introductie van een webservices landschap, losjes koppelen.
- **Optie C**: Alle decentrale databases centraliseren

Rationale
Hieronder volgt de verantwoording van en gedachtegang achter de gemaakte ontwerpbeslissingen van de vorige modellen, zoals opgeleverd bij het informatieplan. De CIO verwacht dat je goed gebruik maakt van deze informatie in het kader van het wiel niet opnieuw willen uitvinden. Hij zal er ook naar zoeken in de documentatie van je ontwerpbeslissingen.

Probleem
Vraag:
- Verbeter de snelheid van het webportaal.

Condities
- Waarborging overgang van decentraal naar centraal.
- Security is een primaire zorg.

Opties
A. Versnellen verzamelen gegevens door introductie van een caching applicatiefunctie door databases speciaal voor het portaal.
B. Introductie van een webservices landschap, losjes koppelen.
C. Alle decentrale databases centraliseren.

Voordelen en Nadelen:
Optie A:
Voordeel: toepassen bestaande soorten technologie op databases. Security volgens bestaande policies en technologieën. Relatief lage kosten en snelle levertijd.
Nadeel: introductie van een extra database, mogelijke synchronisatieproblemen. Geen bijdrage aan een structurele verbetering van het landschap.
Optie B:
Voordeel: levert een structurele verbetering op van het landschap met een open migratie pad (vele paden en snelheden mogelijk). Toekomstbestendigheid: meer mogelijkheden om aan te sluiten met nieuwe dienstverlening, ook van derden.
Nadeel: introductie van een nieuwe technologie, mogelijk onbekend bij huidige it organisatie. Mogelijk kinderziektes.
Optie C:
Voordeel: structurele verbetering van het informatie-landschap door uitfasering decentrale databases. Geen introductie van een nieuwe technologie.
Nadeel: doorlooptijd is relatief hoog doordat alle decentrale diensten nu ook aangepast dienen te worden.
Aannames
- De performance van de web services is zo op te schalen dat de hele keten sneller is dan voorheen.
- De huidige IT organisatie heeft voldoende en actuele capaciteit om databases te beheren.
- De huidige IT organisatie heeft onvoldoende kennis om een services landschap te kunnen ontwerpen, bouwen en onderhouden.
- De huidige systemen voldoen aan de security eisen van de toezichthouder.

Beperkingen
- De verbeteringen die centralisatie nastreven (opties B en C) zijn erg gericht op de nieuwe klanten. Het kan zo maar zijn dat de decentrale filiaalhouders hierdoor benadeeld worden in hun bedrijfsvoering en dienstverlening.
- Banken moeten voldoen aan de security eisen van een toezichthouder. De huidige systemen voldoen (aanname) aan deze eisen. Een nieuw SOA platform betekent nieuwe security risico's met mogelijk onbekende maatregelen.
- Kan / wil de IT organisatie een nieuw avontuur aan en willen ze een SOA landschap introduceren?
- Is er een hoge snelheid van realisatie geboden? Is er een burning platform?

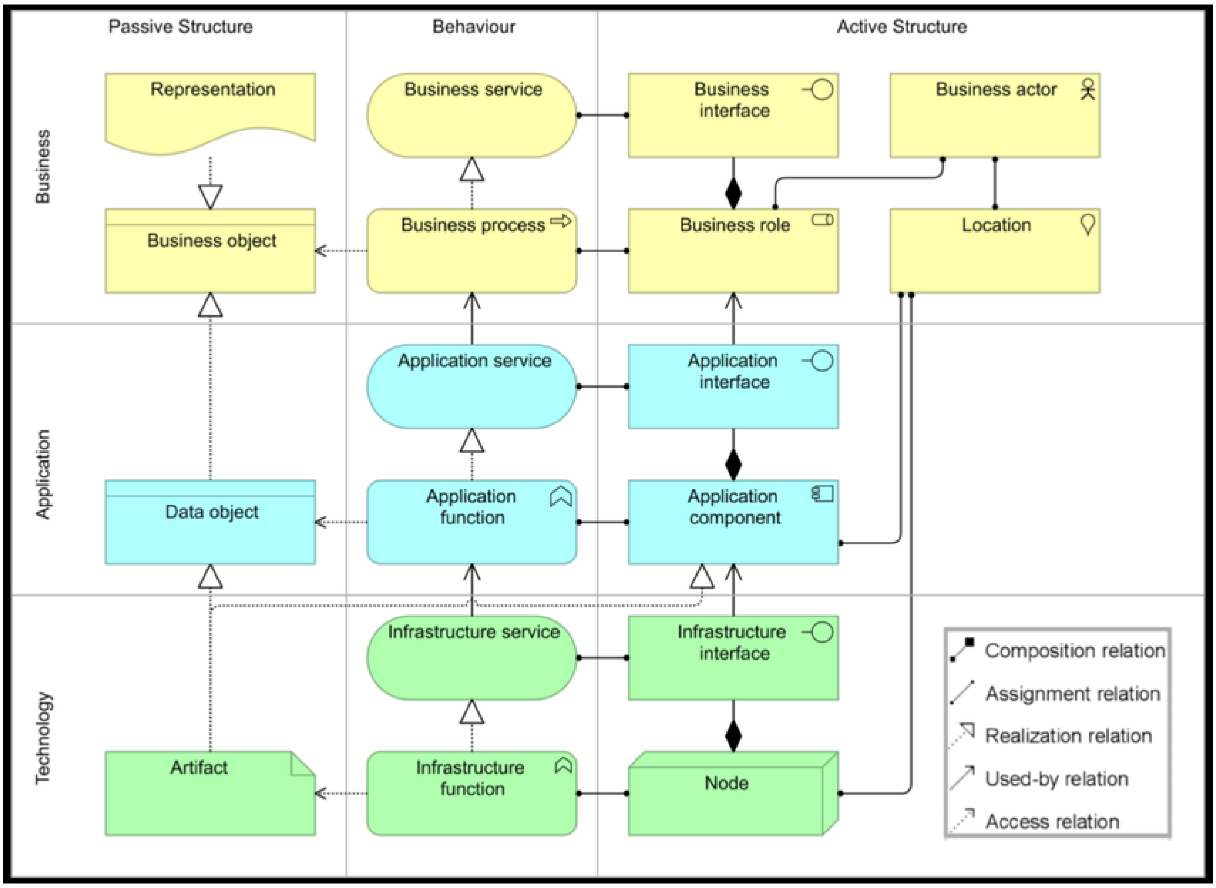
Risico analyse
Risico: Aannames kunnen onjuist zijn. Deze dienen gevalideerd te worden alvorens het besluit voor een optie genomen kan worden.

Trade-off analyse
Voorkeur voor Optie B: introductie van SOA.

Omdat naast de huidige scope van de vraag voor het verbeteren van de snelheid van het web portaal er meer is. Deze optie geeft een open migratie pad om een structureel probleem op te lossen en vooral: nieuwe diensten mogelijk te maken die niet allemaal meer zelf door de bank ontwikkeld hoeven te worden.

Output
Om overeenstemming omtrent de gewenste producten te garanderen verwacht de CIO dat je je aan de volgende globale **principes** houdt:

- Je dient verduidelijking op te leveren omtrent de zorgen van het bestuur. Je wordt vrijgelaten in de keuze voor oplossingen, zelfs als deze (gedeeltelijk) buitenshuis moeten worden gerealiseerd. Het rekening houden met de Utrecht Bank en de door haar opgelegde criteria zijn erg belangrijk.
- Je hoeft niet in (technisch) detail te treden, als de benodigde informatiestromen en –behoeften maar in kaart worden gebracht. Noch hoef je volledig nieuwe modellen te ontwerpen, als de veranderingen maar helder zijn.
- De CIO ziet graag dat je conform de standaarden in de organisatie werkt en vraagt je ArchiMate te hanteren. Hij geeft je het volgende metamodel van ArchiMate 3.0 als handvat:



Figuur 5. ArchiMate 3.0 metamodel

De volgende verwachtingen zijn duidelijk gemaakt door het bestuur:

3. Het bestuur ziet graag dat je alle opties (A, B en C) hebt uitgewerkt en deze van **minimaal één** verandering voorziet. Het bestuur verwacht dus **drie** modellen (A, B en C), waarbij in elk model en bijbehorende documentatie duidelijk is hoe de zorgen van het bestuur worden gewaarborgd.
4. Het is **cruciaal** dat je je gemaakte ontwerpbeslissingen goed **documenteert**. De CIO is van mening dat een architectuurmodel zonder context minder waarde heeft. Daarom wil hij dat je je beslissingen van rationale voorziet. Daarvoor heeft hij een **architectuur documentatie template** opgesteld. Deze is te vinden in je **mailbox** en dient bijgehouden te worden terwijl je je modellen ontwerpt. Uiteindelijk wil hij deze ook in zijn mailbox weer ontvangen.

5. De Utrecht Bank wil uniformiteit onder haar werknemers en het werk dat ze leveren en verwacht dat je producten oplevert in de **ArchiMate** notatie. De Utrecht Bank heeft hier zelf geen licentie voor beschikbaar, maar gebruikt **Archi** als gratis tool. Daarnaast ontvangt de Utrecht Bank graag de digitale modellen en de documentatietemplate via mail naar het onderstaande adres.

Hoogachtend,

Pim de Jong
p.s.h.dejong@students.uu.nl / pim.de.jong@sogeti.com
De Utrecht Bank

Survey

De volgende vragen gaan over je ervaring tijdens het maken van de casus. De vragen gaan voornamelijk in op de documentatie en hoe dit je ontwerpproces beïnvloedde.

1. Was de huidige architectuur eenvoudig te begrijpen? Kon je de gedachtegang achter de architectuur interpreteren? Neem je de originele gedachtegang en context ook mee in je ontwerp?

____

____

____

2. Welke informatie mis je in het informatieplan? Welke elementen ontbraken om de bestaande architectuur te herontwerpen of interpreteren?

____

____

____

3. Welke elementen uit de rationale (aannames, risico's, probleemstelling etc.) heb je gebruikt bij het interpreteren van de architectuur? En welke bij het ontwerpen?

____

____

____

4. Welke elementen uit de rationale (aannames, risico's, probleemstelling etc.) heb je **niet** gebruikt bij het interpreteren van de architectuur? En welke bij het ontwerpen? Welke elementen waren niet nuttig?

____

____

____

5. Was de rationale gestructureerd? Volgde de rationale documentatie een logische opbouw?

Rationale Capture Cycle

De volgende vragen gaan over het gebruik maken van het model tijdens het maken van de casus. De vragen gaan specifiek over in hoeverre het model gebruiksvriendelijk, laagdrempelig, effectief en behulpzaam was tijdens het architectuurontwerpproces.

1. Was het model van toegevoegde waarde? Hielp het bij het bedenken van oplossingen, risico's, aannames etc. waar je normaliter wellicht niet aan zou denken?

2. Was het model effectief bij het documenteren van je ontwerpbeslissingen en architectuur?

3. Was het model leesbaar? Was het direct duidelijk wat er met de termen werd bedoeld? Leidde de syntax van het model af van de inhoud?

4. Was het model eenvoudig in het gebruik? Was het direct duidelijk hoe het model gebruikt kon worden? Was het model intuïtief?

____

____

____

5. Voelde het model natuurlijk? Kon je het model eenvoudig gebruiken tijdens je ontwerpproces? Voelde het model als ontwrichtend of storend aan?

____

____

____

____

6. Is de volgorde van de rationale elementen in het model logisch? Volgt het model een correcte opbouw?

____

____

____

____

7. Ik wil het volgende nog graag kwijt over het model, de casus of het onderzoek:

____

____

____

____

8. Ik heb de volgende activiteiten uitgevoerd:
    d. Mijn **naam**, **starttijd** en **eindtijd** zijn ingevuld.
    e. Ik heb de modellen **geëxporteerd** en **opgestuurd**.
    f. Ik heb de architectuur documentatie template **bijgehouden** tijdens het ontwerpproces en deze **opgestuurd**.

Einde sessie 2
Bedankt voor de participatie! Je levert onmisbare input voor het vorderen van het onderzoek. Nadat ik alle resultaten binnen heb kunnen jullie een mail van mij verwachten met daarin de vraag de sessie af te ronden door middel van een assessment. Hier krijgen jullie een **drietal** oplossingen, gemaakt door jullie collega's, en dienen jullie deze van een **waardering** te voorzien. Dit kan simpelweg digitaal via mail. Hiervoor heb je maximaal **30** minuten de tijd. Verdere instructies zullen, via mail, volgen.

Nogmaals dank en fijne avond,

Pim de Jong
Universiteit Utrecht
p.s.h.dejong@students.uu.nl / pim.de.jong@sogeti.com
06-29409111

7 SOLUTION MATRICES

The matrices below represent the Excel tables that were used in order to assign the solutions fairly and evenly among the participants.

Moment 1:

Solutions	Integer	How many times will it be ranked?
T1	1	3
T2	2	3
T3	3	3
T4	4	3
T5	5	3
C1	6	3
C2	7	3
C3	8	3
C4	9	3
C5	10	3

Sol	1	2	3	4	5	6	7	8	9	10
1	x		1	1	1	1			1	1
2		x			1	1	1	1	1	1
3			x	1		1		1	1	1
4				x			1	1	2	
5					x	1	1		1	1
6						x	1	1		
7							x	1		1
8								x		1
9									x	
10										x

		randomizer.org				
Ranks own solution?	1 = no, 2 yes	1	1	1	1	1
	Participant	**_1_**	**_2_**	**_3_**	**_4_**	**_5_**
		2	3	1	5	2
		6	6	4	1	10
		7	8	9	6	8
		Test group				
		8	Test solutions			
		7	Control solutions			

1	1	1	1	1			
_6_	**_7_**	**_8_**	**_9_**	**_10_**			
1	3	5	7	7		15 higher than 5	(control group)
3	4	2	10	8		15 lower than 6	(test group)
10	9	9	5	4			
Control group							
7	Test solutions						
8	Control solutions						

Moment 2:

Solutions	Integer	How many times will it be ranked?
T1	1	3
T2	2	3
T3	3	3
T4	4	3
C1	5	3
C2	6	3
C3	7	3
C4	8	3

Solutions	1	2	3	4	5	6	7	8	
1	-	x	x	x	x	x	x		
2		-	x		x	x	x	x	Occurring twice:
3			-		xx	x	x		3 vs 5
4				-	x	x	x	xx	4 vs 8
5					-			x	
6						-	x	x	
7							-	x	
8								-	
9									
10									

Sessie #2					
		T1	T2	T3	T4
		randomizer.org			
Ranks own solu	1= no, 2 yes	1	1	1	1
	Participant	*1*	*2*	*3*	*4*
		2	6	4	1
		3	7	6	2
		5	3	8	6
		Test group			
		6	Test solutions		
		6	Control solutions		

C1	C2	C3	C4			
1	1	1	1			
5	*6*	*7*	*8*			
7	1	5	1	12	Higher than 3	Control group
8	4	8	3	12	lower than 4	Test group
2	7	4	5			
	Control group					
	6	Test solutions				
	6	Control solutions				

8 ARCHITECTURE DESCRIPTION TEMPLATE

The word document below is the Architecture Decision Description Template that was used by the participants during the experiment sessions to document their architecture models and explicate their reasoning and rationale.

Aanvullende architectuurdocumentatie

Naam:
Emailadres:

Instructies

In dit document dien je de gemaakte architectuurmodellen van aanvullende documentatie te voorzien. Dit document dient als template en dient als handvat tijdens het beschrijven van de gemaakte modellen. De modellen kunnen als geheel worden geëxporteerd als .archimate model. Dit doe je door naar 'File → Save As' te navigeren en het bestand op te slaan op een locatie naar keuze.
Het moet duidelijk zijn over welk model de beschrijving en rationale gaat. Schrijf achter elke kop de titel van het desbetreffende model in Archi. Je kan in Archi verschillende modellen aanmaken door op CTRL + N te drukken, en met de rechtermuisknop kan je via 'Rename' de modellen een andere naam geven. Je mag zelf bepalen in welke volgorde de modellen voorkomen en je mag ook bepalen in hoeverre je de modellen wil opsplitsen. De modellen veelal aan elkaar hangen is dus prima, zolang uit deze documentatie duidelijk blijkt naar welk model of element je refereert.

Architectuurmodel: Vul hier de titel van het gemaakte model in Archi in.

Beschrijving: Beschrijf het bijgevoegde model. Waar kijken we naar? Uit welke elementen bestaat het model en hoe hangen ze samen?

Rationale: Voeg een verantwoording van het gemaakte ontwerp toe. Waarom is het model zoals het is? Waarom kies je voor een bepaald ontwerp? Hoe ben je tot die beslissing gekomen? Waarom kies je voor bepaalde elementen ten opzichte van anderen? Het doel is dat je je **beredeneringsproces** zo nauwkeurig mogelijk expliciet maakt.

Op de volgende pagina's volgen lege ruimtes om de modellen in te documenteren. Ze hoeven uiteraard niet allemaal gebruikt te worden en je kan, mocht dat nodig zijn, zelf pagina's toevoegen.

Model 1

Architectuurmodel:

Beschrijving:

Rationale:

9 BRADLEY TERRY MODEL OUTPUT

These excel tables represent the exact output when fitting the Bradley Terry pairwise ranking model.

Moment 1:

	A	B	C	D
1	**Team 1**	**Team 2**	**Win 1**	**Win 2**
2	T2	C1	1	0
3	T2	C2	1	0
4	C1	C2	1	0
5	T3	C3	1	0
6	T3	C1	1	0
7	C1	C3	1	0
8	T4	T1	1	0
9	T4	C4	1	0
10	T1	C4	1	0
11	T1	C1	1	0
12	T1	T5	1	0
13	C1	T5	1	0
14	T2	C5	1	0
15	T2	C3	1	0
16	C5	C3	1	0
17	C5	T1	1	0
18	C5	T3	1	0
19	T1	T3	1	0
20	T3	T4	1	0
21	T3	C4	1	0
22	C4	T4	1	0
23	T5	C4	1	0
24	T5	T2	1	0
25	T2	C4	1	0
26	C2	C5	1	0
27	C2	T5	1	0
28	C5	T5	1	0
29	T4	C3	1	0
30	T4	C2	1	0
31	C3	C2	1	0
32	T2	C1	1	0

Summary statistics:

test	Total	mber of wi	mber of los	umber of tie	% of wins	% of losses	% of ties
C1	6.00	3.00	3.00	0.00	50.00	50.00	0.00
C2	6.00	2.00	4.00	0.00	33.33	66.67	0.00
C3	6.00	1.00	5.00	0.00	16.67	83.33	0.00
C4	6.00	1.00	5.00	0.00	16.67	83.33	0.00
C5	6.00	4.00	2.00	0.00	66.67	33.33	0.00
T1	6.00	4.00	2.00	0.00	66.67	33.33	0.00
T2	6.00	5.00	1.00	0.00	83.33	16.67	0.00
T3	6.00	4.00	2.00	0.00	66.67	33.33	0.00
T4	6.00	4.00	2.00	0.00	66.67	33.33	0.00
T5	6.00	2.00	4.00	0.00	33.33	66.67	0.00

Estimated parameters:

	Estimate	ndard devia	ower boun	Jpper bound
C1	0.721	0.418	-0.097	1.540
C2	0.342	0.247	-0.142	0.826
C3	0.186	0.179	-0.165	0.537
C4	0.199	0.205	-0.202	0.600
C5	1.671	0.745	0.210	3.132
T1	1.561	0.708	0.174	2.948
T2	2.525	0.906	0.749	4.300
T3	1.471	0.669	0.160	2.782
T4	0.957	0.466	0.045	1.869
T5	0.368	0.274	-0.169	0.905

Probabilities of winning:

	C1	C2	C3	C4	C5	T1	T2	T3	T4	T5	
C1	0.000	0.678	0.795	0.784	0.301	0.316	0.222	0.329	0.430	0.662	0.452
C2	0.322	0.000	0.648	0.632	0.170	0.180	0.119	0.189	0.263	0.482	0.300
C3	0.205	0.352	0.000	0.483	0.100	0.107	0.069	0.112	0.163	0.336	0.193
C4	0.216	0.368	0.517	0.000	0.107	0.113	0.073	0.119	0.172	0.351	0.204
C5	0.699	0.830	0.900	0.893	0.000	0.517	0.398	0.532	0.636	0.820	0.622
T1	0.684	0.820	0.893	0.887	0.483	0.000	0.382	0.515	0.620	0.809	0.609
T2	0.778	0.881	0.931	0.927	0.602	0.618	0.000	0.632	0.725	0.873	0.697
T3	0.671	0.811	0.888	0.881	0.468	0.485	0.368	0.000	0.606	0.800	0.598
T4	0.570	0.737	0.837	0.828	0.364	0.380	0.275	0.394	0.000	0.722	0.511
T5	0.338	0.518	0.664	0.649	0.180	0.191	0.127	0.200	0.278	0.000	0.314
	0.448	0.600	0.707	0.696	0.278	0.291	0.203	0.302	0.389	0.586	

Moment 2:

Team 1	Team 2	Win 1	Win 2
T2	C1	1	0
C1	T3	1	0
T2	T3	1	0
T3	C2	1	0
T3	C3	1	0
C2	C3	1	0
C4	C2	1	0
C4	T4	1	0
C2	T4	1	0
T1	T2	1	0
T1	C2	1	0
T2	C2	1	0
T2	C3	1	0
T2	C4	1	0
C3	C4	1	0
T1	T4	1	0
T1	C3	1	0
T4	C3	1	0
C1	C4	1	0
C1	T4	1	0
C4	T4	1	0
C1	T1	1	0
C1	T3	1	0
T1	T3	1	0
T2	C1	1	0

Summary statistics:

test	Total	umber of wi	nber of los	umber of ti	% of wins	% of losse	% of ties
C1	6.00	5.00	1.00	0.00	83.33	16.67	0.00
C2	6.00	2.00	4.00	0.00	33.33	66.67	0.00
C3	6.00	1.00	5.00	0.00	16.67	83.33	0.00
C4	6.00	3.00	3.00	0.00	50.00	50.00	0.00
T1	6.00	5.00	1.00	0.00	83.33	16.67	0.00
T2	6.00	5.00	1.00	0.00	83.33	16.67	0.00
T3	6.00	2.00	4.00	0.00	33.33	66.67	0.00
T4	6.00	1.00	5.00	0.00	16.67	83.33	0.00

Estimated parameters:

	Estimate	ndard devia	ower boun	Jpper bound
C1	2.693	0.926	0.878	4.507
C2	0.004	0.004	-0.003	0.011
C3	0.001	0.002	-0.002	0.005
C4	0.006	0.004	-0.002	0.014
T1	2.612	0.908	0.833	4.392
T2	2.619	0.917	0.822	4.415
T3	0.064	0.052	-0.039	0.166
T4	0.001	0.001	-0.001	0.004

Probabilities of winning:

	C1	C2	C3	C4	T1	T2	T3	T4	
C1	0.000	0.998	0.999	0.998	0.508	0.507	0.977	1.000	0.748
C2	0.002	0.000	0.751	0.426	0.002	0.002	0.062	0.783	0.253
C3	0.001	0.249	0.000	0.197	0.001	0.001	0.022	0.545	0.127
C4	0.002	0.574	0.803	0.000	0.002	0.002	0.082	0.830	0.287
T1	0.492	0.998	0.999	0.998	0.000	0.499	0.976	1.000	0.745
T2	0.493	0.998	0.999	0.998	0.501	0.000	0.976	1.000	0.746
T3	0.023	0.938	0.978	0.918	0.024	0.024	0.000	0.982	0.486
T4	0.000	0.217	0.455	0.170	0.000	0.000	0.018	0.000	0.108

10 SPSS OUTPUT

The following pages offer the SPSS output of the various conducted tests. For the sake of size restriction of the thesis, only the most relevant and significant outcomes are added as an appendix. For any other tests and results, the author may be contacted.

10.1 Total Frequencies Rationale Documentation

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	ResearchGroup	Statistic	df	Sig.	Statistic	df	Sig.
Total	1.00	.361	5	.032	.658	5	.003
	2.00	.270	5	.200*	.917	5	.512

*. This is a lower bound of the true significance.

a. Lilliefors Significance Correction

Test Statistics[a]

	Total
Mann-Whitney U	4.000
Wilcoxon W	19.000
Z	-1.798
Asymp. Sig. (2-tailed)	.072
Exact Sig. [2*(1-tailed Sig.)]	.095[b]

a. Grouping Variable: ResearchGroup

10.2 T2 and T4 Results Moment 1

10.2.1 Versus entirety of research groups

Tests of Normality

		Kolmogorov-Smirnov[a]			Shapiro-Wilk		
	ResearchGroup	Statistic	df	Sig.	Statistic	df	Sig.
RationaleFrequency	1.00	.265	8	.103	.826	8	.053
	2.00	.260	2	.			

a. Lilliefors Significance Correction

Ranks

	ResearchGroup	N	Mean Rank	Sum of Ranks
RationaleFrequency	1.00	8	4.50	36.00
	2.00	2	9.50	19.00
	Total	10		

Test Statistics[a]

	RationaleFrequency
Mann-Whitney U	.000
Wilcoxon W	36.000
Z	-2.115
Asymp. Sig. (2-tailed)	.034
Exact Sig. [2*(1-tailed Sig.)]	.044[b]

a. Grouping Variable: ResearchGroup

b. Not corrected for ties.

10.2.2 Sequence and Rationale Frequency

Correlations

		RationaleFrequency	Sequence
RationaleFrequency	Pearson Correlation	1	.918**
	Sig. (2-tailed)		.000
	N	10	10
Sequence	Pearson Correlation	.918**	1
	Sig. (2-tailed)	.000	
	N	10	10

**. Correlation is significant at the 0.01 level (2-tailed).

10.3 Design Rationale Frequency and Original Design Rationale Frequency

Correlations

		RationaleFrequency	OriginalRationaleFrequency
RationaleFrequency	Pearson Correlation	1	.788*
	Sig. (2-tailed)		.020
	N	8	8
OriginalRationaleFrequency	Pearson Correlation	.788*	1
	Sig. (2-tailed)	.020	
	N	8	8

*. Correlation is significant at the 0.05 level (2-tailed).

10.4 Ranking, Wins and Rationale Frequency Correlations

Correlations

		RankingScore	RationaleFrequency	Wins
RankingScore	Pearson Correlation	1	.906**	.836**
	Sig. (2-tailed)		.000	.003
	N	10	10	10
RationaleFrequency	Pearson Correlation	.906**	1	.717*
	Sig. (2-tailed)	.000		.020

	N	10	10	10
Wins	Pearson Correlation	.836**	.717*	1
	Sig. (2-tailed)	.003	.020	
	N	10	10	10

**. Correlation is significant at the 0.01 level (2-tailed).

*. Correlation is significant at the 0.05 level (2-tailed).

10.5 Spearmans rank correlation

Correlations

			ArchitectRanking	BTRanking	RationaleRanking
Spearman's rho	ArchitectRanking	Correlation Coefficient	1.000	.842**	.915**
		Sig. (2-tailed)	.	.002	.000
		N	10	10	10
	BTRanking	Correlation Coefficient	.842**	1.000	.830**
		Sig. (2-tailed)	.002	.	.003
		N	10	10	10
	RationaleRanking	Correlation Coefficient	.915**	.830**	1.000
		Sig. (2-tailed)	.000	.003	.
		N	10	10	10

**. Correlation is significant at the 0.01 level (2-tailed).

10.6 Sanity check correlation

Correlations

			ResearchRank	SanityCheckRank
Spearman's rho	ResearchRank	Correlation Coefficient	1.000	.830**
		Sig. (2-tailed)	.	.003
		N	10	10
	SanityCheckRank	Correlation Coefficient	.830**	1.000
		Sig. (2-tailed)	.003	.
		N	10	10

**. Correlation is significant at the 0.01 level (2-tailed).