Universiteit Utrecht

MBI Graduation Project – Thesis

Sam Leewis

Version 1.0

''The most difficult thing is the decision to act, the rest is merely tenacity.''

- Amelia Earhart

MBI Graduation Project – Thesis

A Framework to Support the Implementation of Business Rules Management Solutions

November 2016/ July 2017

Author:
Sam Leewis
Student #: 5664969
De Dieze 8
3448 CP – Woerden
06-28827585

Utrecht University
Graduate School of Natural Sciences
Master of Business Informatics
Year 2, Semester 3 and 4

First supervisor:
Dr. M. Ruiz
M.Ruiz@uu.nl
Utrecht University

Second supervisor:
Prof. Dr. Sjaak Brinkkemper
S.Brinkkemper@uu.nl
Utrecht University

Daily supervisors:
Dr. Ing. Martijn Zoet
Martijn.zoet@zuyd.nl
Zuyd University of Applied Sciences

Drs. Koen Smit
Koen.smit@hu.nl
HU University of Applied Sciences

Woerden, 11 June 2017

Acknowledgements

This thesis is the last phase of the Business Informatics master at Utrecht University. In November 2016, I started my graduation period, which was carried out at the HU University of Applied Sciences Utrecht (HU) and in collaboration with Zuyd University. This research is triggered by earlier research on the topic of Business Rules Management that is performed for the chair Digital Smart Services.

I would like to express my gratitude to some special people who supported me during this research project. First, my special gratitude goes out to Dr Marcela Ruiz and Prof. Sjaak Brinkkemper for guiding me through the graduation process, but also providing me with feedback on how to perform 'proper' and rigorous research, which eventually resulted into delivering a conference paper and this thesis. Furthermore, I would like to thank my daily supervisor Dr Martijn Zoet for being my BRM go-to-guy. Further, I would like to thank Koen Smit for always being there when I needed practical or scientific advice. Last but not least, I would like to mention my thanks to Prof. Johan Versendaal who provided me with the chance of conducting research at the HU.

Furthermore, I would like to thank the BRM community experts who supplied me with valuable insights and personal experience on implementing BRMS. Without their commitment, it was impossible to deliver the current level of quality and completeness at which this research is right now.

Lastly, I would like to express my appreciation to everyone which I had contact with during this process and provided me with valuable hints and tips, either regarding the subject matter or graduating with a Master's degree in general. Furthermore, I would also like to thank the people close to me for the support and the much-needed distraction while doing this research project.

Woerden, July 11, 2017

Sam Leewis

Abstract

The implementation of software products is a time-consuming activity and needs a high level of expertise to be completed successfully. This is especially the case for software products related to an immature field, such as Business Rules Management (BRM) and Business Rules Management Solutions (BRMS). Support is essential to successfully guide the organisational implementation of a BRMS. Motivated by the diversity of organisational structures and their BRMS implementation contexts, we present the development of a situational-oriented BRMS implementation framework. We adopted the situational artefact construction technique to ensure that the framework can be applied in different situations. The BRMS implementation framework consists of four main elements: 1) the BRMS observation technique, 2) the BRMS construction process, 3) the BRMS metamodel, and 4) the BRMS metamodel support tool. For the construction of the BRMS implementation framework, this study utilises data of 13 BRMS implementation cases distributed over the financial and public sectors in the Netherlands. The BRMS implementation cases were gathered by utilising the BRMS observation technique. Being that these BRMS are already implemented and therefore successful, we utilise these successful cases for the creation of the BRMS implementation framework. 25 design problems are identified related towards implementing a BRMS. This resulted in the discovery of the 9 BRMS problem classes. The BRMS construction process identified 54 design factors, which were distributed over the 9 problem classes. The design factors were created into 37 different design situations, which specified each problem class towards the ideal configuration of the BRMS towards the specified situational factors. This all was created into method fragments, which resulted in methods specific for different situations. The elements identified utilising the BRMS observation technique, and the BRMS construction process resulted in the BRMS metamodel. Lastly, the BRMS metamodel is supported with the BRMS metamodel support tool which is created with the use of the model-driven development tool Eclipse Graphical Modeling Framework. The BRMS implementation framework and its elements (the BRMS observation technique, the BRMS construction process, the BRMS metamodel, and the BRMS metamodel support tool) are a stepping-stone towards further research on situational implementation methodology in the BRM domain.

Table of Content

Table of Tables

Table of Figures

1 Introduction

The implementation of software products is a time-consuming activity and need a high level of expertise to be completed successfully. This is especially the case for software products related to an immature field, in practice and research. As is the same for the implementation of a Business Rules Management Solution (BRMS). The technical implementation of a BRMS, code based, is on a higher level of maturity compared to the organisational implementation of a BRMS. Therefore, support is needed to guide the organisational implementation of a BRMS successfully. This research will focus on the development of a BRMS implementation framework. This chapter will contain the motivation, objectives, proposed solution, research methods, and the scope of this research.

1.1 Motivation

An increasing amount of laws and regulations and the demand for automation raises the demand of handling business rules in a proper way (Boyer & Mili, 2011; Graham, 2007). Relating this to practice, this means that laws and regulations need to be transformed to products and services, which could be used to create added value. Thereby, giving the business rules domain an important and valuable task for practice. Making use of business rules and the increasing amount of business rules, a BRMS is needed to keep a clear overview of all the business rules required in an organisation. Implementing such a BRMS is a difficult task to perform taking into account the maturity of the organisation and the structure of the organisation this again has an effect on the amount of business rules and the complexity of such business rules. Therefore, a solution is needed to support the implementation process and thereby creating added value for the people implementing these BRMS. Potential benefits that a BRMS is expected to add value is that vendors could, for example, increase the traceability, reduce effort in the design phase or creating an overview of what changes occur in the business rules of an organisation.

Mismanagement of business rules and the use of BRMS where these rules are built in are a rising problem. In the Netherlands alone, many examples can be seen related to this issue (Algemene Rekenkamer, 2013; NVZ, 2014). This arising problem result in not delivering products and services on time or the requested quality of those products and services, therefore, resulting in a high-cost factor for these organisations. The past years IT project costs increased significantly, and the failing of IT projects is a notorious problem for example at the Dutch government which loses 1 to 5 billion euro each year due to this issue (Elias, Ulenbelt, Fokke, Bruins Slot, & Meenen, 2014). Implementing a BRMS is partly an IT project and therefore part of these statistics.

Organisations that are involved in creating added value with business rules run into problems concerning the lack of knowledge on how to implement a BRMS from an organisational perspective. The business is searching for knowledge on how to implement a BRMS from an organisational perspective. Examples of organisations that try to create added value with products or services involving business rules are Usoft (Urule), Everest (Blueriq), IBM (IBM Operational Decision Manager), and Oracle (Oracle Policy Automation). The lack of validated knowledge on implementing a BRMS is not appealing to these organisations to enter this problem domain. It seems that the triggers are too complex to solve for these businesses with their current knowledge on the BR domain.

In the field of information systems, the domain of *Business Rules Management* (henceforth BRM) is a relatively young subject of study and gained the interest from researchers the past several years (Zoet, 2014). The scientific world sees many opportunities in BRM, but the BRM topic is certainly not over-researched. This is especially the case for the technical implementation of a business rules management solution (Arnott & Pervan, 2005; Rosca & Wild, 2002). Therefore, this research will focus on the organisational implementation of a BRMS, which compared to the technical implementation of a BRMS, lacks published research (Nelson, Rariden, Sen, & Texas, 2008).

A BRMS contains nine capabilities (as shown in Figure 1, concept BRMS CAPABILITIES), which an organisation can utilise for their own purposes to reap the benefits of BRM (Smit & Zoet, 2016; Zoet, 2014). The nine capabilities of a BRMS are as follows: 1) the ELICITATION capability, 2) the DESIGN capability, 3) the SPECIFICATION capability, 4) the VERIFICATION capability, 5) the VALIDATION capability, 6) the DEPLOYMENT capability, 7) the EXECUTION capability, 8) the MONITORING capability and, 9) the GOVERNANCE capability (Smit & Zoet, 2016; Zoet & Versendaal, 2013).

Previously conducted research has shown that solutions comparable to a BRMS have a common DESIGN PROBLEM (Aier, Riege, & Winter, 2008; Baumöl, 2005; Bucher & Winter, 2010; Klesse & Winter, 2007; Lahrmann & Stroh, 2009; Leist, 2004). A common DESIGN PROBLEM is the difference between the goal state and the current state of a system. A common DESIGN PROBLEM is an indication that common PROBLEM CLASSES, for which DESIGN SOLUTIONS can be created, exists (Winter, 2011b).

Winter, (2011) depicts a PROBLEM CLASS as a set of comparable DESIGN PROBLEMS. A PROBLEM SPACE is a collection of multiple problem classes. An instantiation of a problem class in a specific organisation is defined as a design solution, as shown in Figure 1. In the case of the BRMS problem space, the DESIGN SOLUTION is a specific configuration of the earlier mentioned nine BRMS CAPABILITIES (Smit & Zoet, 2016; Zoet & Versendaal, 2013).

Both the PROBLEM SPACE and the DESIGN SOLUTIONS are subject to SITUATIONAL FACTORS, as shown in Figure 1 (Winter, 2011b). These SITUATIONAL FACTORS describe the context in which an information system artefact or organisation has to operate such that the deployed artefact fits the context of the environment. Situational factors might be elicited directly from the specific context in which a BRMS can be potentially implemented. Research identifying these situational factors is conducted in the situational method engineering research field (Brinkkemper, 1996; Karlsson, Ågerfalk, & Hjalmarsson, 2001; Rolland & Prakash, 1996; van Slooten & Hodes, 1996), with specific applications in software product management (Bekkers, van de Weerd, Brinkkemper, & Mahieu, 2008; van de Weerd, 2009), and business process management (Bucher & Winter, 2010; Ravesteyn & Jansen, 2009).

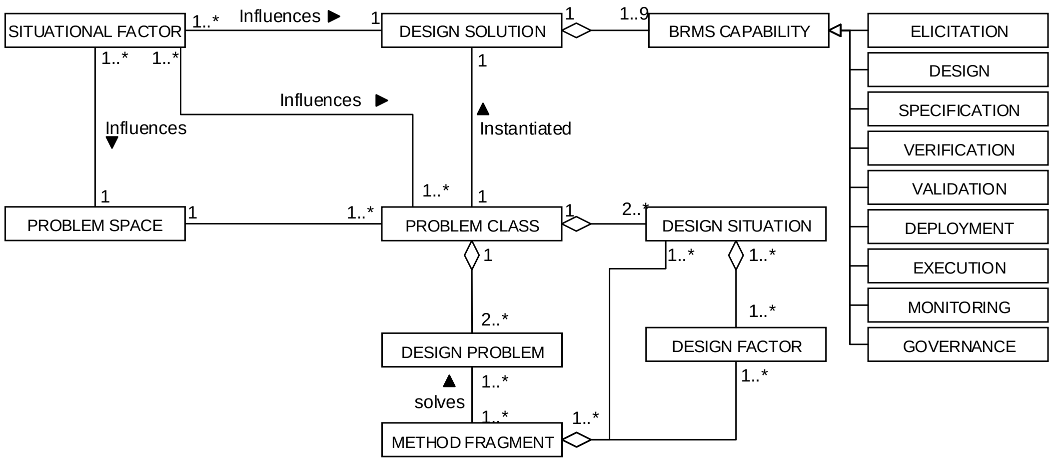


Figure 1 Problem space metamodel

Coming back from what is stated earlier, a BRMS is a complex software system and in that way the implementation is similar to that of the implementation of other software products. The BRMS handles the business rules of an organisation but at the same time needs rules for its development process and is, therefore, a complex software product.

According to the motivation described above, this will result in the following problem statement:

''The current knowledge is based on the technical implementations of business rules management solutions and do not cover the implementation of a business rules management solution from an organisational perspective. This will result in the increase of failing IT project and inherent the increase of IT costs.''

1.2 Objectives

This section contains the research goal and the research questions for the study that is going to be conducted. For this study the following research goal is depicted:

''Develop a framework to support the implementation of a business rules management solution from an organisational perspective."

This research goal results in the following main research question and sub-questions:

''How to develop a framework that supports the organisational implementation of a business rules management solution?''

The main research question is answered with the use of the following sub-questions:

RQ1: How to specify situational factors and problem classes in the business rules management problem space?

The specification of situational factors and problem classes in the BRM problem space will be supported by the discovery of any theory on the specification of situational factors and problem classes. Furthermore, the specification will be validated through expert interviews to validate the correctness.

RQ2: How to design the business rules management solution implementation framework?

RQ2 will be answered with the use of survey research, which discovers any relations between the situational factors and problem classes in the BRM domain. Statistical analysis (cluster analysis) will be conducted to map the relations between the situational factors and problem classes in each specific situation. A state-of-the-art literature review will support the discovery of any existing theories on creating situational artefacts to support the design and development of the BRMS implementation framework.

RQ3: How to validate the correctness of the business rules management solution implementation framework?

RQ3 will be answered with the use of validation expert interviews. These research methods will focus on the correctness of the designed BRMS implementation framework.

1.3 The proposed solution

This section contains a generic description of the proposed BRMS implementation framework. The aim of the BRMS framework is to support the organisational implementation of business rules management solutions. This generic version (as shown in Figure 2) of the BRMS implementation framework focuses on the presentation of the elements of the framework and the relationships among the different elements. The BRMS implementation framework provides guidelines of how to implement a BRMS in a specific situation. Besides the BRMS implementation framework as a deliverable, the framework itself contains four major contributions. These contributions are the BRMS observation technique, the BRMS construction process, the BRMS metamodel, and the BRMS metamodel support tool. Thereby, supporting any future research on situational artefacts in the BRM domain. The framework enables the possibility to support each implementation of a BRMS in each instantiation of a problem class. Therefore, the traceability of the content of the framework is of high importance. An example of this traceability is that the business rules management situational factor contain the source of origin of the situational factor.

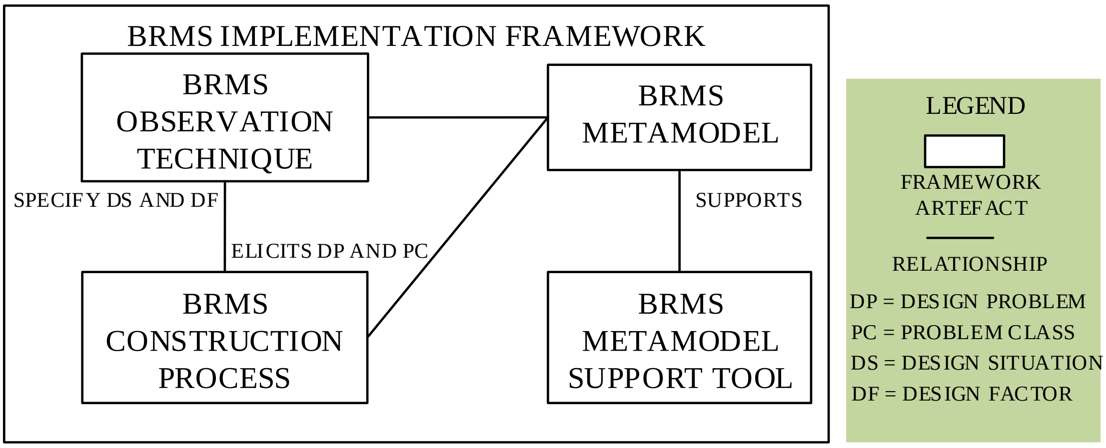


Figure 2 BRMS implementation framework

BRMS observation technique

The BRMS observation technique is used to gather design problems and problem classes extracted form organisations which did a successful BRMS implementation. In this research a survey is utilised as observation technique to gather BRMS implementation cases. Therefore, the BRMS implementation cases are the output of the BRMS observation technique. The BRMS observation technique is described further in Chapter 3.1.

BRMS construction process

The BRMS implementation cases are analysed in the BRMS construction process, which results in the specification of design situations and design factors. The BRMS construction process is described further in Chapter 3.2.

BRMS metamodel

The BRMS metamodel is the result of the BRMS observation technique and the BRMS construction process. The BRMS metamodel contains all the identified elements in the previous elements of the BRMS implementation framework together with their specific values and the relations between these elements. The BRMS metamodel is described further in Chapter 3.3.

BRMS metamodel support tool

The BRMS metamodel is supported by the BRMS metamodel support tool. Eclipse Graphical Modeling Framework (GMF) [1] is used for the model-driven development of the BRMS metamodel support tool. The BRMS metamodel support tool supports the BRMS metamodel and can be utilised by organisations

[1] http://www.eclipse.org/gmf-tooling/

to setup their own configuration of a BRMS. The BRMS metamodel support tool is described further in Chapter 3.4.

1.4 Research method

This section contains the research method that is defined and adopted for this study. Due to the explorative nature of this study the Design Science Framework of Hevner, March, Park, and Ram (2004) will be used to structure the research process of this study. This research methodology focuses on developing solutions for the needs of the 'business' and its environment in the field of information systems (IS). Design research revolves around developing solutions in the form of artefacts.

Figure 3 shows the application of the design science framework (Hevner et al., 2004). This study tries to deliver solutions to the needs as described in the motivation. The delivered solution will be the BRMS implementation framework. This artefact is validated quantitatively through survey research and qualitatively through semi-structured expert interviews. The expert interviews are focused on validating the correctness of the designed BRMS implementation framework. The artefact is refined with the use of the proposed feedback in the assess phase by with the use of the validation expert interviews. The validated artefact and the created knowledge flows back into the environment as a framework on how to select certain business rules management problem classes in different situations. The knowledge base is supplied with a framework for implementing a BRMS for a specific situation needed for an organisation.

Consultants, executives, architects and subject matter experts are the group of people (stakeholders) who are going to use the created artefact (the BRMS implementation framework) in practice. The created artefact is going to be utilised by organisations which focus on offering products and services which create added value by using BRMS or organisations which utilise BRMS themselves. The organisations utilise the created artefact with the use of the BRM and BRMS technology. In the case of the BRMS technology, the technology is selected with the use of the created artefact.

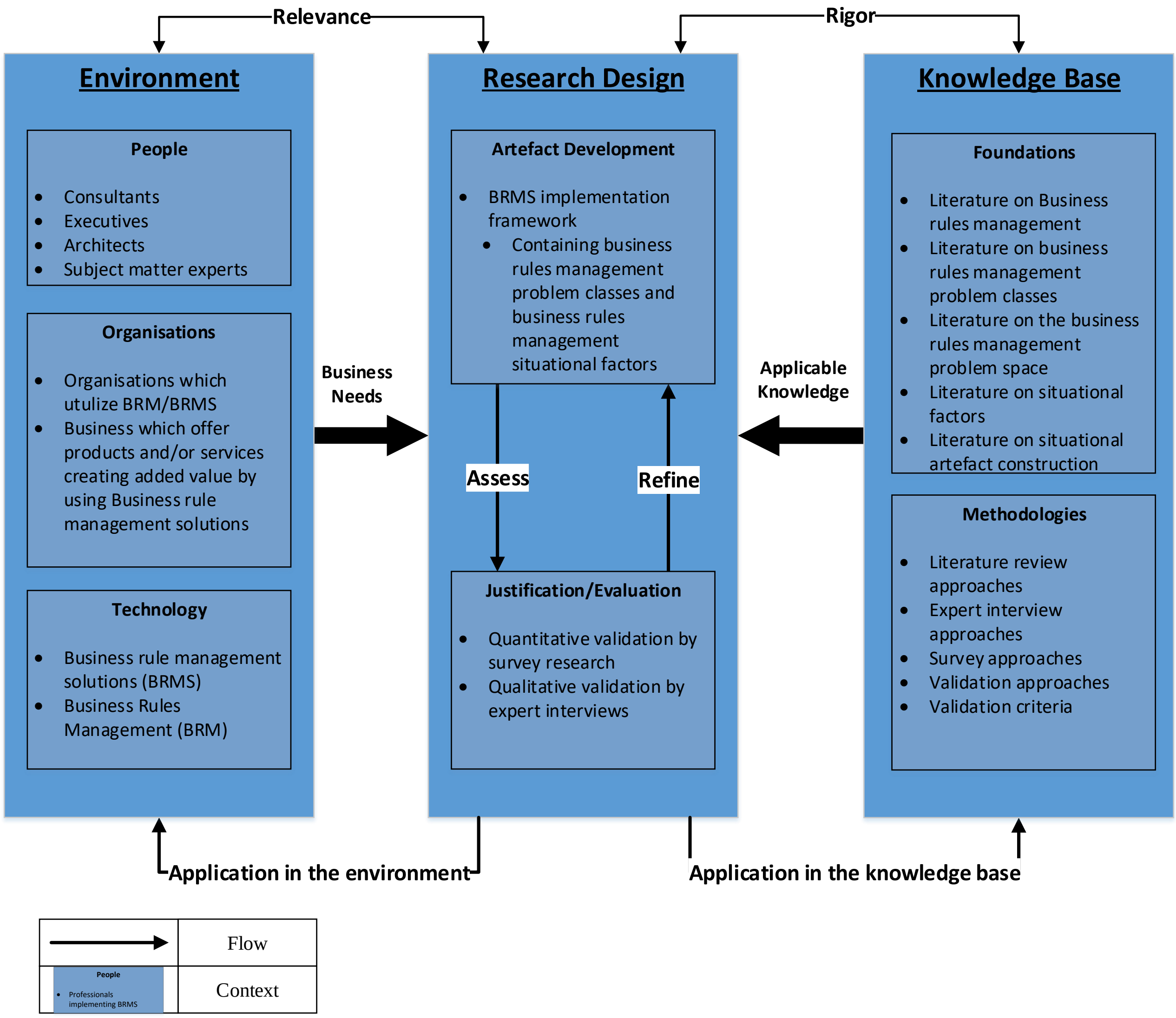


Figure 3 Research design science framework (Hevner et al., 2004)

In order to achieve a suitable solution, it is necessary that the solution is developed iteratively and is validated. Design Research also distinguishes a development and validation process. Developed solutions are added to the knowledge base and thus provide a basis for future research. In addition to these scientific contributions, the developed solution is also practically applicable for the operating environment into which the solution is realised. Hevner et al. (2004) created seven guidelines which should be used when performing IS research. The seven guidelines of Hevner et al. (2004) are adopted for this study and are processed in the design science framework as shown in Figure 3.

1.4.1 Design science research cycle

Ensuring that the choice of design science research is the correct choice we look into other research that is published as Design Science Research (DSR). This section contains the Design Science Research Methodology (DSRM) proposed by Peffers, Tuunanen, Rothenberger, and Chatterjee (2007). Peffers et.al. constructed a design science process model which takes into account features that are shared throughout literature. The design science process model can be found in Figure 4. The DSRM includes principles, practices and procedures required to carry out design science research, offering a nominal process model for doing DSR, and it offers a mental model for presenting and evaluating DSR in IS.

The nominal process models for design science takes into account seven different design science approaches which led to six common process elements being part of the nominal process sequence.

Peffers et al. (2007) and Hevner et.al. (2004) state that, besides the six process steps described earlier, the aspect of iteration also plays a major role when doing design science research. Important is to define beforehand of doing this research the entry point for this research to start the DS process with. The process makes a distinction between four entry points, the four entry points are:

1. Problem centred initiation
2. Objective centred solution
3. Design and development centred initiation
4. Client/context initiated

The design science entry point 'problem centred initiation' is the best fit for this design science research. The problem centred initiation entry point is applicable because of the problem being observed by researchers and businesses within the BR domain. Figure 4 shows the DSRM proposed by Peffers et al. (2007) and together with the activities adapted to this research.

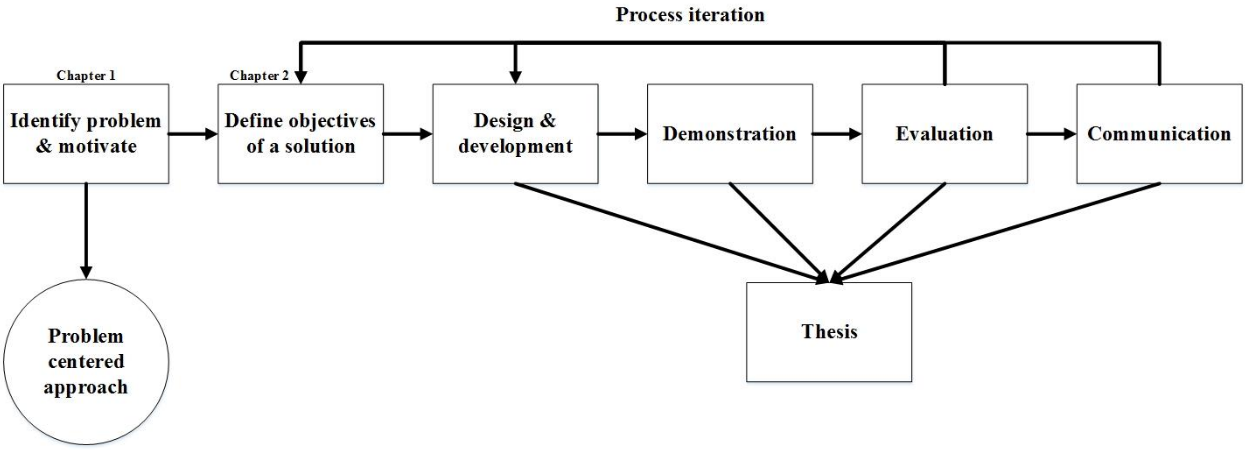


Figure 4 Design science research methodology (Peffers et al., 2007)

Problem identification and motivation;

Define the specific problem statement and justify the value of the solution. The problem identification and motivation can be found in Chapter 1.1, 1.2 and 1.3.

Definition of the objectives for a solution;

Infer the objectives of the solution from problem statement and the knowledge of what is possible and feasible. The research will start with the literature gathering stage. Literature gathering aims to gather existing knowledge from the scientific knowledge base. Aspects of the Systematic Literature Review (SLR) are selected to increase the scientific value and the quality of the state-of-the-art literature overview as a method for conducting the literature review (Kitchenham, 2004). The results of the state-of-the-art overview are used to answer the research questions and therefore supporting the construction of the artefacts later in the research process. Furthermore, the literature gathering state also identifies any knowledge gaps in current research on BRM. Aspects of an SLR are chosen due to its higher scientific value and quality, compared to conventional literature studies, often performed in the first stage(s) of the research (Kitchenham, 2004). The SLR enforces documentation of the review protocol, a search strategy, the applied search procedure, and making the inclusion and exclusion criteria explicit.

In this study, several aspects of the systematic approach proposed by Kitchenham (2004) are chosen because of: 1) its rigorous application of a systematic literature review which is tailored for IS research domain, 2) its practical approach for students conducting research and, 3) the enforcement of deliverables and thereby ensuring the study to be replicated or elaborated further on.

After the literature gathering, the literature is analysed. The literature analysis stage focuses on analysing the outcome of the previous stage. The outcome, all relevant sources, are analysed and put into perspective with the research topic. Included in this stage is also creating a summary of foundations mentioned in the research design science framework in Figure 3. The goal of this literature analysis to create an overview of the existing body of knowledge and to discover similar research conducted in related research fields (business process management, software product management and enterprise architecture).

The definition of the objectives for a solution can be found in Chapter 2 State-of-the-art.

Design and development;

These artefacts are potential constructs, models, methods, or instantiations of new properties of the technical, social or informational resources. The design and development stage will utilise the theoretical results, derived from earlier stages, to create a survey to discover the business rules management problem classes and business rules management situational factors and the relationships between these concepts. The BRMS observation technique is constructed with the results of the previous stages.

The results of the survey are processed and validated with the use of qualitative expert reviews. The outcome of the expert interviews is used to create a validated set of BRM problem classes and BRM situational factors with the relationships between these two concepts. Furthermore, the earlier mentioned concepts are developed, after validation by the experts, into the BRMS implementation framework with the use the situational artefact construction technique of Winter (2011b). The goal of this framework is to propose what configuration of a BRMS is needed to be implemented in a specific situation.

The design and development can be found in Chapter **Fout! Verwijzingsbron niet gevonden.** and **Fout! Verwijzingsbron niet gevonden.**.

Demonstration;

The demonstration of the artefact is conducted during the validation expert interviews. The demonstration of the created artefact can be found in Chapter 4.

Evaluation

In the evaluation stage, the BRMS implementation framework developed in the previous stages is validated on correctness. Validation also occurs during the development stage due to the creation of artefacts, which are used in the creation of other artefacts. The set of BRM problem classes and BRM situational factors are validated with the use of the results of the survey and by the qualitative semi-structured expert interviews. Semi-structured interviews are preferred for this study because this creates the possibility to step on possible new not yet covered knowledge (Bogner, Littig, & Menz, 2009). One of the artefacts, the BRMS implementation framework, is validated with the use of validation expert interviews. McGrath (1984) created the research strategy circumplex and visualises the full spectrum of research strategies. During this study two research strategies are used which are 1) the sample survey for the creation and thereby validation of the BRMS implementation framework and 2) the validation expert interviews to validate the correctness of the BRMS implementation framework. McGrath (1984) identifies three main focuses areas based upon the research approaches. These focus areas are 1) The concern of the generalizability of the gathered evidence over the actors, 2) the concern of the precision of measurement of the behaviours, and 3) The concern of the realism of the situation or context. In this study, the sample survey is used to ensure the maximum concern with generality over actors. The expert interviews contribute to the maximum concern with precision measurement of behaviour. Using these two strategies will ensure that this study is generalizable and that the artefacts are validated without any influence by external factors. The published work of Hevner et.al. (2004) only states that the notion of validation as a general principle which should not be left out when doing design science research. In the work of Hevner et.al. (2004) the how questions stay unanswered and to answer the *''how to''* question a validation method should be used. Validation involves scaling up to practice, which means that tests take place under realistic conditions, the level of realism could be increased after each round of testing (Wieringa, 2014). Wieringa (2014) proposes that scaling up two aspects would be needed to increase the street credibility of the created artefacts, these two aspects are:

1. Inductive generalisation, which is the sample size utilised.
2. Generalisation by analogy, which is the level of realism of the case.

The theory of Wieringa (2014) is visualised as a model in Figure 5. The research strategies selected for this study contribute to both dimensions. The survey research will contribute to the increase of the sample size and also gathers information to create a realistic case when validating the artefacts with the expert interviews. Gathering more realistic case to validate the artefacts will increase, with each case, the credibility of the study.

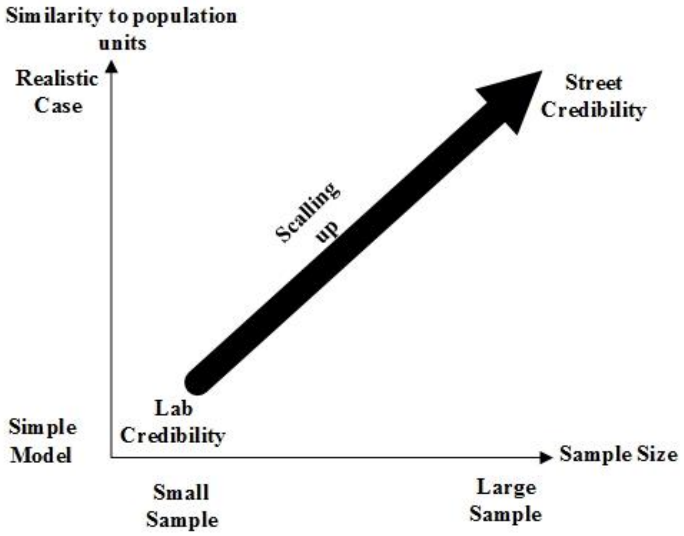


Figure 5 Inferences supporting design theories (Wieringa, 2014)

The expert interviews focus on validating the correctness of the different elements in the BRMS implementation framework, the applicability of the BRMS implementation framework, and the BRMS implementation framework as a whole. In this case, expert interviews are used for validating the correctness of the framework and its elements.

The evaluation of the created artefact can be found in Chapter 4.

Communication

Communicate the problem and the importance of the problem, the artefact, utility and novelty, the rigour of its design, and the effectiveness to researchers and other relevant audiences. In the communication stage, the designed artefacts are presented by means of the MBI colloquia, Thesis presentation and defence, and by a conference paper and presentation.

1.4.2 Research model

This section contains the research model, which is used to create structured, logical stages together with their input, throughput, and output throughout the research process. The research model for this study is visualised, and this is shown in Figure 6.As the research design science framework shows, in Figure 3, the study will go through several stages. Every stage will have assigned research goals to ensure relevance. Each stage will result in knowledge which can be processed in the knowledge base. The stages processed in Figure 6 are depicted from the Peffers et al. (2007) design science cycle shown in Figure 4 and include the activities: Definition of the objectives for a solution, Design and development, Demonstration, Evaluation, and Communication.

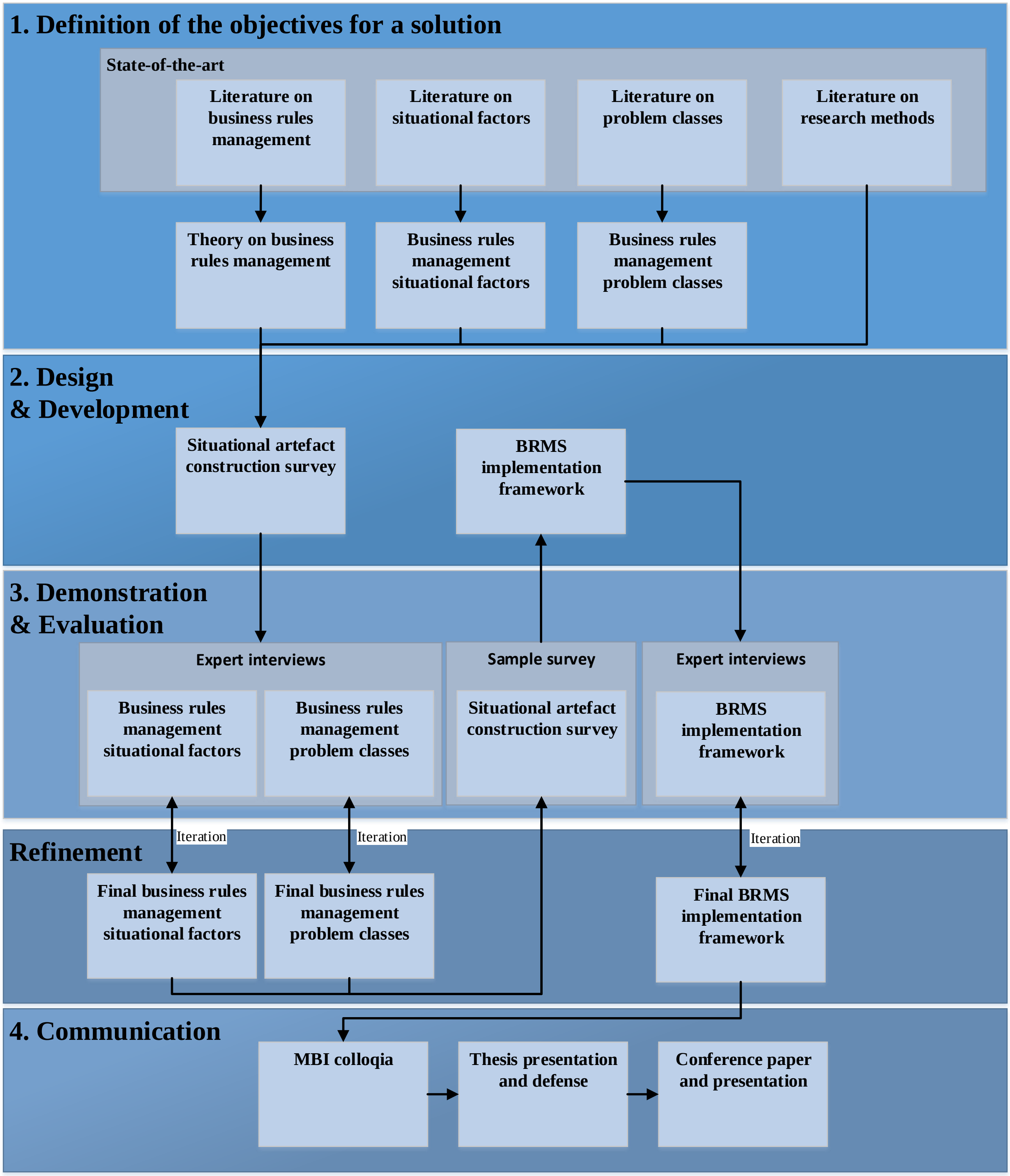


Figure 6 Research Model

1.5 Research scope

This section contains the research scope. The BRMS implementation framework will contain validated sets of problem classes and situational factors. Due to time constraints, this study focusses only on creating the BRMS implementation framework and validating this through expert interviews. During these expert interviews, the correctness of the construction of the framework is measured and redesigned if needed. Wieringa (2014) depicted the position of an expert interview on the scaling up scale and described that an expert interview is focused on a simple model and a small sample, as shown in Figure 7. Realistic scenarios and large samples are necessary to validate the BRMS implementation framework even more. Taking into account the work of Wieringa (2014), on designing an artefact with high credibility, more realistic cases and larger samples are needed when conducting future research. An example to achieve this in future research is to conduct case studies (Wieringa, 2014). Case studies are focused on realistic cases and larger samples, as shown in Figure 7.

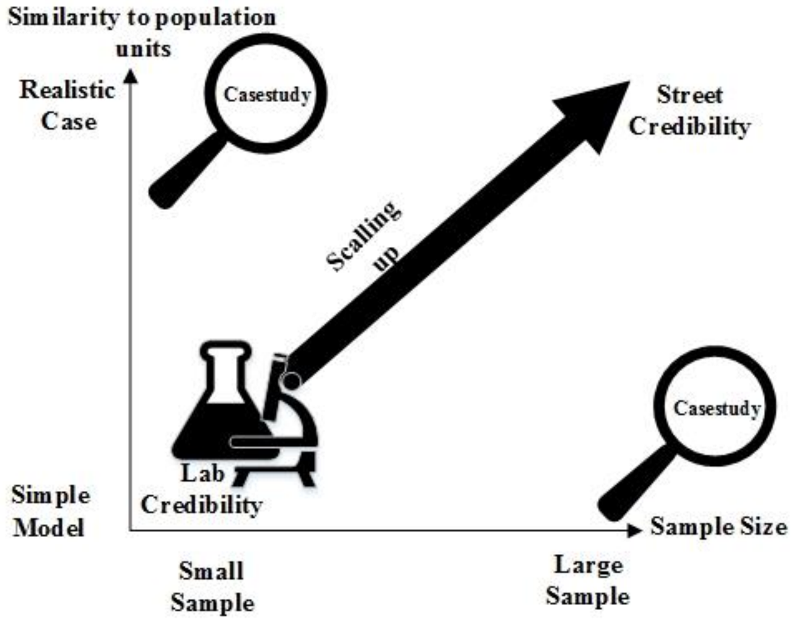


Figure 7 Research method positions on the scaling up to practice approach (Wieringa, 2014)

2 State-of-the-art

This chapter contains a state-of-the-art literature review to create an understanding of the existing knowledge on business rules management and situational factors. This is necessary to set a framework of the knowledge where we are working with in the latter stages of this research. Furthermore, this literature review defines the knowledge gap in the field of business rules management and what is needed to create a solution to fill this gap.

This literature review is performed to create a state-of-the-art literature overview as input for the design of the artefacts. This literature will contain knowledge from the neighbouring research fields and research fields which are comparable with the BRM research field. The state-of-the-art literature overview conducted for this research follows certain aspects of the process of a Systematic Litterature Review (SLR) described by Kitchenham (2004). Kitchenham describes requirements which ensure validity and rigorousness of the literature gathering stage. One of these requirements is the review protocol. The review protocol ensures the traceability and reuse of the gathered data for further research. Before conducting an SLR, it is important to define the review protocol. The review protocol contains:

- The background of the topic
- Which research questions are covered by the review
- The data sources used during the review
- The search strategy, which contains the search terms used
- The selection strategy contains criteria for including or excluding a study from, the literature review.

These requirements are also used for this research to ensure validity and rigorousness of the literature gathering stage. The activities for the literature review process for this research are shown in Figure 8.

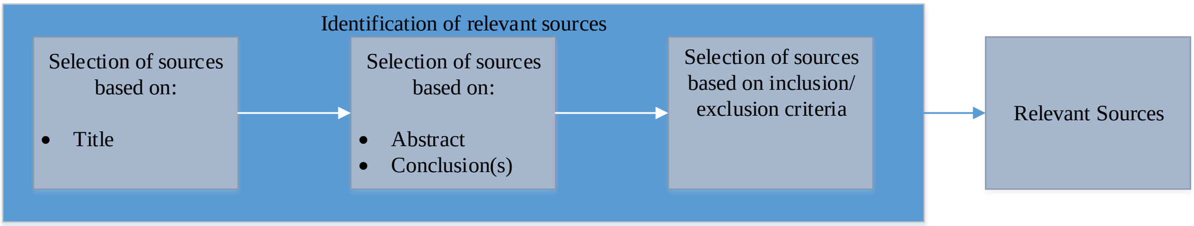


Figure 8 Literature review process

A semi-structured method is chosen for the identification of relevant sources. At first, the literature is selected based on the relevance of the title. The second step is the selection based on abstract and conclusion. The next step, if the sources pass the abstract and conclusion selection, is to compare the literature with the setup inclusion and exclusion criteria. These criteria included:

- Minimum of 100 citations:

  This criterion increases the impact of the chosen paper and increases the credibility of the research. This criterion is more set as a preference because it is not always a possibility for literature to have 100 citations, due to the research field maturity.

- Recent publications (start of 2011):

  This criterion is also more of a preference than criterion because of the immaturity of the research field and therefore possible the literature on this topic is scarce when using this criterion.

For this literature review, a wide range of academic search engines are utilised for literature gathering. These search engines were:

- ACM (Association for Computing Machinery) Digital Library
- Google Scholar
- IEEE Computer Society Digital Library
- Researchgate
- Springer Online
- Wiley Online Library

During this specific search, terms are used, and these are shown in Table 1. The tracking for search terms is done for reliability purposes and the possible repeatability of the study.

Table 1 Search term literature review

Business Rules Management	Situational Factors	Neighbouring fields on Situational Factors
• Separation of concerns • Separation of workflow • Separation of business logic • Business rules management • Business rules • Business rules management problem space • Business rules management problem classes • Business rules management solutions	• Situational factors • Situational factors in IS • Situational implementation • Situational implementation method	• Business process management • Software product management • Software engineering • Information systems research • Enterprise architecture

A breakdown structure of the literature review is created which contains the causal relationships between all the concepts which are going to be covered in sections of the literature review. The sections of the literature review are related to the research questions discussed earlier. The breakdown structure can be found in Figure 9.

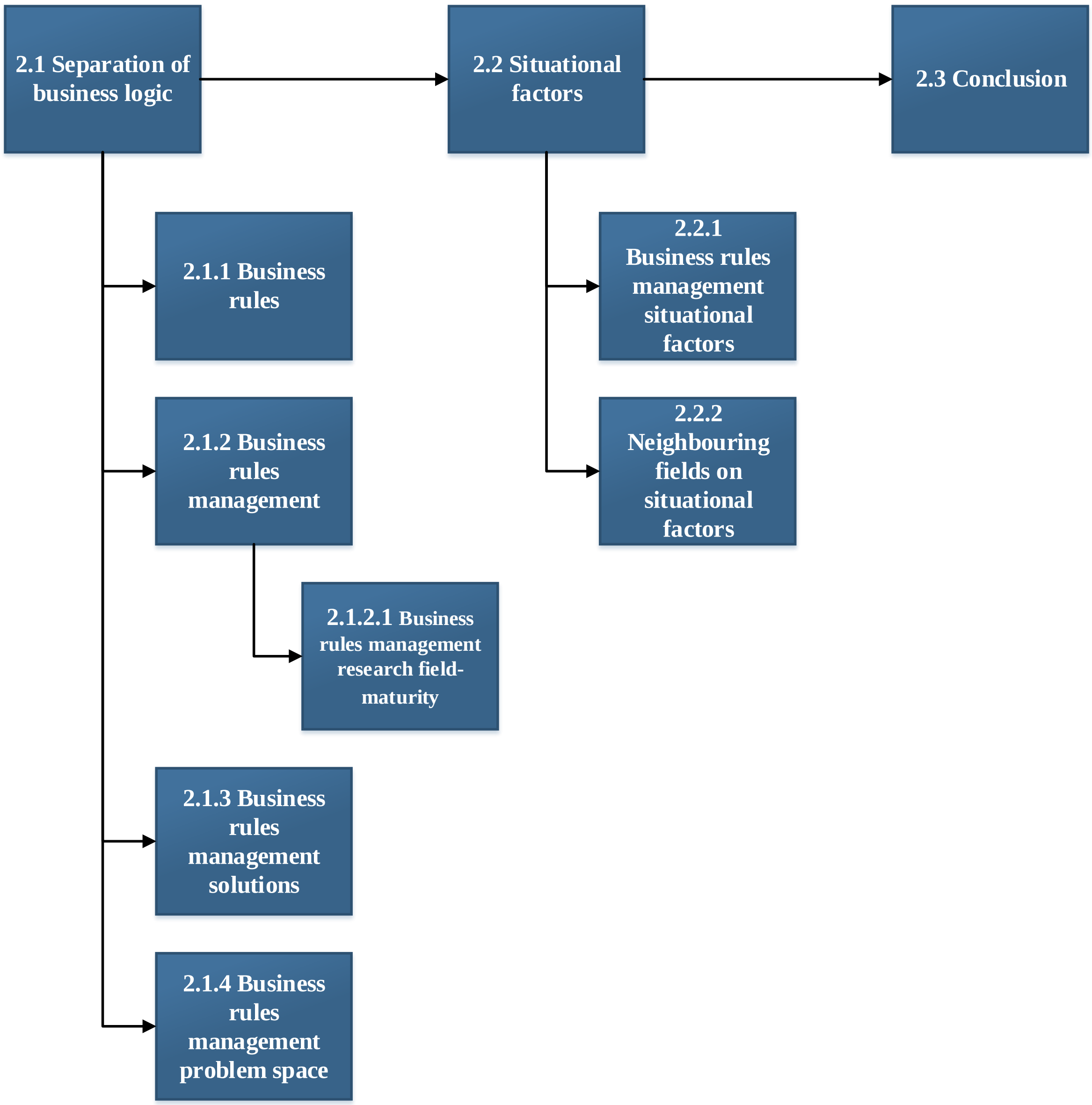


Figure 9 State-of-the-art breakdown structure

2.1 Separation of business logic

The main goals of software engineering are the improvement of software quality, to improve the facilitation of maintenance and evolution, and to reduce the cost of the production of software. Software engineers constantly search for development technology and methodologies that focus on reducing software complexity, improve the comprehensibility of the software, promoting the reuse of software, and facility the evolution of the software. In turn, these software engineering properties induce several specific requirements on the formalisms used to develop these software artefacts (Tarr, Ossher, Harrison, & Sutton Jr., 1999). These mentioned goals all contribute to the overall quality of software systems and software engineering in general. Completing the goals mentioned earlier are often conflicting in nature and thereby increasing the complicity of software engineering. Despite the research done in the software engineering domain, many of the mentioned problems still exist. These diverse problems are due, to limitation and unfulfilled requirements related to the separation of concerns (Dijkstra, 1982; Parnas, 1972).

Achieving the separation of concerns will ask the ability to manage the separate concerns important in the software engineering process. The separation of concerns moved to the workflow management field. Van der Aalst (1998) proposed in his work on Workflow Management Systems (WMS) the separation of concerns from the applications. Van der Aalst proposes that researchers should look back in time and evaluate how the separation of concerns evolved. Starting in the 1960's an information system consisted of a number applications standing on its own. An application-specific UI and database system had to be developed for each of these separate applications. In the 1970's data was pushed out of the applications. Database Management Systems (DBMS) were developed for the same purpose. By using this technology, applications were separated from the data management burden. In the 1980's a similar transaction happened for UI management. The appearance of the User Interface Management Systems (UIMS) enabled application developers to separate the user interaction from the applications. In the 1990's workflow software emerged, allowing application developers to separate the business procedures from the applications. This whole history perspective of WMS is shown in Figure 10.

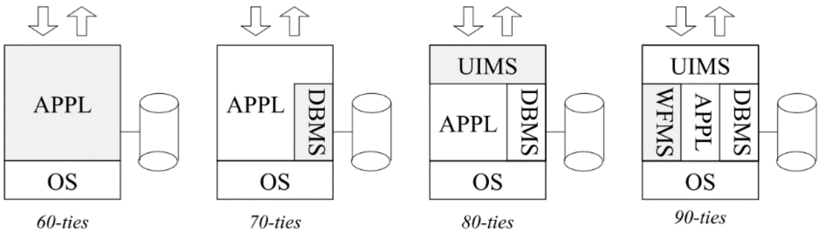


Figure 10 History of Workflow Management Systems (van der Aalst, 1998)

The separation of the workflow (van der Aalst, 1998) was a logical step, and the separation of business logic is a similar and comparable logical step in the separation of concerns. For this study the definition from Von Halle and Goldberg (2009) is adopted:

''Business logic is simply a set of business rules represented as atomic elements of conditions leading to conclusions.'' (Von Halle & Goldberg, 2009)

Business logic represents business thinking about how business decisions are made. An example of a business decision includes the decision on whether to grant a loan or not. Business logic represents the *''rules of the business''* (Von Halle & Goldberg, 2009). Due to the ever-changing business situations, changing policies and changing products and services, applications and services must be flexible to cope with this (Vanthienen, 2001). Flexibility is a major challenge for businesses especially when business logic contains business rules. The current situation is that many of the business rules are buried in programming code (Charfi & Mezini, 2004; Vanthienen, 2001; Von Halle & Goldberg, 2009; Zoet, 2014). Besides the business rules being hard coded into programming code Zoet (2014) proposes the possibilities that business rules can also be buried in the minds of the employees, business reports (such as but not limited to manuals and contracts), and business processes. Furthermore, the separation of business logic from applications should be considered because of the increase in a number of business rules, the frequency at which the business rules change, the different types of business rules and the necessity of business rules which support efficiency automation in the business (Boyer & Mili, 2011; Graham, 2007). The construction of systems requires flexibility, business rules, and the management of business rules in an organised setting. This could be achieved by bringing business rules into the business management side (Date, 2000; Graham, 2007; Morgan, 2002; Ross, 2003; Von Halle, 2002).

Chapin, Hale, Khan, Ramil, and Tan (2001) state that alongside the other concerns (workflow, database, applications and user interfaces) business rules changes are the most frequent and have the highest impact on software and business processes. The authors confirmed that the other concerns are depending heavily on the support of business rules and that the change of business rules are the most significant regarding the amount of work effort. Thereby, confirming the need separating business logic.

Hohwiller, Schlegel, Grieser, and Hoekstra (2011) argue why and how the business process, data, and business rules are separated. Hohwiller et al. (2011) state that these practices are good solutions to increase flexibility and efficiency. Important side note, it is important that these practices are seen independent from each other while complementing each other. With the use of the Business Process Modelling Notation (BPMN) (Object Management Group (OMG), 2011) an example of separation of business logic is created based on the work of Hohwiller et al., (2011). Figure 11 shows a process in which a business process is visualised and still containing the business rules.

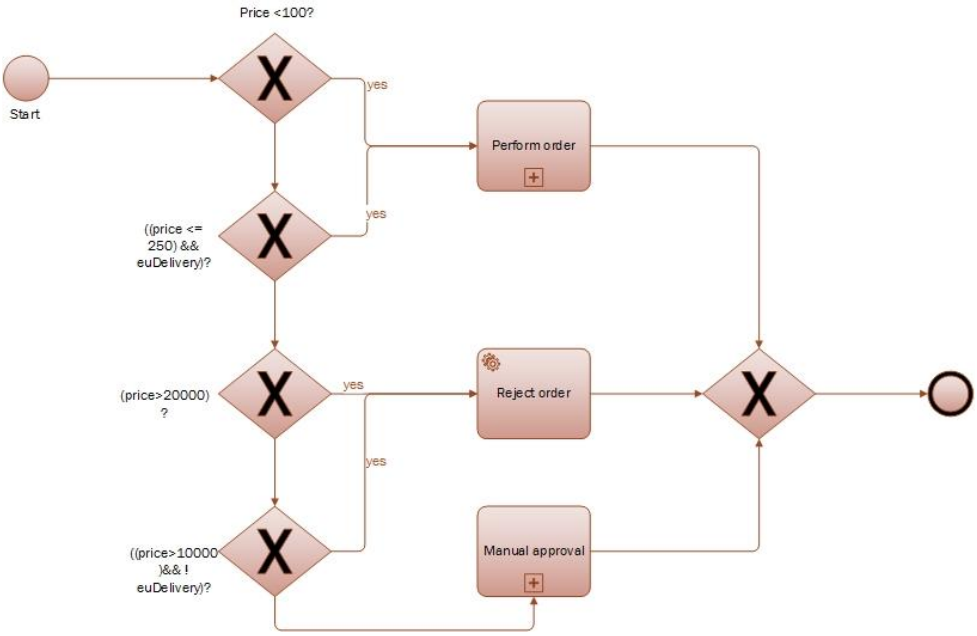


Figure 11 Business process containing business rules (Hohwiller et al., 2011)

The example shows that half of the business process is intertwined with business logic. This is an example of that separating business logic from a business process is high in work effort to separate this. Even at this stage research shows that even the content of the business rule will be separated due to the significant difference in change of the elements of a business rule (Zoet, Smit, & Leewis, 2015).

2.1.1 Business rules

Business rules describe the state of affairs what the business demands (Morgan, 2002) and are essential for business and technology models (Business Rules Group, 2003). For this research we adopt the definition of a business rule by Morgan (2002):

''a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business.'' (Morgan, 2002)

It is clear that Morgan gave priority to the business perspective and not the IT perspective, which is in the case of business rules, leading and IT a supporting role. Furthermore, the importance of a BRMS is shown in this definition and therefore this definition of a business rule is used throughout this research. Supporting his definition, Morgan gave examples of business rules which are as followed:

Example 1: A valid inbound Contact must be associated with a specific Customer.

Example 2: A withdrawal from an account may be made only if the account balance is greater than zero

A more complex business rule usually consists of multiple conditions, as shown as followed:

Example 3: A customer must be considered as high worth if at least one of the following is true:

- The Customer is a Platinum member,
- The Customer has a current balance of at least €100.000,-
- The Customer has held an account for more than 10 years.

Multiple types of business rules are defined in the literature and are used in practice. Von Halle (2002) defines business rules in five classifications:

1. Constraints are mandatory restriction or suggested restrictions on the behaviour of business events,
2. Guidelines are complete statements that express a warning about a true or false circumstance.
3. Action Enablers are complete statements that test conditions, and upon finding these conditions true it initiates another activity,
4. Computations are complete statements that provide an algorithm for arriving at the value of a term. Such an algorithm can contain for example a sum, minimum, maximum, etc.
5. Inferences are complete statements that test conditions and upon finding these conditions true, establishes the truth of a new fact.

Other classification of business rules published by other authors (Ross, 2003; Zoet, 2014) are comparable or overlapping with the classifications made by Von Halle (2002). Taking into account all types of business rules defined in literature and practice it would not fit within the scope of the research, and therefore we only focus on explaining the concept of what a business rule is.

Date, (2000), Graham (2007), Morgan (2002), Ross (2003) and Von Halle (2002) proposes the management of business rules in an organised setting, this could be achieved by bringing the business rules into the business management side.

2.1.2 Business rules management

Business Rules Management (BRM) is defined as: *''a systematic, and controlled approach to get a grip on business decisions and business logic to support the elicitation, design specification, verification, validation, deployment, execution, governance, and monitoring of both business decisions and business logic''* (Smit, Zoet, & Berkhout, 2017). BRM is usually centred around the activities of elicitation, design, verification, validation, deployment, execution, audit, or the monitoring of business logic and business rules (Zoet, 2014). To tackle previously mentioned challenges and to improve the grip on business rules organisations search for a controlled approach to support the previously mentioned activities, this approach is called BRM (Boyer & Mili, 2011; Ross, 2003). The mentioned activities are part of a process which is translated into products or services through internal policies, external laws and regulations.

The elicitation of business logic from employees and other sources is an important characteristic of BRM. This activity ensures that the business logic is explicit and could be stored for future use. This business logic could be utilised for example to train new employees to help them in their daily activities (Zoet, 2014). In order to use the business logic for future use is to transform the business statements to IT statements. This action ensures that the systems working with the business logic could understand the given business logic (Morgan, 2002). The business statements are relatable with natural language and describe how businesses operate or deal with certain situations. The IT language is interpretable by IT systems and is in a more formal form.

Boyer and Mili (2011), and Jones (2012) describe the added value of BRM for the business. Three elements arise and are described as followed:

Differentiation:

BRM enables easier distinction between different groups of people. A typical example is to use business rules for different types of customers (high, medium and low valued customers) and then define other conditions apply to these categories, for example, applying different levels of service to them. Exposing these policies as business rules makes them more flexible and visible;

Accessibility and availability:

Ensure accessibility and availability to support staff if business rules are coded in programming logic, it is rarely easy to change them quickly - in part because it can be difficult to determine where a particular business rule is located. Abstracting rules in a BRMS makes them more accessible and available to change if needed;

Agility:

Once business rules have been made explicit and accessible, they can easily be modified. For an organisation that is accustomed to waiting on changes to an overburdened IT function, this advantage may also constitute a governance-related challenge. It will be important to ensure that business rules are tested prior to implementation.

The earlier mentioned activities of BRM need to be supported by some sort of IT system. This can be done in many ways, but the most economical is to use a business rules management solution (Graham, 2007).

2.1.2.1 Business Rules Management research field-maturity

This section will address the needs of the level of maturity of the business rules research field. Edmondson and Mcmanus (2007) address that research fields have three levels of maturity and can be classified as nascent, intermediate and mature. Edmondson and Mcmanus (2007) define these three level of maturity by the state of prior theory and research. Zoet (2014) states in his work that the Business Rules Management field is in the nascent stage and as can be derived from the motivation section. The dissertation of Zoet (2014) was the last published work that contributed something to the level of maturity. Every level of research field maturity has different focused to the best fit when it comes to problem-solving. In this level of maturity in the research field for BRM, research should focus on “Provisional explanations of phenomena, often introducing a new construct and propose relationships between it and established constructs” (Edmondson & Mcmanus, 2007). The BRM field still lacks knowledge on the organisational implementation of the business rules management solution, no validated sets of situational factors or problem classes and no knowledge on the relationships between these two concepts.

2.1.3 Business rules management solutions

A Business rules management solution (BRMS) is a set of software components for the elicitation, design, specification, verification, validation, execution, monitoring, and governance of business rules and is a configuration of the capabilities elicitation, design, specification, verification, validation, deployment, execution, monitoring, and governance (Smit & Zoet, 2016; Zoet & Versendaal, 2013).

1. **The elicitation capability**

   The elicitation capability is a bifold function. The first function is to determine the knowledge which realises the value proposition of the business rules. This knowledge needs to be captured from various sources including but not limited to laws and regulations. The second function is the to initiate an impact analysis, this is only done when a business rule architecture is already in place.

2. **The design capability**

   The design capability creates the non-platform specific rule system. The output of the design capability is the business rule architecture. The business rule architecture contains a combination of derivation structures and context designs.

3. **The specification capability**

   The specification capability specifies the content of each separate context design. The function of this capability is to create the business rules and fact types needed to constrain or define particular aspects of the business.

4. **The verification capability**

   The verification capability verifies the created business rule architecture to check for semantic and syntax errors.

5. **The validation capability**

   The validation capability reviews the created value proposition. The goal of this capability is to check for possible errors in its expected behaviour.

6. **The deployment capability**

   The deployment capability transforms the verified and validated value proposition to implementation-dependent executable business rules. The actor that utilises the value proposition is not necessarily a system; a subject-matter expert could also utilise this.

7. **The execution capability**

   The output of the deployment capability is then executed in the execution capability, which delivers the actual value proposition. To realise the added value, human or information system actors execute the business rules.

8. **The monitoring capability**

The monitoring capability observes, checks and keeps a record of not only the execution of the value proposition but also the full range of activities in the previously explained BRM capabilities that are conducted to realise the value proposition.

9. **The governance capability**

All capabilities provide output for the governance capability. Data collected about realising changes to a specific input, output and other capability elements are registered, changes made to the data source, platform specific rule models, non-platform specific rule models, and all other input and output are registered.

The governance capability consists of three sub-capabilities; version management, traceability, and validity management. The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities. The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time. The purpose of the traceability capability is to ensure the possibility to trace created elements to their corresponding laws and regulations.

Each implementation of a BRMS is one capability or a combination of capabilities. These BRMS capabilities give the business the ability to make changes and updates to the business rules that drive these capabilities and frees up resources for IT to tackle other high-value problems. A BRMS provides the support of the following (Taylor, 2011):

- The development of business rules
- The testing of business rules
- Linking business rules to data sources
- Identifying conflicts and quality issues regarding business rules
- Measure business rules and decision effectiveness
- Reporting of the business rules and decision effectiveness
- The deployment of the business rules to different computing environments.
- Business rule maintenance after the business rules are deployed
- The integration of business rules with other application and services

Using business rules requires the business rule being written in a specific format (the earlier mentioned IT language), so these can be exploited for knowledge or communication. Business rules are automated by 1) hard-coding into existing IS, 2) using a business rule engine, or 3) using a BRMS (Zoet, 2014). A BRMS, in its most basic form, consist of two parts: a rule engine, which is an inferencer, and a business rules repository (Boyer & Mili, 2011). For this study, there is no need to dive any deeper into the technical aspects of a BRMS because this is out of the scope of the research. The technical implementation of a BRMS is covered extensively in research (Arnott & Pervan, 2005; Rosca & Wild, 2002; Zoet, 2014). Analysing the literature on organisational implementation of a BRMS it comes to mind that there is not a lot published and if published no real standards and frameworks are provided (Nelson et al., 2008; Zoet, 2014).

Limited research has been conducted in the business rules fields and thereby consistent reasoning about business rules, and BRM is limited (Zoet, 2014). To approach this problem, the problem space concept is used. A problem space is a set of similar design problems for which solutions need to be designed (Winter, 2011a). Zoet (2014) and Zoet and Versendaal (2013) suggest that a BRM problem space is required that can capture and position instantiations of BRM.

2.1.4 Business rules management problem space

The earlier mentioned BRMS is stated as singular problem-oriented, meaning that a BRMS is designed to solve one problem (Liao, 2005; Wagner, Otto, & Chung, 2002). Recent research contradicts the singular problem orientation and proposes that different BRMS have a common design problem (Aier et al., 2008; Baumöl, 2005; Bucher & Winter, 2010; Klesse & Winter, 2007; Lahrmann & Stroh, 2009; Leist, 2004). Common design problems indicate that problem classes exist. Therefore, design solutions can be created (Simon, 1997; Winter, 2011b).

A problem space can contain one or more problem classes. An instantiation of a specific problem class in a specific organisation is defined as a design solution (Winter, 2011a; Zoet, 2014). In the BRMS problem space the design solution is a specific configuration of the earlier mentioned nine capabilities (Smit & Zoet, 2017b; Zoet, 2014). The problem space and design solutions are influenced by situational factors as described by Winter (2011b). The context in which the organisation of artefact has to operate is described by the situational factors. More on situational factors later on in this literature review.

To create a clearer overview and explain the difference between the problem space, problem class, design situations and situational factors the Chinese house example is adopted by Winter (2011a, 2011b) and later used by Zoet (2014). The example is shown in Figure 12.

The problem space, in this case, is *''constructing a Chinese style house''*. The problem space, constructing a Chinese style house, is divided into problem classes by situational factors, described in the work of Winter (2011b). An example of such situational factor is the foundation and the framing of the house reducing the choice of houses (degree of freedom) thereby driving the problem classes. The situational factor can again further specify the problem classes. Thereby, differentiating in the levels degrees of freedom can occur. The problem classes of constructing a Chinese style house are defined when no further reduction in freedom can occur. Thereby, ensuring that every problem class represents a design situation of the Chinese style house that can be built. The design situation itself can also be influenced by situational factors. An example here is, if the problem class Chinese house B state that the roof and windows must be square shaped it does not imply anything about the material that should be used during the construction. This is different with each house, as is shown in Figure 12. The situational factor is, in this case, the material that is used. Thereby, influencing the construction of the house. The situational factors that affect the problem space at the minimum number of situational factors necessary to classify a problem class. Thereby, this minimum number of situational factors is defined as the classification freedom of the problem space (Zoet, 2014). Therefore, the situational factors that impact the reduction of freedom of a problem class is present in all instantiations of the design situations, whereas the situational factors that only affect the design situations itself are not.

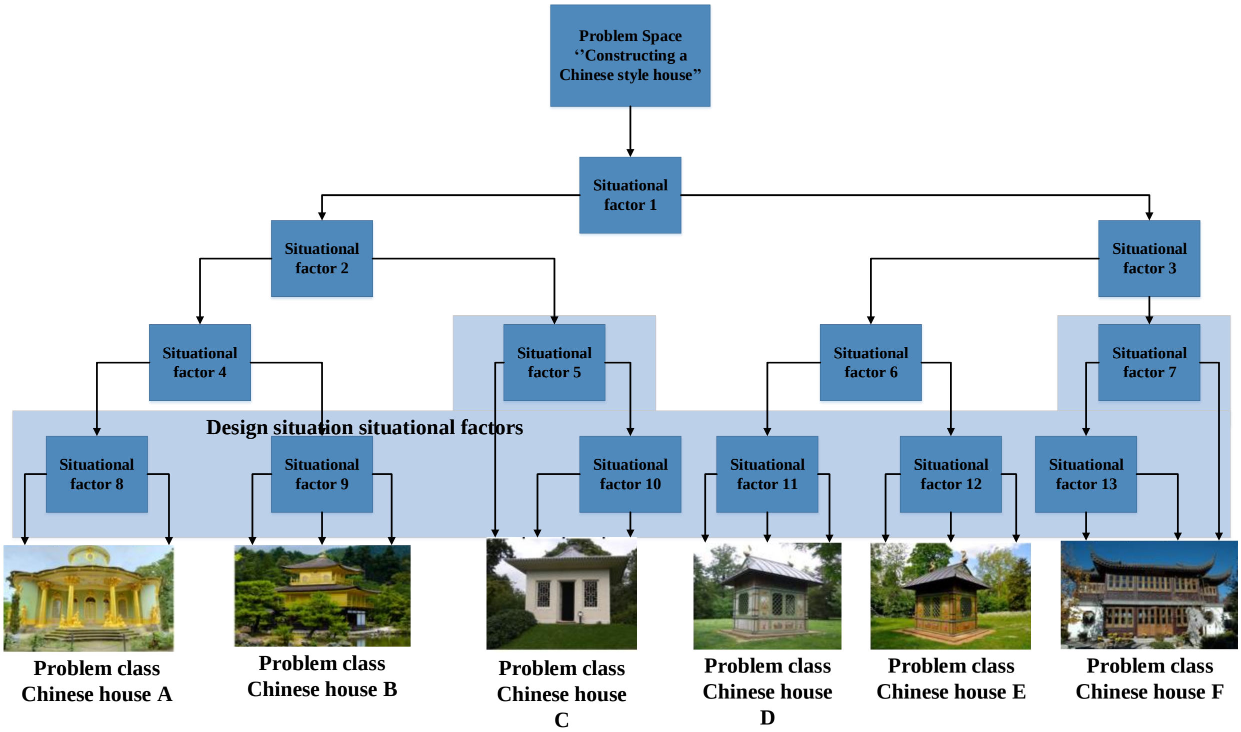


Figure 12 Problem space "constructing a Chinese style house"

Winter (2011b) proposed in his work a technique to create situational artefacts. The goal of this research is to create a BRMS implementation framework which could be used for implementing BRMS from an organisational perspective. In other words, situational artefacts are created. This technique contains the following steps:

(1) Initial demarcation of the design problem class

Describe a rough idea on the delineation of the design problem class is developed. Results of this step are used definitions, a description of the system in the context of the analysis and a notion of the design goals used for this class of design problems

(2) Identification of potential contingency factors

A literature analysis is conducted whit a purpose to identify potential contingency factors for that class of design problems.

(3) Field study based analysis of design problems in practice

A field study is conducted in order to analyse design problems of that class in practice. A result from this is that the list of potential contingency factor candidates is reduced to a smaller more relevant set of "design factors". Design factors might be aggregations of several contingency factors that need to be semantically interpreted. A factor analysis would contribute to an in-depth understanding of the retrieved data.

(4) Refining specifications of the design problem class

The design problem class is redefined by specified value ranges for the design factors.

(5) Calculation of ultra-metric distances

The field study data of observations which still belong to the redefined design problem class are used to calculate ultrametric distances between specific design problems. The calculation is based on certain ‘similarity’ metrics. A cluster analysis would fit these specifications.

(6) Determination of a useful level of generality

A level of solution generality is determined. Generally, clustering errors related to the number of clusters are used for this analysis.

(7) Specification of design situations

By using the desired solution generality, the resulting design situations are specified. The situations should be specified formally (by value ranges of the design factors) and by semantic interpretation (“design problem types”). The observations are grouped together in consideration with the design factors identified earlier.

(8) Identifying characterising design factors

The specified design situations consist of design factors, these need to be specified further. Every design factor has different values, these values influence the characterisation of the design factor and thereby the design situation.

(9) Linking design factors to design problems

The design factors described in the earlier sections need to be linked to design problems which are known identified earlier. All the earlier conducted steps analyse the existing design solutions. These design solutions were created with a certain purpose, and therefore the design factors should be qualitatively interpreted and linked to the known design problems,

(10) Deriving elementary problem-solving actions

The ideal next step is that of deriving elementary problem-solving actions by comparing design solutions with design problems. Out of these elementary problem-solving actions, method fragments are created.

(11) Deriving method configuration rules

Based on the set of identified design problems and specified method fragments, method configuration rules need to be derived. The method fragments identified in the earlier sections need to be related to respective design situations and which design problems these method fragments solve.

2.2 Situational factors

This section contains the literature findings on the concept situational factors, situational factors in the business rules management field, different types of situational factors, situational factors in neighbouring fields and the knowledge gap in business rules management situational factors. The situational factor has its origins in the field of social sciences but is already well-known in the information systems field. Situational factors influence the context in which an artefact or organisation has to operate, in such a manner that the deployed artefacts fit the environmental context (Zoet, 2014).

Brinkkemper (1996) described that a situational factor is any factor relevant to product development and product services. Van de Weerd (2009) describes in her research that situational factors can be expressed in a quantitative or qualitative manner. Examples of quantitative and qualitative situational factors are:

quantitative factors:

- Company size;
- Submitted requirements per month;

qualitative factors:

- Company sector
- Development method in use

Other research differentiates situational factors in other types, besides the quantitative and qualitative. The situational factors have been structured using the dimensions of the ontological foundations of information systems framework proposed by Weber (1997) and further extended by Strong and Volkoff (2010). The framework depicts four types of situational factors which occur in IS research, these types are:

Deep structure situational factors:
The deep structure situational factors are described as scripts that provide some form of representation of a real-world system, the system properties, the system state and system transformation (Wand & Weber, 1995; Weber, 1997).
Organisational structure situational factors:
The organisational structure situational factors represents the roles, control and culture of the organisation or within the solution (Strong & Volkoff, 2010).

The physical structure situational factors:
The physical structure situational factors are describing the physical technology and the software in which the deep structure is surrounded with (Weber, 1997).

The surface structure situational factors:
The surface structure situational factors are describing the interface between the IS and the users (Weber, 1997).

These situational factors are used in further research in the construction of artefacts in the fields of Business Rules Management (Zoet & Versendaal, 2013), Business Process Management (Bucher & Winter, 2010; Ravesteyn, 2011; Ravesteyn & Jansen, 2009) Software Product Management (Bekkers et al., 2008; van de Weerd, 2009) and, Enterprise Architecture (Klesse & Winter, 2007).

2.2.1 *Business rules management situational factors*

As described in the sections earlier, limited research is conducted in the business rules field. Therefore drastically limiting the literature available on situational factors in the business rules domain. Zoet (2014) conducted research focused on identifying the situational factors that characterise BRMS problem space. A total of 39 BRMS were analysed spread over multiple industries. The sample distribution is shown in Table 2. The choice of separating into these five industries is made for the difference of external factors influencing each industry in a different way.

Table 2 Sample distribution research conducted by Zoet (2014)

Industry	Number of BRMS
Financial	11
Medical	4
Transport	1
Government	19
Remainder	4
Total	39

Out of these cases, six situational factors were identified. These situational factors are a value proposition, approach, standardisation, change frequency, n-order compliance, and the integrative power of the software environment (Zoet, 2014):

1. **Value proposition (VP)**
The value proposition is the first situational factor that is defined. Analysing the 39 BRMSs did indicate a significant number of different value propositions. Zoet (2014) compared the different value propositions and created three generic value propositions. The three different values for the value proposition situational factors are in line with the types of situational factors of van de Weerd (2009):

   A) Guidance (constraint),

   Guidance elements describe constraints with some form of guidance regarding the business entity behaviour. This value proposition can be applied to a broad range of application areas and business rule statements. A business entity can be anything of value to the business.

   B) Communication,

   The value proposition communication is realised by describing the characteristics of the business entity and the relationships of the business entity with other business entities. The business entity can be defined as a student card is an authorization for the bearer that he/she is enrolled at a university. Therefore, a student card belongs to a person.

   C) Decisioning.

   Decisioning describes conditions evaluating business facts that lead to a conclusion. The application of this statement depends on the application area. When applied for the assessment of decisioning business rules are used to formulate a decision. However, when decisioning is applied to monitoring the business rules are used to formulate norms.

2. **Approach (A)**
Approach is the second situational factor that is defined. The choice for a specific approach determines the model abstraction needed. This results in a reduction of applicable capabilities for the BRMS. The analysis revealed three different values for approach proposed by Zoet and Versendaal, (2013). Nelson, Peterson, Rariden, and Sen (2010) identified the same values but through a maturity model perspective. The three values for Approach are:

A. IT-oriented value,

The IT-oriented value stresses on enactable platform specific rule models. Enactable specific rule models are models that can be executed by physical hardware or software. The output of the capabilities is IT-related products such as design documents (technical and functional).

B. Business-oriented value,

On the other hand, the business-oriented value is focused on realising non-platform specific rule models. Non-platform specific rule models serve mainly for simulation and communication

C. Balanced value.

The balanced value fills that gap between both worlds. In the balanced value, the business units develop the non-platform specific rule model where the IT department translates it to enactable platform-specific rule models.

Nelson et al. (2010) classify in their work the IT value as the lowest level and the business, and balanced value are classified as the highest. During the identification in BRMS, the other way around is also recognised.

3. Standardisation (S)
Standardisation is recognised as the third situational factor. The analysis identified two different values for standardisation:

A. standardised modelling language (i.e. BPL and OPA),
B. non-standardized modelling language.

4. Change frequency (CF)
Change frequency is the fourth situational factor. The change frequency of business rules affects the BRMS organisational structure. Change frequency indicates the number of times business rules change which we classify as:

A. low,
B. medium,
C. high.

When the change frequency is classified as high, it is requisite to set up processes, roles, input and output for the required capabilities, audit and version.

5. n-order compliance (NC)
The fifth situational factor that is identified is n-order compliance. This is used to measure a number of actors between for example the creator of a law and the actual implementation by means of business rule models. Only one role within organisations has the power and the knowledge to provide that defines the regulation, first order compliance. This actor can achieve this by translating the law into a business rule model or by validating the business rule model created by other actors. Not for all situations, it is possible to do first order compliance. Therefore, the capabilities (design, verification and validation) needed to be designed differently.

6. The integrative power of the software environment (IP)
The sixth situational factor is the integrative power of the software environment. Multiple software functions are needed to support the different aspects of a BRMS and can be integrated into one specific software package or distributed across multiple software packages. The integrative power of the software environment measures the distribution of functions needed for the BRMS. The analysis revealed two values:

A. Integrated,

An integrated software environment provides software functions for one or multiple capabilities within one software package

B. Non-integrated.

A non-integrated software system delivers functions to support only one capability.

These six identified situational factors discovered by Zoet (2014) are plotted over the situational factor types defined by Strong and Volkoff (2010), and Weber (1997) which can be seen in Figure 13.

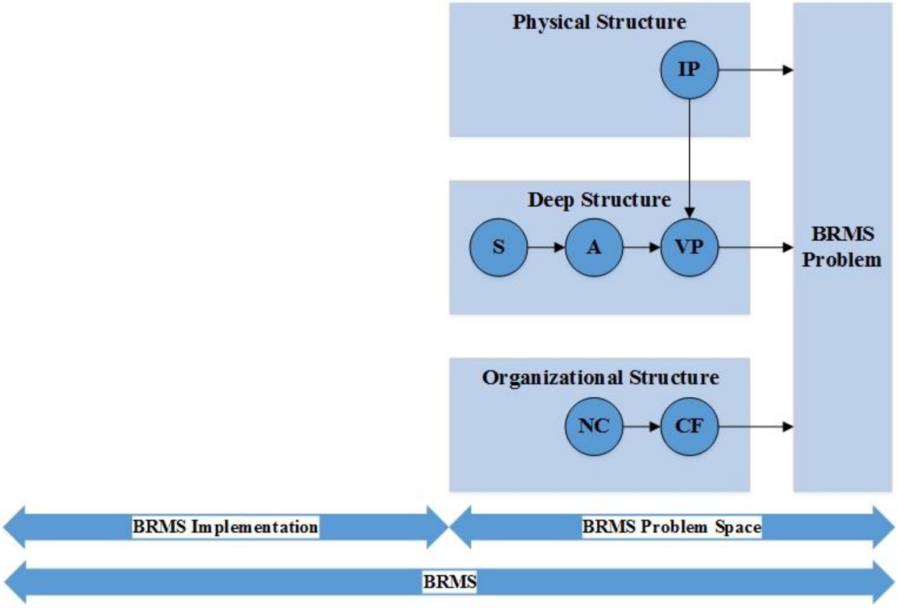


Figure 13 BRMS situational factor relations

Wang, Indulska, and Sadiq (2016) recapped in their work twelve factors that are thought to be important and potentially influence the decision on how business rules should be modelled. Empirical study with academic experts a set of twelve factors were identified together with its level of importance. In descending order the importance of the set of factors is as followed (Wang et al., 2016):

1. Agility refers to at what speed a business rule can be adapted to change.
2. Criticality refers to the level of importance of the business rule
3. Rate of change refers to the frequency level at which a business rule is to be revised.
4. Reusability refers to the potential that a business rule has to be used in new contexts.
5. Accessibility refers to the user's ability to view and manipulate a business rule.
6. Awareness of Impact refers to how comprehensively the implications of a business rule are understood.
7. Complexity refers to the difficulty level in defining or understanding a business rule.
8. Governance Responsibility refers to who ensures that related business activities are in compliance with business rules.
9. Scope Impact refers to what the level of impact is the business rule (activity, process or department level).
10. Aspect of Change refers to the component of the business rule that can be changed.
11. Implementation Responsibility refers to who is in charge of the implementation of the business rule.
12. Rule source refers to the origin of the business rule, for example, laws or regulations.

2.2.2 Neighbouring fields on situational factors

Due to the rather immature state of the BRM research field, neighbouring fields or similar research fields are analysed on the use of situational factors. The fields of Software Product Management, Business Process Management, and Enterprise Architecture are analysed on the use of situational factors.

2.2.2.1 Situational factors in Software Product Management

The BRMS being a software product and therefore the field of Software Product Management (SPM) is analysed on the use of situational factors. The work of Bekkers, van de Weerd, Brinkkemper, and Mahieu (2008) is focused on finding out which situational factors has influenced the selection process of method fragments. Thereby, enhancing SPM by adapting the tuning selection method fragments to the specific company environment. The situational factors are provided by literature and by knowledge communities. The researchers held interviews to retrieve weights (Likert-scale 1-7) on which situational factor is the most important in the eyes of the interviewee. A Method Engineering approach was used for the analysis of the used method fragments used by the interviewees. Thereby, resulting in a list of situational factors focused on the business unit characteristics, customer characteristics, market characteristics, product characteristics, and stakeholder characteristics of the SPM process. During the development phase, the situational factor model is created which exists of the weight for each situational factor and the applicability to label SPM method fragments with each situational factor. Domain experts from practice and science validate the results of the development phase and this result in a refined model. Future research is needed to validate the list of situational factors further and the increase of the case study sample to improve the generalizability of the study.

2.2.2.2 Situational factors in Business Process Management

Ravesteyn and Jansen (2009) did a comparable study to this study in the field of business process management (BPM), implementing a business process management system. The BPM field contains many different implementation methods but lacks a situational implementation approach. Examples of these implementation methods are Six Sigma (De Feo & Barnard, 2003), the Strategy Driven Approach (Jeston & Nelis, 2014) and the BPM implementation method (Burlton, 2001). Because of the different variables in organisations, various implementation methods are needed (Bucher & Winter, 2010; Ravesteyn & Jansen, 2009). Ravesteyn and Jansen (2009) proposed a context BPMS implementation method consisting of the critical factors of the earlier mentioned implementation methods. Examples of situational factors used in the context BPMS implementation method are the project plan, best practices and IT infrastructure. The future research asks for more content for the method, only containing 14 implementation fragments. Possible additions could be more activities and culture context. Besides the addition of new content more validation of the method is needed, not only for each fragment but for the entire implementation method.

Bucher and Winter (2010) focused on creating a more generic method to fit in more situations. Thereby, adapting, if necessary, to each given situation. A survey was conducted to create a more and deeper understanding of the problem situation. In the first steps of the survey, four distinct BPM design factors were identified. These factors were: (1) the degree of process performance measurement, (2) the overall professionalism of process management, (3) the process manager impact, and (4) the utilisation of established methodology and standards. Four different generic approaches were created on with these factors as a base. When the BPM approaches together with the BPM plans are used in the future, five BPM project types are differentiated. This all was merged into a method and has now the possibility the be adapted when needed (Bucher & Winter, 2010). Future research is required to validate the BPM design factors, the generic approaches to BPM, and the project types. The usefulness of the proposed situational method engineering approach needs to be justified by the use of more case studies.

Chapter conclusion

The technical implementation aspects in the business rules domain are researched extensively (Arnott & Pervan, 2005; Rosca & Wild, 2002) compared that to the organisational implementation aspects (Nelson et al., 2008). Superficial business rules organisation knowledge is present which include: knowledge of what business rules are and what BRM is, what the added value of business rules and BRM is and knowledge of the so-called business rules approach. The focus of this research will lie in the lack of knowledge on how to organisationally implement a BRMS, specifically the business rules management problem class.

The preceding sub-sections showed literature on situational factors and their role in business rules management, neighbouring fields such as SPM (Bekkers et al., 2008) and BPM (Bucher & Winter, 2010; Ravesteyn & Jansen, 2009). The neighbouring fields discovered validated sets of situational factors in their respected fields. The literature showed the urge to a *''one size fits all''* solution, and the same literature concludes that this is an impossible goal, for now, because of the context depended on organisations. The work of Winter (2011b) shows a technique which is used for the discovery of situational artefacts. The technique for the discovery of situational artefacts could therefore by utilised for the purpose of this research. Zoet (2014) discovered several situational factors in the BRMS problem space which could be used as a guidance tool for the discovery of situational factors in the BRMS implementation problem space. Wang, Indulska, and Sadiq (2016) discovered a set of twelve factors that only influence how a business rule should be modelled. Therefore, revealing the gap in the business rules management situational factor knowledge. Revealing the lack of situational factors in the business rules management solution implementation problem space.

The current BRM research field maturity request for the focus on the exploration of concepts (BRM problem classes and BRM situational factors) and the discovery of the relationships between these concepts. (Edmondson & Mcmanus, 2007). The BRM field still lacks knowledge on the organisational implementation of the business rules management solution, no validated sets of situational factors or problem classes and no knowledge on the relationships between these two concepts, a knowledge gap is identified. This implies that the research will go towards the exploration of concepts (BR problem classes and BR situational factors) and discover the relationships between these two concepts.

Therefore, the research will aim at constructing validated sets of situational factors and problem classes in the BRM domain. These sets of situational factors and problem classes will be constructed into a framework supporting the implementation of BRMS.

The next chapter will focus on the design of the BRMS implementation framework, existing of the BRMS observation technique, the BRMS construction process, the BRMS metamodel, and the BRMS metamodel support tool.

3 The BRMS implementation framework

This chapter focusses on the creation of the elements of the BRMS implementation framework. The observation technique to elicitate BRMS implementation cases from organisations. Thereby, elicitating design problems and problem classes. The BRMS construction process utilises the BRMS implementation cases and analyses the data to specify design situations and design factors. The BRMS observation technique and the BRMS constructions process result into the identification of the elements of the BRMS metamodel. The BRMS metamodel shows the relations of the identified elements together with each their specific values present when implementing a BRMS. Model-driven development (Eclipse GMF) is used to create a tool to support the BRMS metamodel. The BRMS metamodel support tool supports the elements of the metamodel in a model-driven environment. Furthermore, an 'How To' is shown to successfully guide the user of the BRMS implementation framework.

3.1 BRMS observation technique

This section is focused on designing and implementing the BRMS observation technique. The BRMS observation technique aims at gather successful BRMS implementation cases in the public and financial sector. In this research, a questionnaire is used to gather BRMS implementation cases, which aims to elicitate design problems and problem classes in the BRMS domain.

3.1.1 Designing the BRMS observation technique

This subsection will focus on the construction of the BRMS observation technique and the collection of the data utilising the BRMS observation technique. The situational artefact construction technique of Winter (2011b) requires a field study to gather design solutions for the creation of a situational artefact. This research utilises a survey for the gathering of BRMS implementation cases, also known as the BRMS observation technique. The goal of this survey is to collect as much data as possible on different BRMS implementations distributed over different sectors. These collections of different implementations can create an overview of clusters of situational factors that influence the different problem classes in the business rules management solution problem space. Thereby, creating a situational artefact for each different sector and possibly for more instantiations in these different sectors.

The questionnaire is constructed with a combination of existing literature and by use of experts from the BRM community. The experts are chosen on their experience and knowledge in the field of BRM and BRMS. The experts consisted of a professor lecturing and performing research in the field of BRM and BRMS, a lecturer and PhD with practical and research experience in the field of BRM and BRMS, and a student with 3 years of practical and research experience on BRMS capabilities. All the interviews were conducted in the same controlled environment and each interview had a length of 90 minutes.

The following sections will elaborate more on the existing literature supporting the construction of certain questions in the questionnaire. Certain questions are specific for each section and can only occur when dealing with these elements, but there are questions which are more general. Therefore, these questions are frequently used in different sections. There are possibilities that sections only exist of the general questions, the reason of this is that the focus lies on the specifics of a BRMS implementation and not an information system in general. Therefore, no relevant questions exist to complement the more general questions.

Furthermore, this section will contain the questions that are created by the practical experience of the BRM expert or by use of existing literature. Therefore, general questions are not discussed and can be found in the complete questionnaire in Appendix B.

3.1.1.1 *Participant information*

The participant information section intends to retrieve specific participant data and some general situational artefact construction questions. The questions asked ranged from what the sector is in which the BRMS is implemented, the employee range of the organisation where the implementation is carried out and the scope of the BRMS implementation. The complete questionnaire can be found in Appendix B (the more general questions that don't need any additional explanation can also be found there). The questions aimed at retrieving participant information are, further on in this research, used as situational factors of implementing a BRMS.

Q1:In which sector is the BRMS implemented?

Public
Financial
Other...

The sole reason for selecting these sectors (public and financial) is that, for now, we can say with confidence that sufficient participants can be gathered to have significant results out of the questionnaire.

Q2:How many employees work in the organisation where the implementation is carried out?

< 50
50 - 100
101 - 250
251 - 500
501 - 1000
1001 - 2000
2001 - 5000
> 5000

Q2 intends to retrieve the employee range of the organisation at which the implementation is conducted. These employee ranges could influence different implementation setups. Example: Organisation A with <50 employees needs possibly a different setup of BRMS capabilities than Organization B with >5000 employees. The employee ranges are adopted from previous surveys on the construction of situational artefacts (Bekkers et al., 2008; Winter, 2010; Zoet & Versendaal, 2013).

Q3:What is the scope of the BRMS implementation?

Application focused
Line of business focused
Organisation-wide

Q3 intends to retrieve the scope at which the BRMS implementation is conducted. Three main organisations scopes can be identified which are: Application focused, Line of Business focused and organization-wide. The work of Nelson et al., (2010) showed the scoping from narrow (single application focused) and expanded to line of business focused and eventually to organization-wide. This question intends to retrieve the data of what the scope was of the BRMS implementation conducted by the participant.

3.1.1.2 *Characterization of Business Rules Management*

The characterization of BRM section (Q4) intends to define the view of the organisation at which the BRMS is implemented on how and why they are using BRM. Therefore, the known goals of BRM (Boyer & Mili, 2011; Date, 2000; Graham, 2007; Jones, 2012; Morgan, 2002; Ross, 2003; Von Halle, 2002; Zoet, 2014) are presented, and a true or false answer should be given. Thereby, depicting the view of the organisation on why and how they are using BRM and as a result implementing a BRMS to achieve these goals.

Earlier questionnaires on situational artefact construction also propose a characterization section and therefore, for this questionnaire, this is also adopted (Winter, 2010, 2011b; Zoet & Versendaal, 2013).

Q4:The goals of BRM are (from the organisation's viewpoint):

Improving the productivity of elicitation
To improve the effectiveness of elicitation
The construction of a library of decisions
Giving insight into relationships between artefacts
Reducing effort in design (requirements and specifications)
Shortening the design phase
The improvement of the productivity in design
To improve the effectiveness of design
Mobilizing experts
Business rules mapping as much as possible
Quality assurance validation and verification
Automated verification of business rules
Reducing efforts at verification
Improve the productivity of verification
Improve the effectivity of verification
Generate automated test cases
The automated testing with validation
Give insight which artefacts are hit
Validated and accessible business rules
Reducing testing for implementation independent models
Reducing testing for implementation-dependent models
Reducing the effort for validation
Working with implementation independent business rules to export models
Simplify converting from models into code
Separating of implementation 'know' and 'flow.'

3.1.1.3 *Business Rules Management Solution Capabilities*

This section will contain the nine capabilities of which a BRMS can exist of which are addressed in the work of Smit and Zoet (2017), and Zoet and Versendaal (2013). The capabilities have each a specific set of questions which are unique (not for all) to each capability and thereby creating possible instantiations within the BRMS implementation.

Q5:Has this capability been part of the BRMS implementation?

Yes
No

The question Q5 focusses on the reason that not every BRMS implementation contains all the nine capabilities. Thereby, each capability section contains the question if it was part of the BRMS implementation. The following nine BRMS capabilities will be covered in the sub-sections below: 1) the elicitation capability, 2) the design capability, 3) the specification capability, 4) the verification capability, 5) the validation capability, 6) the deployment capability, 7) the execution capability, 8) the monitoring capability and, 9) the governance capability.

Elicitation

The elicitation capability exists of two functions. The function is to determine the knowledge which realises the value proposition of the business rules. This knowledge needs to be captured from various sources including but not limited to laws and regulations. The second function is the to initiate an impact analysis; this is only done when a business rule architecture is already in place (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017a).

Q6:From which source(s) are business rules elicitated?
People Documents Data

Q6 intends at extracting what source is used for the elicitation capability at a specific organisation. Examples here are: Subject-Experts (people), existing organisation regulations and guidelines (documents), existing database data (data), or a combination of the previously mentioned examples can be used as a source for this capability.

Q7:The end result of the elicitation capability is a set of selected relevant sources?
Yes No

The purpose of Q7 is to measure if this capability is actually used as it was intended to be used. The possibility is that only data is extracted and nothing is done with the sources that are used for extracting data. Thereby, for example, creating extra work if new rules should be created because of the change in laws or regulations.

Q8:Which type(s) of analysis have been applied?
Source analysis, elicitation based on sources Scenario analysis, elicitation based on actual business scenarios

Q8 measures which type of analysis is applied in the elicitation capability. The source analysis compares sources with each other, analyse where the source is from and if the source is reliable or not (Smit, Zoet, & Versendaal, 2017a).

Scenario analysis is the development and comparison of possible business scenarios. Scenario analysis is a process of analysing possible business events by considering possible alternative outcomes. Scenario analysis does not focus on one exact snapshot of the future. Instead, it presents several alternative future developments without taking into account any historical data (Aaker, 2008; Smit, Zoet, & Versendaal, 2017a).

Q9:In the case of a change in business rules, is there an impact analysis conducted? Yes No

Originally, the impact analysis should be performed in the design capability. Nonetheless, the BRM experts state that in practice this is also performed in the elicitation capability. This impact analysis is conducted when there already is a business rule architecture in place (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017a).

Design

The output of the design capability is the business rule architecture. The business rule architecture contains a combination of derivation structures and context designs (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017a).

Q10: Which of the following 5 V's are used in the design? Value represents the value of the decision to the organisation Velocity represents the speed at which the decision is to be taken Volume represents the number of times a decision is taken in a given time period Variety represents the variety of execution paths of a decision which is to be taken Veracity represents the accuracy at which the decision is to be taken

Q10 has the purpose to measure if the 5 V's are respected when implementing the design capability. The five V's (value, velocity, volume, variety and veracity) are indicators when dealing with Big Data. In this case, the five V's are adopted to describe the design of a BR architecture. The Big Data five V's are adopted from the work of Demchenko, Grosso, De Laat, and Membrey (2013), Gandomi and Haider (2015), and McAfee and Brynjolfsson (2012) and are adapted to the field of BRM by Zoet (2015).

Q11: Which of the following roles are defined (also known as a RAPID)? Recommend Agree Perform Input Decide

Good decision making depends on the assignment of specific and clear roles. The key is to be clear who has the input, who gets to decide, and who gets it done. Rogers and Blenko (2006) created the RAPID model to clarify the decision-making process. RAPID stands for Recommend, Agree, Perform, Input and Decide.

Recommend, people carrying this role are responsible for gathering input, and proving the correct data to ensure a sensible decision in a correct and timely order.

Agree, people in this role have the responsibility to state if the recommendation is good or not, respond with yes or no or, in other words, the so-called right to veto the recommendation.

Perform, someone or multiple people have the responsibility of executing the decision, once the decision is made. The possibility is present that the people recommending the decision are the same people executing the decision.

Input, the role of input is consulted on the decision. The people providing input are general the people involved in the implementation of the decision.

Decision, the person in the deciding role is the formal decision maker. This person is ultimately responsible for the decision when it goes right or wrong. The deciding role has the authority to resolve any impasse and enforce the organisation to action.

Q12: In the case of a change in business rules, is there an impact analysis conducted? Yes No

This is the same impact analysis as described earlier in the elicitation capability, but this is the capability where the impact analysis should originally be conducted. This impact analysis is conducted when there already is a business rule architecture in place.

Specification

The specification capability specifies the content of each separate context design. The function of this capability is to create the business rules and fact types needed to constrain or define particular aspects of the business (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017a).

Q13: How are the business rules specified? In models In text

Specifying business rules in models is based on the idea that humans should not use programming language to write code, but instead should create models from which code is generated from (Bass, Clements, & Kazman, 2012). In this case, the business rules (and the elements of the business rule) are specified with the use of models, an example of such a modelling language is that of Decision Model and Notation (DMN) (Object Management Group, 2014).

Specifying business rules in text is based on that business rules are specified with the use any form of language. Any form of language ranges from programming code to natural language. In this case, the business rules are specified in any form of text. Examples of this are the Dutch or English language, or with a language like Rulespeak (Ross, 2009).

Q14: Are the business rules specified in an unambiguous manner? In this case, unambiguous means that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine Yes No

Q14 intends to measure if the language used in the specification capability is implemented in the rules engine without any influence of a person. Thereby, ensuring that the language used in the capability only has one meaning. Therefore, being unambiguous.

Verification

The verification capability verifies the created business rule architecture to check for semantic and syntax errors (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017b).

Q15: Wat is the degree of automation? Detection degree according to the Governance, Risk and Compliance detection matrix. Automated - Detective Automated - Preventive Manual - Detective Manual – Preventive

Tarantino (2008) created a degree of automation matrix which shows in what degree the control system is automated. This degree automation matrix is adopted by Smit, Zoet, and Versendaal (2017b) for the BRM field and therefore used for this study and measures the degree of automation of the verification capability. The matrix consist of the following elements: automated – detective, automated – preventive, manual – detective, and manual preventive.

Manual - detective is the element where employees manually check for possible errors and report back upon the author of the business logic if any errors were found.

Automated - detective is the element that is defined as a system that checks the business logic after its creation and reports back in the form of a list of identified errors.

Manual - preventive is the element that employees are always authoring business logic together with the author and manually intervene when an error is made, enforcing the business logic author to correct the error.

Automated preventive is the element which is applied by the system, suggesting or enforcing certain behaviour regarding the authoring of business logic to prevent errors.

Validation

The validation capability reviews the created value proposition. The goal of this capability is to check for possible errors in its intended behaviour (Smit & Zoet, 2016; Smit, Zoet, & Versendaal, 2017b).

Q16: What kind of validation is applied? Peer review, a colleague checks if the artefacts are in concurrence with its sources Scenario validation, scenario-based validation applies predefined test sets to check the behaviour Source validation, Source-based validation focuses on validation based on the actual sources

Peer review is the validation of work by colleagues of similar expertise and competence to the authors of the work. In the case of peer review, a colleague (peer) checks if the artefacts are similar to its sources. When errors are identified that artefact is rejected and the capability cycle (elicitation, design, specification, and verification) starts from the beginning (Smit, Zoet, & Versendaal, 2017b).

Scenario validation is a validation method that uses hypothetical stories to support the tester through a test system or complex system. In the case of a BRMS, scenario validation makes use of all possible business scenarios. This is in most cases a manual task. Therefore, scenario validation is intensive in terms of resources it requires. When automated the resources needed for scenario validation are reduced (Smit, Zoet, & Versendaal, 2017b).

Source validation validates with the use of actual sources. Actual sources could be legal sources (laws and regulations). The downside of source validation is that the legal sources do not contain all possible scenarios. Thereby, the possibility exists that there is a loss in validation accuracy (Smit, Zoet, & Versendaal, 2017b).

The possibility exists that the earlier mentioned validation methods can be combined in one validation method, also known as a hybrid.

Q17: During validation, the following attributes are checked:
Traceability, provide an audit trail of access to the business rule and of any changes made to the business rule Completeness, which data (elements) need to be registered regarding the objects within the process Accuracy indicates the degree to which the stored data reflects the reality concerning an object Usability, the learnability and ease of use of the business rule Other…

Q17 intends to measure if the validation capability checks on traceability, completeness, accuracy and usability. The attributes are known as quality attributes and the ensuring certain quality of data or information (Rula, Maurino, & Batini, 2016).

Traceability is a quality attribute, which is the ability to provide an audit trail of access to the business rule and of any changes made to the business rule. Thereby, providing the ability to verify history, location, or the application of a business rule by means of documented identification.

Completeness is a quality attribute indicating which data (element) need to be registered regarding the objects within the process.

Accuracy is a quality attribute which indicates the degree to which the stored data reflects the reality concerning an object. Thereby, describing the closeness of a measurement to the true value,

Usability is a quality attribute which indicates the ease of use and learnability of the business rule.

These four quality attributes are selected because of the relevance in the BRM field (Zoet, 2014).

Deployment

The deployment capability transforms the verified and validated value proposition to implementation-dependent executable business rules. The actor that utilises the value proposition is not necessarily a system; a subject-matter expert could also utilise this. No specific questions were included for the deployment capability because this is mainly a project management element of the BRMS and not BRMS unique.

Execution

The execution capability processes the output of the deployment capability and is then executed, thereby delivering the actual value proposition. The realisation of the added value is conducted by executing the business rules by human or information system actors.

Q18: Is 'gaming' taken into consideration?

A system where gaming is possible means a system where users can generate the desired result, by means of testing.

Yes
No

Q18 intends to measure if the concept of gaming is taken into consideration. Gaming gives the user of a system the possibility to generate any desired result (Bevan & Hood, 2006; Morreim, 1991; Rieley, 2000). An example of this is: a user working with a BRMS in the governmental sector needs a tailor-made solution for a citizen, in this case, the result is more important than the way it is executed. Therefore, the user is ‘’gaming the system’’ to generate the desired result. Gaming has also a negative side because the possibility exists that the user of the system is doing this for all the wrong reasons.

Q19: What data is stored during execution?

Input data
Output data
Executed rules

Q19 aims to measure what specific data is stored during the execution. The specific data can be categorised in input data, output data, and executed rules. Input data is the data that is needed to execute the business rules. Output data is the data that is stored that comes out when the business rules are executed. Executed rules are the rules that are executed during the execution.

Monitoring

The monitoring capability observes, checks and keeps a record of not only the execution of the value proposition but also the full range of activities in the previously explained BRM capabilities that are conducted to realise the value proposition.

Q20: Which KPI's are being evaluated?

- The frequency of corrections per selected context design emerging from the verification process
- The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error
- The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule
- The frequency of corrections per selected context design emerging from the validation process per type of validation error
- The frequency of corrections per selected context architecture emerging from the design process per scope design
- The frequency of instantiations per selected context design
- The frequency per selected type of validation error
- The frequency per selected type of verification error
- The number of time units required to define, verify, and validate a single business rule
- The frequency of deviations between an implementation dependent context design and an implementation independent context design
- The frequency of executions of an implementation dependent business rule
- The frequency of execution variants of a scope design
- The number of time units required for the execution per execution variant

- The amount of business rules that cannot be automated

Answers:
Not evaluated
Constantly
Daily
Weekly
Monthly
Quarterly
Yearly

Q20 intends to measure what is being evaluated in the monitoring capability. The Business Rules Management Key Performance Indicators are used from the work of Smit and Zoet (2017b), and Zoet, Smit, and de Haan (2016) to measure what is being evaluated in the monitoring capability. A remark on this research is that the sample group is solely drawn from Dutch government institutions. The authors of the earlier mentioned work state that the government institutions are representative towards organisations implementing BRMS. Furthermore, the frequency of the evaluation of the KPI's are measured with Q20. This could be, for example, a combination of daily and monthly evaluation. The possibility exists that an instantiation of BRMS setup in a specific sector could have a different setting compared to a BRMS with a different frequency is of evaluation.

Q21: Do you measure any additional KPI's? (open question) (IST)

Q22: Do you want to measure any additional KPI's? (open question) (SOLL)

Q21 and Q22 aim to gather additional KPI's in the IST and SOLL situation mentioned by the participant. The existing set of KPI's are limited because of the small sample size and the industry where it was focused on (public) (Smit & Zoet, 2016). Therefore, Q21 and Q22 exists to gather any additional KPI's used by other organisations and other industries.

Governance
The governance capability contains three sub-capabilities. The sub-capabilities are version management, validity management, and traceability (Morgan, 2002; Smit & Zoet, 2016). Q23 intends to measure which sub-capabilities of the governance capability are implemented during the BRMS implementation.

Q23: Governance consists of the following capabilities:

Version management
Validity management
Traceability

Version management

The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities. Correct version management allows organisations to track what elements are used in the execution and the deliverance of the added value of the BRM processes.

Validity management

The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time. Transparency is increased by utilising Validity management. Thereby, enabling organisations to provide the possibility to show when specific value propositions are, were or will be valid.

Traceability

The purpose of the traceability capability is to ensure the possibility to trace created elements to their corresponding laws and regulations. Furthermore, the traceability capability creates a foundation for impact analysis when, for example, new laws are needed to be processed into value propositions.

Alternatively, a combination of the options mentioned above.

3.1.1.4 IST & SOLL

To reduce the effects of low budget and lack of knowledge, which could occur in the current situation of implementing a BRMS (IST), the ideal situation (SOLL) is asked from the participants. Relying on the expertise of the participants and thereby exposing any difference that could occur when there is a low budget of a lack of knowledge. An example is: the validation capability could, in theory, be fully automated but because of the low budget and lack of knowledge this could not be realised. By introducing IST & SOLL, the difference could be shown and taken into consideration when creating the problem class framework.

Identical questions are used for the IST & SOLL situations. Thereby, not to confuse the participants. Previous situational artefact construction questionnaires (Winter, 2010, 2011b) used these concepts, and therefore this is adopted and adapted for the BRM research field.

3.1.1.5 Leader of the capability

The business rules task/service model from Nelson et al., (2010) identifies three areas within a firm relevant when dealing with the responsibility of working with a BRMS: IT, Business and a Central IT/Business group. Q24 focusses on which area has the responsibility of a specific capability. The model provides high-level services, and functions focused on a BRMS as a whole. Focussing more on the capabilities of a BRMS, different responsibilities of capabilities lie with different areas. Often the more technical capabilities of a BRMS lies in the area of IT and the capabilities where managerial decisions are needed in the Business area.

Q24: Who is in the lead of this capability?
IT Business Central IT/Business group

3.1.1.6 Autonomy

Coming into the era of computer automatization, the possibilities are growing where computers take over some, or whole, technical tasks from humans (Parasuraman, Sheridan, & Wickens, 2000). The same is possible with some of the capabilities of a BRMS. Therefore, the question is asked what the level of autonomy of the capability is that is implemented. Measuring the degree of autonomy is done with ten degrees of autonomy, the so-called Sheridan model (Parasuraman et al., 2000).

Q25: What is the autonomy of this capability?

The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman et al., 2000).

1. The computer does not help; people must do it all
2. The computer provides a complete set of action alternatives
3. The computer narrows down the choice to a few action alternatives
4. The computer suggests one action alternative
5. The computer runs one action suggestion if the human approves
6. Man can veto before it runs automatically in a limited time
7. The computer runs automatically and informs people where necessary
8. The computer informs the human after execution only when it is requested
9. The computer informs the human after execution if the computer which decides
10. The computer decides everything independently and ignores humans

Q25 intends at retrieving on what degree of autonomy the capability runs in the organisation where it is implemented. The degree of autonomy ranges from level 1, the computer does not help, and people must do everything, to level 10, the computer takes decision independently without any intervention from humans. A major threshold of implementing a high level of autonomy is having sufficient funds and knowledge, which could be lacking at some organisations where a BRMS is implemented.

3.1.1.7 BRMS observation technique validation

Taken into account that the initial situational artefact construction technique of Winter (2011b) do not state anything about validating the survey or validation overall. When performing design science validation plays a major role. We believe that therefore validation cannot be left out of this process. Validation is needed to ensure the correctness of the created BRMS observation technique. A selection is made from experts from the BRM community, and a pilot test is conducted together with the selected experts. The group existed of a professor conducting research which focusses on utilising BRM, a PhD student conducting research in the BRM domain, and a master student with research and practical experience in the BRM domain. All the expert interviews were held in the same setting with all of them in a time span of 60 minutes. The goal of the expert interviews is to have a correct survey which includes all the questions needed to gather BRMS implementation cases. The survey itself was used as an interview protocol, being that all the elements were used as triggers to elicit examples from practice by the experts. Thereby, validating the correctness of the elements of the BRMS observation technique.

The experts gave examples of what should be included in a correct survey on implementing BRMS. This resulted in comparable structure and content as compared to the BRMS observation technique created out of literature. Elements which were adopted from earlier conducted situational artefact surveys were not mentioned during the expert interviews. Nonetheless, these elements were still included in the BRMS observation technique for the sole reason that previous work, conducted in this field, has proven successfully (Bucher & Winter, 2010; Winter, 2010, 2011b). These elements were the following: the IST & SOLL (3.1.1.4), capability autonomy (0), and the characterisation of BRM (3.1.1.2).

3.1.2 Implementing the BRMS observation technique

In the previous section, a questionnaire is created which aims to measure which different situational factors occur in different situations where a BRMS is implemented. This section focusses on how the survey is set out. Survey research is conducted to ensure a high level of validity of the BRMS implementation framework. The sample survey will contribute to the generalizability of the BRMS implementation framework to practice.

The questionnaire is distributed to people with experience implementing BRMS. The participants are given two options 1) to fill in the questionnaire by themselves or 2) with the support of the researcher.

The first choice will only be considered part of the sample survey where only the questionnaire is filled in without any additional information asked. The second choice is somewhat the same as option one with the addition that the participant will be asked some additional questions. The participant will be asked for any additional information on the questions asked and answered. Thereby, further validating the existing business rules management situational factors and business rules management problem classes.

The non-probability sampling method – self-selection – is chosen to gather sufficient data for the survey. A questionnaire is distributed over LinkedIn groups with the topics on business rules or business rules management. The selected groups are shown in Table 3 together with their number of members. The total members of the group are not unique members being that there is the possibility to be a member of multiple groups. Furthermore, the explanatory text posted in the LinkedIn groups can found in Appendix A.

Table 3 LinkedIn group details

Name:	**Topic:**	**# Members:**
Decision Model and Notation (DMN)	Decision Management	937
The Decision Model	Decision Management	1514
Decision Management Strategies Group	Decision Management	302
BRMS (Business Rules Managment system) group	BRMS	965
Business Decision Management	Decision Management	102
Business Rules Platform Nederland	Business Rules	606
Business Rules (BRMS) Professional Services Consulting	BRMS	1233
Business Rules Engines Interest Group	Business Rules	483
Business Rules	Business Rules	2297
Business Rules Platform	Business Rules	570
	Total:	8526

This survey intends to reach the members who have experience with a BRMS implementation. The groups consists of members distributed over a wide range of sectors, mostly government, healthcare and finance. Furthermore, the possibility exists that a unique group member has experience with multiple BRMS implementation. Therefore, these members can fill in the questionnaire multiple times.

Besides the self-selection sampling method using the LinkedIn groups, e-mails are sent to people with experience implementing BRMS. Thereby, the possibility exists that there will be people contacted who are also in the LinkedIn groups. The e-mail survey participants are drawn from a contact list which is created over several years of BRM projects in practice and in the scientific field. This contact list contains 600 e-mail addresses. The explanatory text used in the e-mail survey can be found in Appendix #. The people contacted by e-mail are participants with a higher level of experience when it comes to the subject of BRM and the implementation of a BRMS. The e-mail technique is used to make the approach more personal and therefore possibly increase the return rate.

One of the pitfalls of survey research is the expectation of a high response rate. Newsted, Huff, and Munro, (1998) state that one of the disadvantages of surveys is the low response rate which impacts the generalisability of the results. This, sadly, also occurred during this research. Therefore, the decision is made to shift from the online self-selection sampling method to a more interview based approach. The questionnaire constructed earlier is still used for the same purpose. The interviews have two focus areas 1) the completion of the questionnaire itself and 2) additional validation of the situational factors and problem classes. Conducting the survey in an interview based approach supports the validation of the elements of the BRMS observation technique. Therefore, countering another one of the surveys disadvantages which is the lack of rich or 'thick' description (Newsted et al., 1998). The BRMS observation technique elements are 1) the BRM situational factors and 2) the BRMS problem classes. The situational factors are represented as the survey questions; each answer gives a different instantiation of the implemented BRMS. The BRMS problem classes are constructed from literature (Smit & Zoet, 2016; Zoet & Versendaal, 2013) and from input from the BRM community. These elements are the earlier depicted nine BRMS capabilities, which are: 1) the elicitation capability, 2) the design capability, 3) the specification capability, 4) the verification capability, 5) the validation capability, 6) the deployment capability, 7) the execution capability, 8) the monitoring capability and, 9) the governance capability. Besides the additional validation, when filling in the questionnaire in an interview approach, is the supervision if there is a different interpretation of a question, concept or definition thereby influencing the validity of the given answers.

The data is gathered among different organisations distributed over the financial (*n*=6) and public sector (*n*=7), as shown in Table 4 down below. Employee ranges included 251 – 500 (*n*=2), 501 – 1000 (*n*=2), 2001 – 5000 (*n*=6), and >5000 (*n*=3). The implementation focus added an additional characterisation on the BRMS implementation cases. The implementation focusses are divided into: Application focus (*n*=3), Line of business focused (*n*=4), and Organization wide (*n*=6). The gathered data is going to be analysed in the next chapter.

Table 4 Gathered sample

Case ID:	Employees:	Sector:	Implementation focus:
1	2001 - 5000	Public	Organisation wide
2	2001 - 5000	Public	Organisation wide
3	>5000	Public	Application focused
4	2001 - 5000	Public	Organisation wide
5	2001 - 5000	Financial	Application focused
6	2001 - 5000	Financial	Line of business focused
7	501 - 1000	Financial	Line of business focused
8	251 - 500	Financial	Line of business focused
9	>5000	Financial	Organisation wide
10	251 - 500	Public	Application focused
11	>5000	Public	Line of business focused
12	501 - 1000	Financial	Organisation wide
13	2001 - 5000	Public	Organisation wide

BRMS observation technique conclusion

Following the situational artefact steps from Winter (2011b) the BRMS observation technique is created to gather BRMS implementation cases. The BRMS observation technique focusses on knowledge of implemented BRMS cases. The survey is separated into sections focussing on: the reason why to implement a BRMS, specific questions for every BRMS capability, and for each capability an IST and SOLL situation to discover the difference between what is actually implemented and how the people involved in the implementation actually wanted the implementation to have gone.

The first attempt at gathering the BRMS implementation cases turned out as non-successful due to the fact that the response was only one case. Therefore, the switch was made from an online survey to structured interviews. This had no impact on the setup of the questionnaire, and no additional information was provided during these interviews. Therefore, the one case gathered during the online survey was reused in the interview dataset. The response from the respondents of the more personal approach was positive. 13 BRMS implementation cases where gathered distributed over the financial and public sector and with different implementation focus (application, line of business, and organisation wide). In the next subsection, the BRMS implementation cases gathered are analysed in the BRMS construction process.

3.2 BRMS construction process

This subsection aims at analysing the data from the BRMS observation technique resulting in identification of the metaclasses of the BRMS metamodel. The situational artefact construction method from Winter (2011b) is used as a guideline to create a situational artefact in the BRM domain. The eleven steps proposed by Winter (2011b) were as followed:

1. Initial demarcation of the design problem class
2. Identification of potential contingency factors
3. Field study based analysis of design problems in practice
4. Refining specifications of the design problem class
5. Calculation of ultra-metric distances
6. Determination of a useful level of generality
7. Specification of design situations
8. Identifying characterising design factors
9. Linking design factors to design problems
10. Deriving elementary problem-solving actions
11. Deriving method configuration rules

The BRMS construction process from Winter (2011b) is used as a guideline for this research. The deliverables of the BRMS construction process are shown in Figure 14, which shows the initial BRMS construction process from Winter (2011b) together with the deliverables of the process used for this study. The deliverables are described further in this chapter.

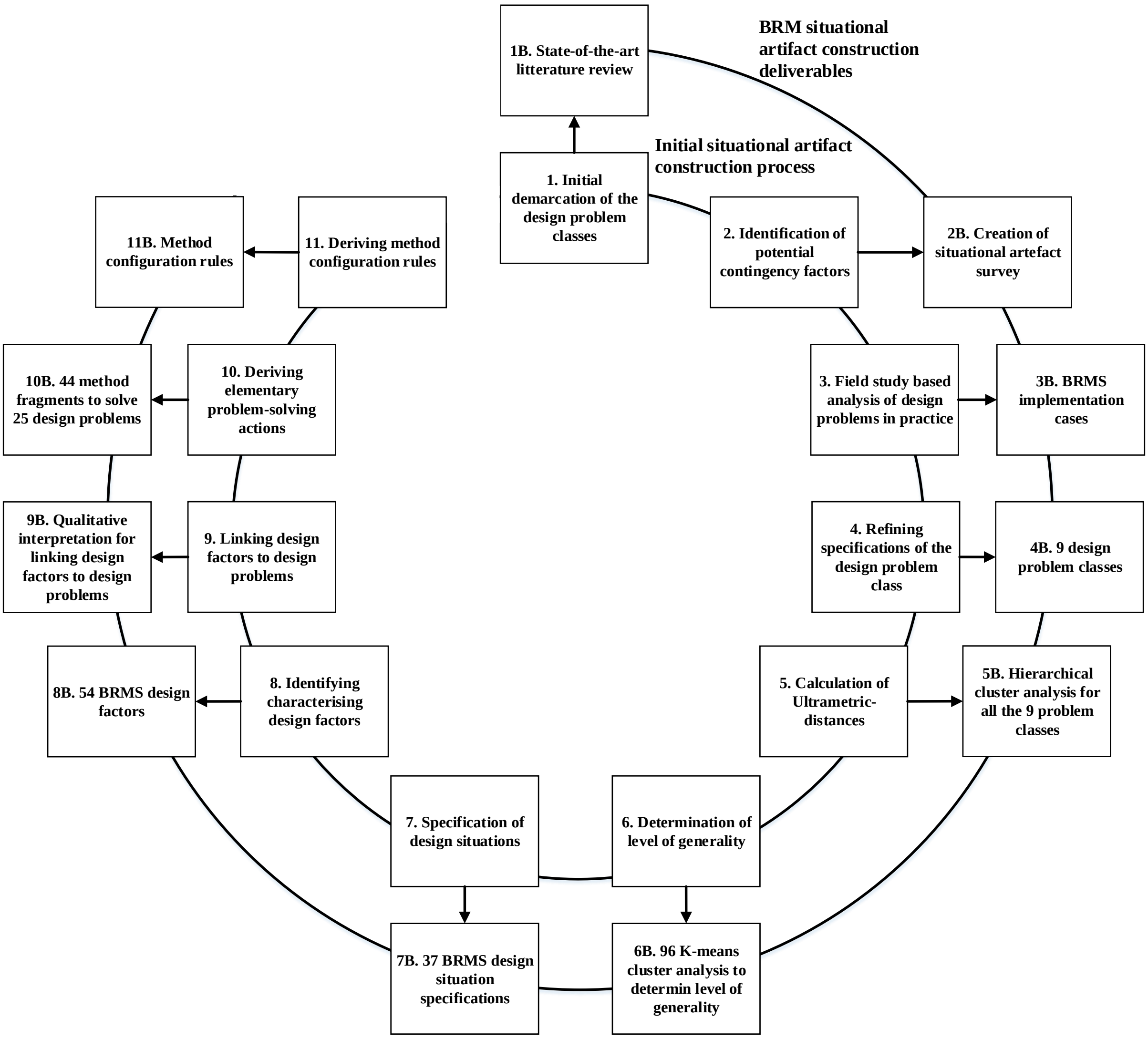


Figure 14 BRMS construction process

The state-of-the-art literature review covered step one and two (Initial demarcation of the design problem class and identification of potential contingency factors) and are therefore finalised. Step three (Field study based analysis of design problems in practice) is also finalised but does need some additional information before continuing. The field study is covered in the data collection chapter with the use of the BRMS observation technique. Step three also asks for that a principal component analysis (PCA) should be conducted (Winter, 2011a), this is discussed in the section below. Steps four to eleven are discussed further on in this chapter and will result in a specific configuration of a solution of how to implement a BRMS in a certain situation.

3.2.1 Principal component analysis

The PCA is a statistical analysis that transforms a set of observations of variables with a possible correlation into a set of linear uncorrelated principal components. The outcome of this analysis is always a set which is less or equal in numbers compared to that of the starting set (Abdi & Williams, 2010; Jolliffe, 2002; Pearson, 1901). Utilising the PCA in this research, the PCA is focused on an outcome of relevant design factors from a list of potential contingency factors (Winter, 2011a).

When conducting a factor analysis, the first key factor is the sample size. This could be looked at from two different perspectives 1) the sample size as a requirement 2) or the subject-to-variables (STV) ratio as a requirement. Literature on the required sample size when conducting a factor analysis differentiates from a minimum of 100 respondents to a minimum of 500, as shown in Table 5.

Table 5 Factor analysis sample requirement

Source:	Minimum sample requirement:
MacCallum, Widaman, Zhang, and Hong (1999)	100 (fair) – 500 (excellent)
Arrindell and van der Ende (1985)	300
Comrey and Lee (2013)	500

The perspective of STV ratio focusses on the ratio of the number of elements in the questionnaire to the number of respondents. This ratio differentiates from 20:1 to 3:1 respondents per questionnaire element, as shown in Table 6.

Table 6 Factor analysis STV ratio requirement

Source:	Minimum ratio requirement:
Hogarty, Hines, Kromrey, Ferron, and Mumford (2005)	20:1
Arrindell and van der Ende (1985)	3:1 – 10:1
MacCallum et al. (1999)	5:1

Taking into account both sample size perspectives, a number of participants who filled in the questionnaire (NUMBER #) and the STV ratio (for a questionnaire with ten elements a minimum of 30 respondents are needed) is not sufficient to conduct a PCA.

Therefore, the main purpose of the PCA (list of potential contingency factors to a relevant set of design factors) is translated to a more qualitative approach. The validation pilot test is used as a qualitative method for reducing the list of potential contingency factor candidates to a smaller set of relevant "design factors". Thereby, the existing questionnaire is already a relevant list of design factors. The next step will cluster the solutions into relevant clusters which are representative towards all the gathered cases.

3.2.2 Specification refinement for the design problem class

The earlier qualitative approach focusses on creating a relevant list of design factors. This step focusses on specifying and refining the design factors into more specific and refined design problem classes. The design problem classes identified in earlier steps should be refined more to ensure a useful degree of homogeneity of the solutions. This will result in the excluding of ''outlier'' design solutions and thereby ensuring a useful degree of homogeneity. Due to the fact that the questionnaire had a low number of respondents, excluding of ''outlier'' design solution (the respondent's data) is not possible, because no outliers can be identified. Therefore, all design solutions are used for further analysis. The problem classes identified and specified are:

- Elicitation problem class
- Design problem class
- Specification problem class
- Verification problem class
- Validation problem class
- Deployment problem class
- Execution problem class
- Monitoring problem class

- Governance problem class

These problem classes are identical to the nine capabilities of a BRMS due to the fact that a capability is something what a solution is capable of and a problem class is a classification of a problem that the solution tries to solve (Smit & Zoet, 2016; Zoet & Versendaal, 2013).

3.2.3 Ultrametric-distance calculation

Design problem classes can be divided into a number of generic design situations depending on the degree if generality. Based on the (squared-) Euclidian distance, the similarity (or dissimilarity) of two design solutions within a problem class can be represented by an ultrametric-distance. Wards method is applied and thereby ensuring the creation of clusters of similar size and clearly defined clusters. The aim of Ward's method is to cluster groups in a way that variation inside these groups does not increase drastically. This results in clusters with a high level of homogeneity (Wolfson, Madjd-Sadjada, & James, 2004).

The cases and their distances are visualised using a dendrogram graph. Ultrametric distances can be visualised by a graph whose y-axis is generality and whose x-axis is the set of observed cases. The similarity or dissimilarity of two (or more) design solutions corresponds to the generality level of their relation. If the similarity is high, their relation is represented on a lower level of generality. Therefore, when there is a low level of similarity, their relation is represented on a higher level of generality (Winter, 2011a). Figure 15 shows the ultra-metric distance of the elicitation problem class. This dendrogram graph contains 13 solutions (elicitation 1...13) for the elicitation problem class. Observed case 2 is represented by its own specific solution, Elicitation solution 2. This level of generality is the same for the other observed cases on this level which are represented by their own specific solution. Observed case 1, 2, 5, 6, 11 and, 13 are represented by a more generic solution, Elicitation solution 1, 2, 5, 6, 11 and, 13. The observed cases 1 - 6, 9, and 11 - 13 are represented by an even more generic solution, Elicitation solution 1 - 6, 9, and 11 - 13. At the top level, the generic solution contains all the observed cases and is the most generic representation towards all the observed cases.

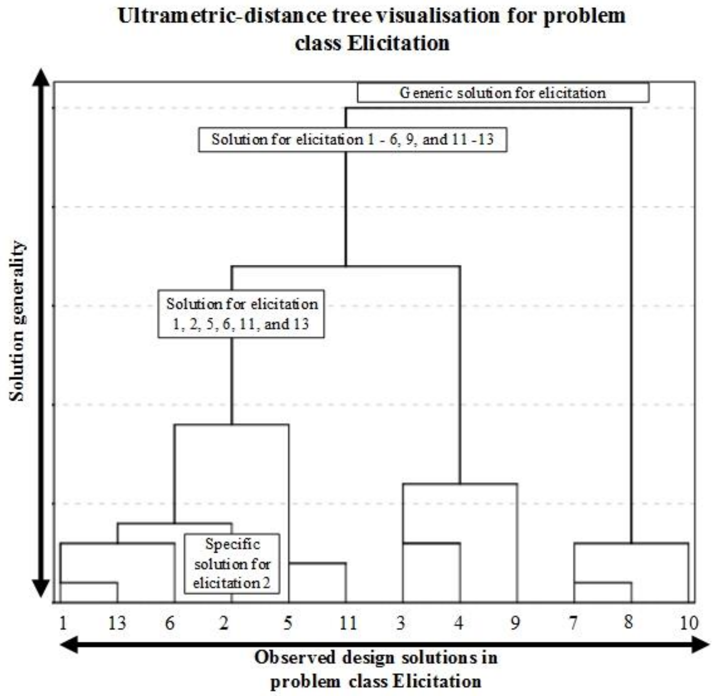


Figure 15 Ultrametric-distance tree visualisation for problem class elicitation

Because this research focusses on a certain configuration of a BRMS (of the nine capabilities described earlier), the ultra-metric distances are calculated for each different capability. The dendrogram graphs of the other eight capabilities can be found in Appendix C – Ultrametric-distance calculations. Determination of how many clusters exists in a design solution and which clusters represents certain BRMS implementation cases is needed to specify the level of generality.

3.2.4 Generality level determination

In order to not only visualise the generic solutions, but also characterise the generic solutions, k-means cluster analyses are applied to the implementation cases. This is to determine the optimal number of clusters for the design solution; several k-mean cluster analyses were conducted with a minimum of 1 to a maximum of 13 clusters. These cluster analyses were conducted for each problem class because the possibility exists that a solution contains only one capability (problem class). Therefore, a total of 96 k-means cluster analyses were conducted to determine the optimal number of clusters for each problem class. For each solution the error total which is calculated from the distances of all implementation cases to the centre of their clusters. Bases on this total, the so-called elbow criterion is used (Everitt, Landau, Leese, & Stahl, 2011; Hardle & Simar, 2007; Winter, 2011b). The elbow criterion indicates which increased number of clusters leads to an above-average improvement in the error sum. The error sum is plotted on the y-axis and the number of clusters is plotted on the y-axis, elbows arise for the 4-cluster, 7-cluster and 10-cluster solution as shown in Figure 16. The elicitation problem class is taken as an example in the analysis and the other eight problem classes elaborated further in Appendix D – Generality level determinations. The elbow criteria of all the nine problem classes are shown in Table 7.

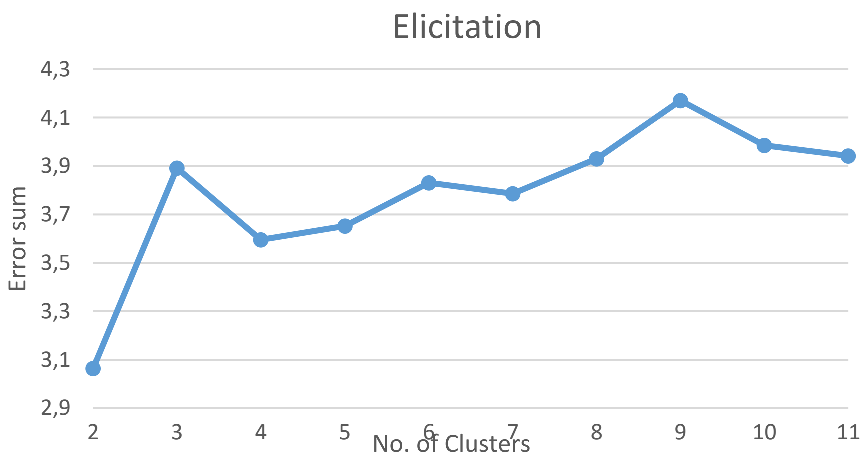


Figure 16 K-means cluster analysis for generality determination (elicitation)

Due to the low number of respondents the elbow criterion is readily affected by change. One added case can shift the whole consistency of a number of clusters in a design solution. Nonetheless, the choice is made for the lowest number of clusters in a solution which is a 4-cluster solution. The 4-cluster solution optimally represents the implementation cases on the different industry sector, employee ranges, and implementation scopes. The choice for a larger cluster solution would be too close to a more specific solution which has economic consequences. A more specific solution would mean a higher total of solution and therefore higher costs of creating the solution, and these more specific solutions are also affected by possible future changes occurring in the cluster content (Winter, 2011a).

Table 7 Elbow criteria

Problem class:	Elbow
Elicitation	**4**
Design	5
Specification	5
Verification	3
Validation	3
Deployment	5
Execution	5
Monitoring	4
Governance	3

Translating the elbow criterion of the elicitation problem class into the ultrametric-distance graph (shown in Figure 17) and thereby representing a generic solution which is a 4-cluster solution.

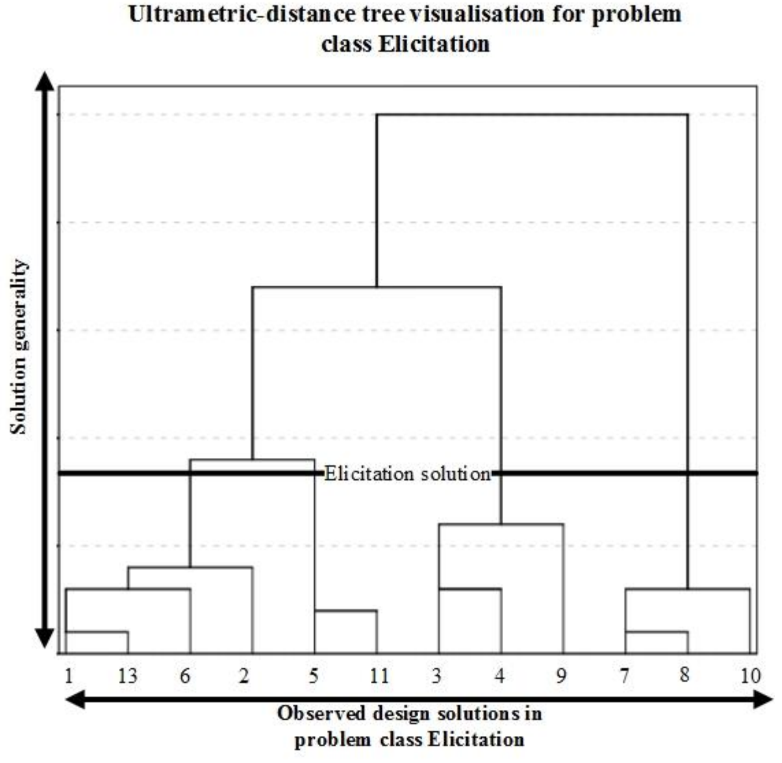


Figure 17 Dendrogram generic elicitation solution

Further specification of the clusters is needed, this will be done in the next section.

3.2.5 Design situation specification

The design situations need more than only formal definition (which is done by the ultrametric-distance calculation), but also need semantic interpretation (i.e. by specifying the design problem types). Due to the low amount of observations, using descriptive statistics from the conducted cluster analysis and thereby using these statistics for specifying the design situations is of an irrelevant manner. Current and any future additional respondents will have an impact on the cluster composition and therefore the design situation. The possibility exists that even the current set of clusters is not representative towards future BRMS implementations. Nonetheless, the ultrametric-distances are used, for semantically specifying the design situations. The Elicitation problem class is specified into a 4-cluster solution, and these four different clusters are specified by means of a mean comparison analysis (ANOVA). This analysis aims at specifying the four design situations. The goal of this is to create four different design situations. Thereby, aiming at what the exact reason why these design solutions are created into a cluster. Elements of all the four design situations which were identical, were excluded from the design situation specification (Table 8 shows an example of such a situation). Specifying design situations aims at showing factors which differentiate a design situation from another design situation, this will not help when identical elements do not differentiate design situation from another design situation.

Table 8 Design situation element exclusion

Question:	Cluster	N	Mean
What sources are used for elicitation of business rules ? (experts)	1	4	1,00
	2	3	1,00
	3	2	1,00
	4	3	1,00
	Total	12	1,00

The Elicitation problem class is used as an example in this section, the other design situations from the other eight problem classes can be found in Appendix E – Design situations. The four design situations of the Elicitation problem class are as followed:

Design Situation 1:

- Public sector focused
- Organisations with 2001 – 5000 employees
- Organisation implementation focused
- Fully manual elicitation focused (autonomy level 1)
- Database data is used for elicitating business rules
- Outcome of the capability is not a relevant set of selected sources
- Scenario analysis is conducted
-

Design Situation 2:

- Public sector focused
- Organisations with more than 5000 employees
- Line of business implementation focused
- Capability is supported with a complete set of action alternatives (autonomy level 2)
- No database data is used for elicitating business rules
- Outcome of the capability is a relevant set of selected sources
- No scenario analysis is conducted

Design Situation 3:

- Financial sector focused
- Organisations with more than 5000 employees
- Line of business implementation focused
- Capability is supported with a narrowed down set of action alternatives (autonomy level 3)
- No database data is used for elicitating business rules
- Outcome of the capability is not a relevant set of selected sources
- No scenario analysis is conducted

Design Situation 4:

- Financial sector focused
- Organisations with 251 -500 employees
- Line of business implementation focused
- Fully manual elicitation focused (autonomy level 1)
- No database data is used for elicitating business rules
- Outcome of the capability is a relevant set of selected sources
- Scenario analysis is conducted

The Elicitation design situations are visualised in Figure 18. The axis shows the design factors which make the design situations different from the other design situations.

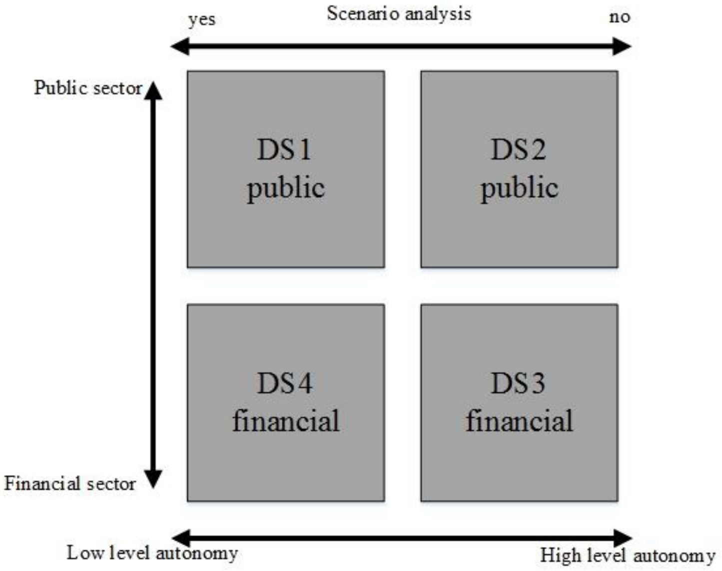


Figure 18 Elicitation design situation matrix

More work is needed to specify the characterising factors defining the design situations. This will be done in the next section.

3.2.6 Identifying characterising design factors

Each design situation consists of characterising design factors; these need to be specified further. Every design factor has different values, these values influence the characterisation of the design factor and thereby the design situation. An example of this is Design factor 5 (capability autonomy), this design factor characterises the level of autonomy the capability in the BRMS operates. The value could range from:

1. The computer does not help; people must do it all
2. The computer provides a complete set of action alternatives
3. The computer narrows down the choice to a few action alternatives
4. The computer suggests one action alternative
5. The computer runs one action suggestion if the human approves
6. Man can veto before it runs automatically in a limited time
7. The computer runs automatically and informs people where necessary
8. The computer informs the human after execution only when it is requested
9. The computer informs the human after execution if the computer which decides
10. The computer decides everything independently and ignores humans

3.2.6.1 *Overall characterising design factors*

The design situations exist of design factors which appear in every design situation. These design factors are the overall characterising design factors. Table 9 shows an overview of all the overall design factors, the design factor name, the possible value's and in which problem classes these design factors characterise design situations.

Table 9 Overall design factors

			Problem classes:								
ID:	**Design factor:**	**Value:**	**Elicitation**	**Design**	**Specification**	**Verification**	**Validation**	**Deployment**	**Execution**	**Monitoring**	**Governance**
1	Sector implementation	Public or Financial	X	X	X	X	X	X	X	X	X
2	Employee range	< 50, 50 - 100, 101 - 250, 251 - 500, 501 - 1000, 1001 - 2000, 2001 - 5000, or >5000	X	X	X	X	X	X	X	X	X
3	Scope focus	Application focused, Line of business focused, or Organisation wide focused	X	X	X	X	X	X	X	X	X
4	Capability leader	IT, Business, or Central IT/Business group		X	X			X	X	X	X
5	Capability autonomy	Autonomy level 1 - 10	X	X	X	X	X	X	X	X	X

3.2.6.2 *Design factors problem class specific*

The problem classes have for each of them different design factors. Every problem class (except Deployment) have different design factors which characterise the design situations of that particular problem class. Table 10 shows an overview of all the identified design factors with its related problem class, the design factor name, the possible value's, and a description of each design factor. Deployment does not have additional design factor because no design factors could be identified out of literature and expert interviews with BRM community members.

Table 10 Problem class specific design factors

ID:	**Problem class:**	**Design factor:**	**Value:**	**Description:**
6	Elicitation	Elicitation source (Data)	Yes/No	Database data is used as a source for elicitation.
7	Elicitation	Relevant set of selected sources	Yes/No	The output of elicitation is a relevant set of selected sources.
8	Elicitation	Scenario analysis	Yes/No	Scenario analysis is used during elicitation. Elicitation based on business scenarios.
9	Design	Velocity design	Yes/No	Velocity is one of the 5v's and represents the speed at which the decision is to be taken.
10	Design	Volume design	Yes/No	Volume is one of the 5v's and represents the number of times a decision is taken in a given time period.

11	Design	Veracity design	Yes/No	Veracity is one of the 5v's and represents the accuracy at which the decision is to be taken.
12	Design	Role: Agree	Yes/No	The Agree (RAPID) role is defined in the design problem class.
13	Design	Role: Perform	Yes/No	The Perform (RAPID) role is defined in the design problem class.
14	Design	Role: Input	Yes/No	The Input (RAPID) role is defined in the design problem class.
15	Design	Role: Recommend	Yes/No	The Recommend (RAPID) role is defined in the design problem class.
16	Design	Impact analysis	Yes/No	An impact analysis checks the impact that a certain change brings forth
17	Specification	Rule specification	In models or in text	Rules are specified in either models or in text.
18	Specification	Unambiguous specification	Yes/No	Unambiguous specification is that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine.
19	Verification	Detection degree	Automated - Detective Automated - Preventive Manual - Detective Manual – Preventive	Detection degree indicates at which degree syntax and semantic errors are detected.
20	Validation	Peer review	Yes/No	Peer reviews are conducted to validate the expected behaviour of the business rules.
21	Validation	Scenario validation	Yes/No	Scenario validation is conducted to validate the expected behaviour of the business rules.
22	Validation	Source validation	Yes/No	Source validation is conducted to validate the expected behaviour of the business rules.
23	Validation	Quality attribute: Traceability	Yes/No	The validation capability checks on the quality attribute traceability.
24	Validation	Quality attribute: Accuracy	Yes/No	The validation capability checks on the quality attribute accuracy.
25	Execution	Gaming	Yes/No	Gaming is made possible for the purpose of testing.
26	Execution	Rules saved	Yes/No	The executed rules are saved in the execution capability.
27	Monitoring	Verification correction frequency	Constantly Daily Weekly Monthly Quarterly Yearly Not evaluated	The frequency of corrections per selected context design emerging from the verification process.
28	Monitoring	Specific Verification correction frequency		The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error.
29	Monitoring	Validation correction frequency		The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule.

30	Monitoring	Specific validation correction frequency		The frequency of corrections per selected context design emerging from the validation process per type of validation error.
31	Monitoring	Design correction frequency		The frequency of corrections per selected context architecture emerging from the design process per scope design.
32	Monitoring	Context design frequency		The frequency of instantiations per selected context design.
33	Monitoring	Validation error frequency		The frequency per selected type of a validation error.
34	Monitoring	Verification error frequency		The frequency per selected type of a verification error.
35	Monitoring	Time worked on business rule		The number of time units required to define, verify, and validate a single business rule.
36	Monitoring	Deviation frequency		The frequency of deviations between an implementation dependent context design and an implementation independent context design.
37	Monitoring	Dependent execution frequency		The frequency of executions of an implementation dependent business rule.
38	Monitoring	Scope design variant frequency		The frequency of execution variants of a scope design.
39	Monitoring	Time required execution		The number of time units required for the execution per execution variant.
40	Monitoring	Non-automated business rules		The amount of business rules that cannot be automated.
41	Governance	Traceability	Yes/No	Traceability ensures the possibility to trace created elements to their corresponding laws and regulations. Furthermore, the traceability capability creates a foundation for impact analysis when, for example, new laws are needed to be processed into value propositions.

The characterising design factors are defined together with their values and related problem classes. The characterising design factors define the design situations, but there are also design factors which do not characterise design situations but are still important for solving design problems. The non-characterising design factors are not specifically solving design problems but are directly related to the problem classes, which in their turn solve certain design problems. The reason that these design factors are non-characterising is that for all the design situations in a problem class these design factors have the same value. The non-characterising design factors are shown down below in Table 11.

Table 11 Non-characterising design factors

ID:	Problem class:	Design factor:	Value:	Description:
42	Elicitation	Elicitation source (People)	Yes	People are used as a source for elicitation.
43	Elicitation	Elicitation source (Documents)	Yes	Documents are used as a source for elicitation.
44	Elicitation	Source analysis	Yes	Source analysis is used during elicitation. Analysis based on sources.
45	Elicitation	Impact analysis	Yes	An impact analysis checks the impact that a certain change brings forth.
46	Design	Value design	Yes	Value is one of the five V's and represents the value of the decision to the organisation.
47	Design	Variety design	Yes	Variety is one of the five V's and represents the variety of execution paths of a decision which is to be taken.
48	Design	Role: Decide	Yes	The Decide (RAPID) role is defined in the design problem class.
49	Validation	Completeness	Yes	The validation capability checks on the quality attribute completeness.
50	Validation	Usability	No	The validation capability checks on the quality attribute usability.
51	Execution	Input saved	Yes	The input is saved in the execution capability.
52	Execution	Output saved	Yes	The output is saved in the execution capability.
53	Governance	Version management	Yes	The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities.
54	Governance	Validity management	Yes	The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time.

The design factors are identified and characterised for each problem class and thereby for each design situation in the problem class. The design factors need to be linked to the identified design problems. The linking of design factors to design problems will be done in the next section.

3.2.7 Linking design factors to design problems

The characterised design factors described in the earlier sections need to be linked to the 25 proposed design problems which are known as reasons to implement a BRMS. All the earlier conducted steps analyse the existing design solutions. These design solutions are cases of successful BRMS implementations. These design solutions were created with a certain purpose, and therefore the design factors can be qualitatively interpreted and linked to the known design problems, which is shown in Table 12. Table 12, contains the known 25 design problems which are mapped against the identified and characterising design factors. The design factors are marked, with an ''X'', against the design problems which they have a positive impact on. An example is design problem ''low productivity of elicitation'' this could be positively impacted by letting the system take over some tasks (capability autonomy), using data as a source when elicitating, the output of elicitation is a relevant set of selected sources (which could be reused), performing a scenario analysis which is based on business scenario's, and defining roles (the input role). In short, a certain combination of characterised design factors could solve a certain design problem.

Table 12 Design factors linked to design problems

		Design factor (ID):																																		
ID:	Design problem:	5	6	7	8	9	10	11	12	13	14	15	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
1	Low productivity of elicitation	X	X	X	X						X																											
2	Low effectiveness of elicitation	X	X	X	X						X																											
3	The construction of a library of decisions			X	X	X	X											X					X		X				X								X	
4	No or low insight into relationships between artefacts												X											X	X	X	X	X	X	X	X	X	X	X	X	X		X
5	Reducing effort in design (requirements and specifications)	X							X	X	X	X	X																									
6	Shortening the design phase	X					X		X	X	X	X	X																									
7	Low productivity in the design phase	X				X	X	X	X	X	X	X	X																									
8	Low effectiveness in design	X				X	X	X	X	X	X	X	X																									
9	Mobilizing experts								X	X	X	X					X	X	X																			
10	Business rules mapping as much as possible																			X			X															X
11	Quality assurance validation and verification															X	X	X	X	X	X			X	X	X	X			X	X	X						
12	Automated verification of business rules	X														X								X														
13	Reducing efforts at verification	X														X								X														
14	Low productivity of verification	X														X								X														
15	Low effectivity of verification	X														X								X														
16	Generate automated test cases																	X	X																			
17	Automated testing with validation	X																X	X																			
18	No insight which artefacts are hit												X							X									X	X	X	X	X	X	X	X	X	X
19	Validated and accessible business rules															X	X	X	X	X	X		X															X
20	Reducing testing for implementation independent models															X	X	X	X	X	X								X	X								
21	Reducing testing for implementation-dependent models															X	X	X	X	X	X	X							X	X								
22	Reducing effort for validation	X															X	X	X	X	X																	
23	Working with implementation independent business rules to export models								X	X	X	X		X	X																							
24	Simplify converting from models into code													X	X																		X					
25	Separating of implementation 'know' and 'flow'		X	X	X									X	X																							

3.2.8 *Deriving elementary problem-solving actions*

The ideal next step is that of deriving elementary problem-solving actions by comparing design solutions with design problems. Out of these elementary problem-solving actions, method fragments are created. Winter (2011a) focuses on a change in maturity (in different research fields), as-is to a to-be situation, which could be supported by method fragments. The situation in this study is different;

this research focusses on supporting the implementation of a BRMS. Therefore, there is a phase where is no BRMS and a phase where there is a BRMS implemented. This is the same for each observed case, and the road to implementing a BRMS is different for each case. This is the same for each observed case, and the road to implementing a BRMS is different for each case. Nonetheless, it is still possible to create method fragments to support solving design problems when implementing a BRMS. A problem related to the elicitation problem class is given down below. Design problem #1 low productivity of elicitation, is proposed to be solved with a certain configuration of characterising design factors. Design problem #1 could be solved with a combination of the following design factors (as shown in Figure 19):

- #5 Capability autonomy
- #6 Elicitation source (Data)
- #7 Relevant set of selected sources
- #8 Scenario analysis
- #14 Role: Input

Design factor #5, #6, #7, #8 and #14 result into method fragment #2, #3, #4, #5 and #12. The combination of these design factors and their created method fragments will evolve into a method which could solve design problem #1; this example is shown down below in Figure 19.

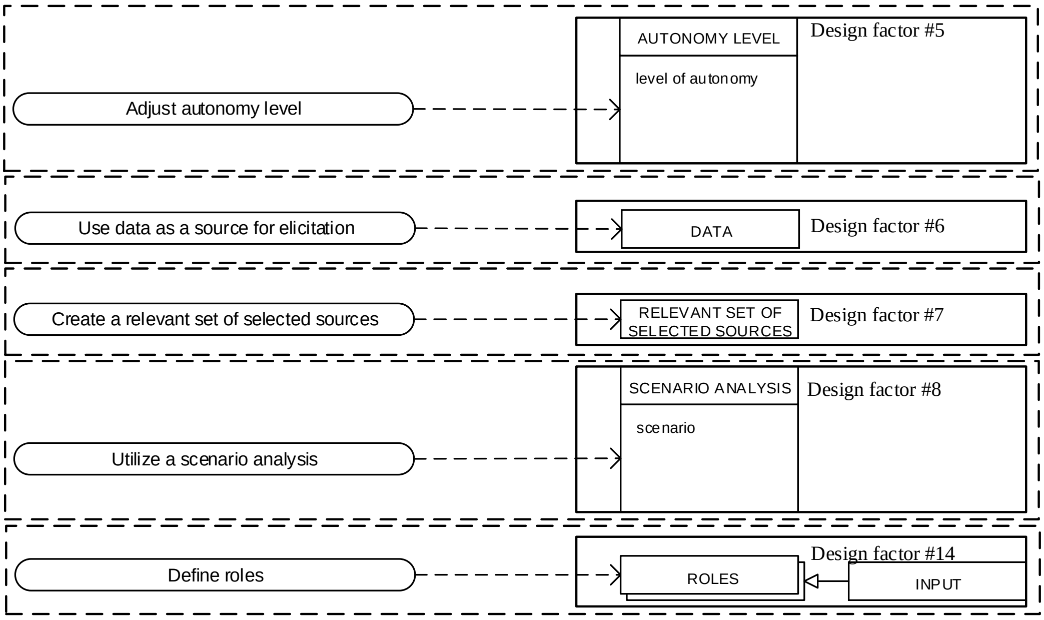


Figure 19 Method fragment example

The other method fragments and their related combination of characterising design factors are shown in Appendix F – Characterising design factor method fragments. Having these design problem solving method fragments requires certain rules to configure and use these method fragments into a complete problem solving solution to support the implementation of a BRMS. The method configuration rules are created into the next section.

3.2.9 Deriving method configuration rules

Based on the output of the previous sections, method configuration rules can be derived. The method fragments identified in the earlier sections need to be related to respective design situations and which design problems these method fragments solve.

Configuration rules need to be designed which guides the configuration of solutions to solve specific design problems. Merging the method fragments into one super method is not sufficient for solving the design problems. A certain combination of design problems and design factors requires additional information and attention. Important is that characterising design factors related to a certain problem class automatically means that a whole problem class (capability) is implemented. An example of this is as followed:

Design problem #1 low productivity of elicitation need to be solved. The design factors related to this design problem are:

- #5 Capability autonomy
- #6 Elicitation source (Data)
- #7 Relevant set of selected sources
- #8 Scenario analysis
- #14 Role: Input

These design factors are related to method fragments #2, #3, #4, #5 and #12. This configuration results in a method as shown down below in Figure 20.

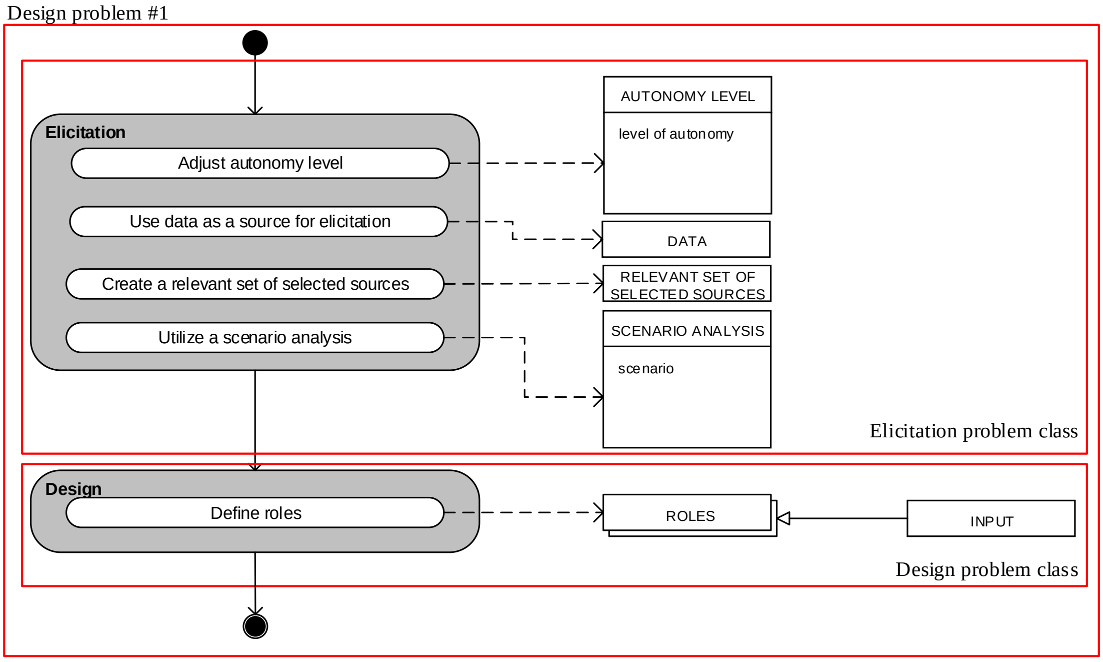


Figure 20 Method configuration example 1

It is not possible to only implement certain parts of a problem class. Therefore, the whole problem class is implemented together with the related non-characterising design factors and the related characterising design factors (the method fragments related to the non-characterising design factors are shown in

Appendix G – Non-characterising design factor method fragments). Therefore a notion arises, this notation is as followed: When a method fragment is selected to solve a problem, the whole problem class related to that method fragment must be implemented.

- This includes characterising design factors
- This includes non-characterising design factors

An example of such a complete method configuration, together with its related problem classes, is shown down below.

Design problem #1 low productivity of elicitation need to be solved. The method fragments related to this design problem are method fragments #2, #3, #4, #5, and #12. These method fragments are in their turn related to the Elicitation problem class (#2, #3, #4 and #5) and the Design problem class (#12). The method fragments related to the elicitation problem class (N1) and the design problem class (N2) are added to the previous method (Figure 21) resulting in a method consisting of characterising design factors and non-characterising design factors as shown down below.

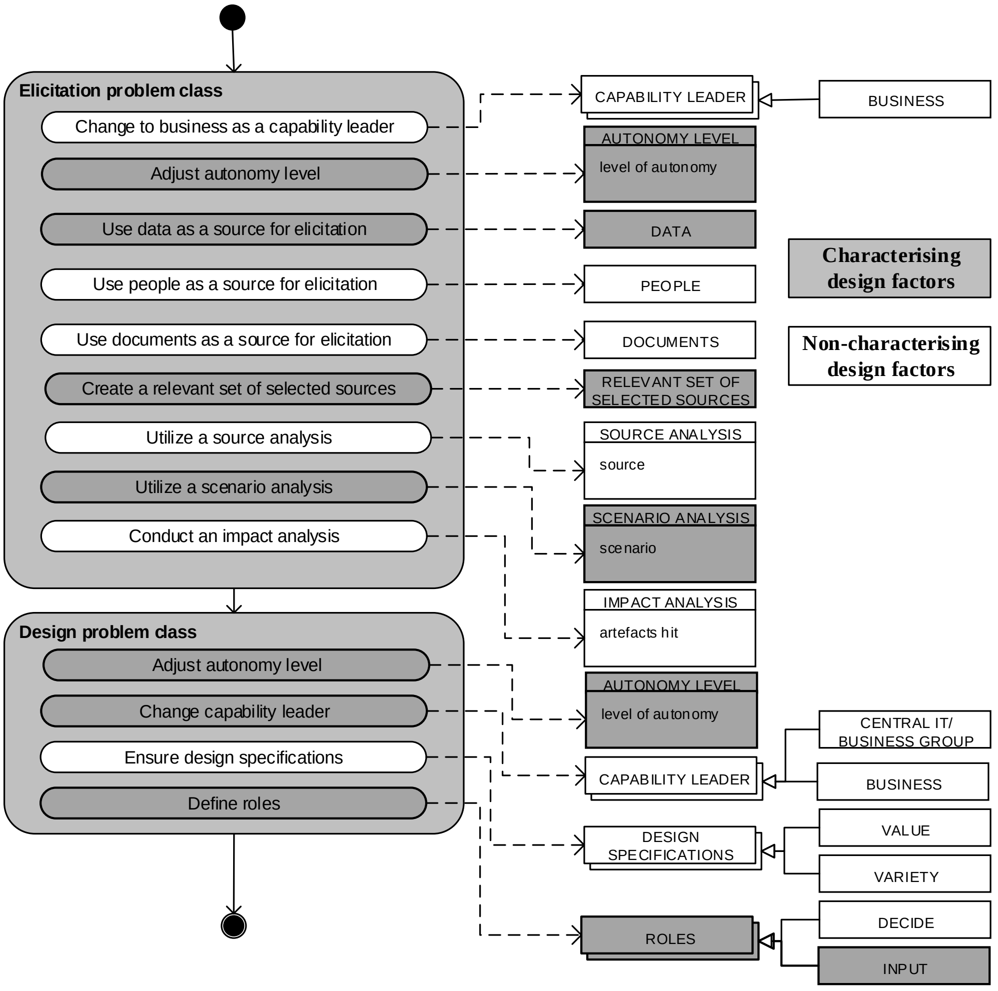


Figure 21 Method configuration example 2

Figure 21 consists of the method fragments which are related to the characterising design factors and method fragments which are related to the non-characterising design factors.

In order to create a method, as shown in the previous example, certain rules are needed. This results in a set of configuration rules which includes method fragments with all the characterising design factors (#) and non-characterising design factors (N) which relates to the method fragments aimed to solve a specific design problem. The method configuration rules are shown in Table 13 down below.

Table 13 Method configuration rules

Design problem #:	Design problem:	Implement a method consisting of method fragments:
1	Low productivity of elicitation	N1, N2, and #1 - #13
2	Low effectiveness of elicitation	N1, N2, and #1 - #13
3	The construction of a library of decisions	N1, N2, N4, N5, #1 - #13, #17 - #21, and #22 - #37
4	No or low insight into relationships between artefacts	N2, N6, #1, #2, #6 - #13, #24 - #36, and #38
5	Reducing effort in design (requirements and specifications)	N2, #1, #2, and #6 - #13
6	Shortening the design phase	N2, #1, #2, and #6 - #13
7	Low productivity in the design phase	N2, #1, #2, and #6 - #13
8	Low effectiveness in design	N2, #1, #2, and #6 - #13
9	Mobilizing experts	N2, N3, N4, #1, #2, #6 - #13 and #17 - #21
10	Business rules mapping as much as possible	N4, N5, N6, #1, #2, #17 - #23, and #38
11	Quality assurance validation and verification	N3, N4, #1, #2, #16 - #21, and #24 - #37
12	Automated verification of business rules	N3, #1, #2, #16, and #24 - #37
13	Reducing efforts at verification	N3, #1, #2, #16, and #24 - #37
14	Low productivity of verification	N3, #1, #2, #16, and #24 - #37
15	Low effectivity of verification	N3, #1, #2, #16, and #24 - #37
16	Generate automated test cases	N4, #2, #17 - #21
17	Automated testing with validation	N4, #2, and #17 - #21
18	No insight which artefacts are hit	N2, N4, N6, #1, #2, #6 - #13, #17 - #21, and #24 - #38
19	Validated and accessible business rules	N3, N4, N5, N6, #1, #2, #16 - #23, and #38
20	Reducing testing for implementation independent models	N3, N4, #1, #2, #16 - #21, and #24 - #37
21	Reducing testing for implementation-dependent models	N3, N4, N5, #1, #2, and #16 - #37
22	Reducing effort for validation	N4, #2 and #17 - #21
23	Working with implementation independent business rules to export models	N2, #1, #2, and #6 - #15
24	Simplify converting from models into code	#1, #2, #14, #15, and #24 - #37
25	Separating of implementation 'know' and 'flow'	N1, #1 - #5, #14, and #15

Certain design situations indicate that there exist implementations which consciously did not have a design factor implemented even if this design factor should solve a specific design problem. The configuration of a design situation depends on the values of design factors #1 Sector implementation, #2 Employee range and, #3 Scope focus. A specific value of these three design factors gives an indication which design situation fits the organisation needs. An example to illustrate such a process:

An organisation wants to implement a BRMS with a goal to solve design problem #1 (Low productivity of elicitation) and has the following values on design factor #1, #2, and #3:

Sector implementation:	Public sector
Employee range:	Organisations with 2001 – 5000 employees
Scope focus:	Organisation implementation focused

To solve design problem #1 and #2, all the method fragments related to the Elicitation problem class and Design problem class should be implemented, as explained in the previous examples. The given values of design factor #1, #2, and #3 indicate that the proposed implementation relates best to design situation 1 of the elicitation problem class and design situation 2 of the design problem class.

Elicitation problem class
Design Situation 1:

- **Public sector focused**
- **Organisations with 2001 – 5000 employees**
- **Organisation implementation focused**
- Fully manual elicitation focused (autonomy level 1)
- Database data is used for elicitating business rules
- Outcome of the capability is not a relevant set of selected sources
- Scenario analysis is conducted

Design problem class
Design Situation 2

- **Public sector focused**
- **Organisations with 2001 – 5000 employees**
- **Organisation wide implementation focused**
- The business is the capability leader
- Capability is supported, but a Man can veto before it runs automatically in a limited time (autonomy level 6)
- Velocity is taken into consideration in the design capability
- Volume is taken into consideration in the design capability
- Veracity is not taken into consideration in the design capability
- The agree role is defined in the design capability
- The perform role is defined in the design capability
- The input role is not defined in the design capability
- The recommend role is not defined in the design capability
- An impact analysis is conducted in the design capability

The required method to solve is already shown in Figure 22 and is discussed in the previous example, but due to the specific values of the required design situations, the method could be more specified to this design situations. This results in the method shown down below.

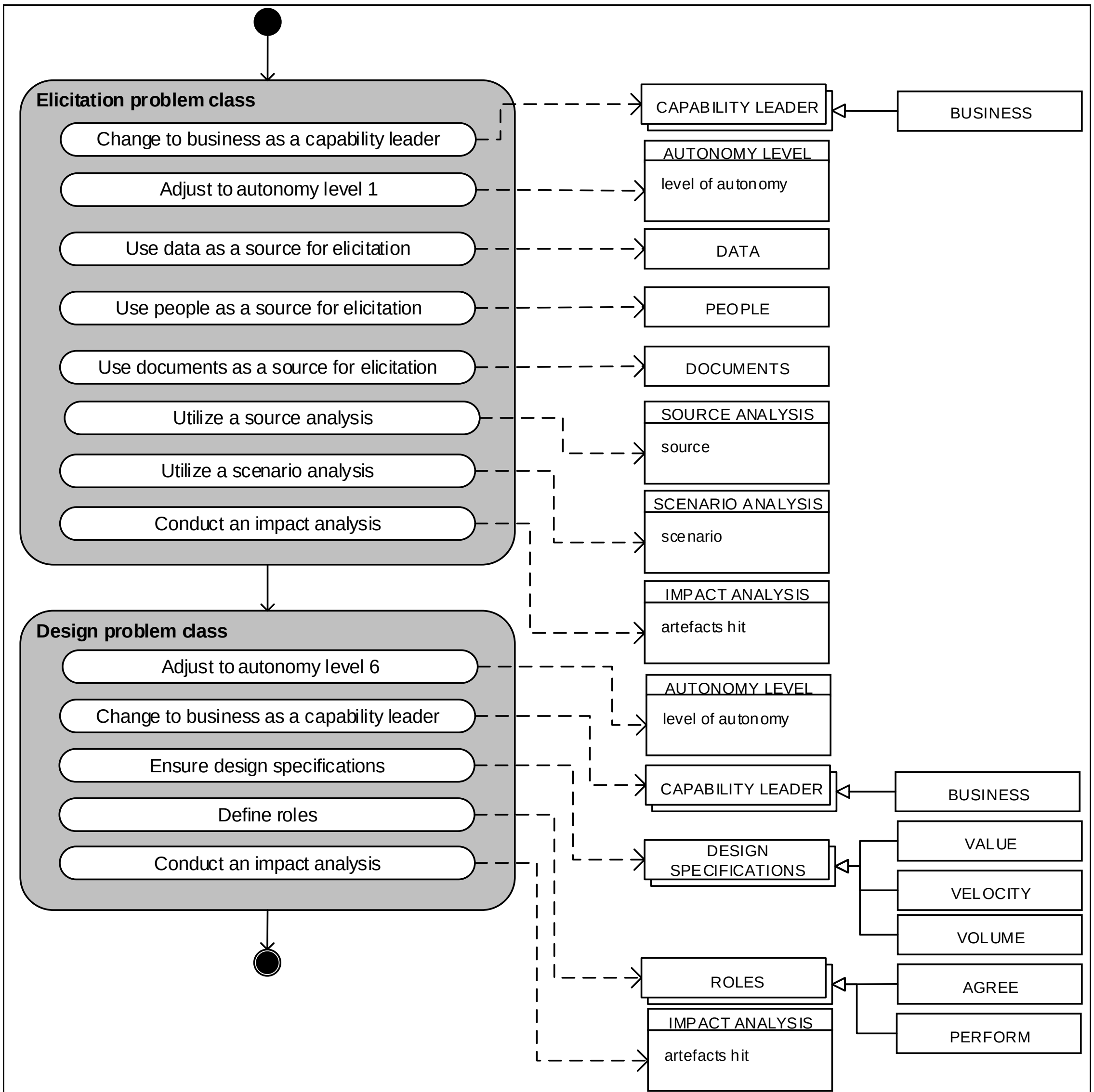


Figure 22 Method configuration example 3

The previous example results in the following notion:

> A specific design situation should be selected to further specify the method which solves a certain design problem or problems. The selection of a specific design situation depends on the following design factors:
>
> - Design factor #1 (public or financial sector)
> - Design factor #2 (< 50, 50 – 100, 101 – 250, 251 – 500, 501 – 1000, 1001 – 2000, 2001 – 5000 and, > 5000 employees)
> - Design factor #3 (application focused, line of business focused and, organisation-wide)

Design factor 1,2, and 3 are identified as the situational factors of implementing a BRMS. The choice of design situations has effect on the selection of method fragments. This could be: including (when the value is Yes) or excluding specific method fragments (when the value is No), or different values of such method fragments (e.g. different autonomy levels 1 – 10). The method configuration rules which indicate which design situation should be implemented are shown in Table 14. This table contains all the possible combinations of situational factors which could be extracted out of the gathered BRMS implementation cases.

Table 14 Design Situation Configuration Rules

Situational Factors			Design Situations per Problem class								
Sector	**Employees**	**Implementation focus**	**Elicitation**	**Design**	**Specification**	**Verification**	**Validation**	**Deployment**	**Execution**	**Monitoring**	**Governance**
Public	251 – 500	Application focused		4							
Public	1001 – 2000	Line of business focused						1			2
Public	2001 – 5000	Application focused									3
Public	2001 – 5000	Line of business focused			1						
Public	2001 – 5000	Organisation wide	1	2	2	2	1	2	1	4	1
Public	> 5000	Line of business focused	2						2		
Public	> 5000	Organisation wide			5			5			
Financial	251 – 500	Line of business focused	4	3	5				5		
Financial	501 – 1000	Line of business focused			4		3	4			
Financial	1001 – 2000	Line of business focused				3			3	1	
Financial	1001 – 2000	Organisation wide								2	
Financial	2001 – 5000	Line of business focused			3				4	3	
Financial	2001 – 5000	Organisation wide		1		1					
Financial	> 5000	Line of business focused	3					3			

3.3 The BRMS metamodel

The previous sections resulted into the identification of important elements of a BRMS together with each their specific value. The elements are created into the BRMS metamodel, as shown in Figure 23. This metamodel shows the relations between the identified elements with each their specific number present in a BRMS implementation. The BRMS metamodel consist of the identified SITUATIONAL FACTORS: EMPLOYEE RANGE, SECTOR, and IMPLEMENTATION FOCUS, the identified PROBLEM CLASSES: ELICITATION, DESIGN, SPECIFICATION, VERIFICATION, VALIDATION, DEPLOYMENT, EXECUTION, MONITORING, and GOVERNANCE, the number of DESIGN PROBLEMS, METHOD FRAGMENTS, DESIGN FACTORS, and DESIGN SITUATIONS, and together with their relationships.

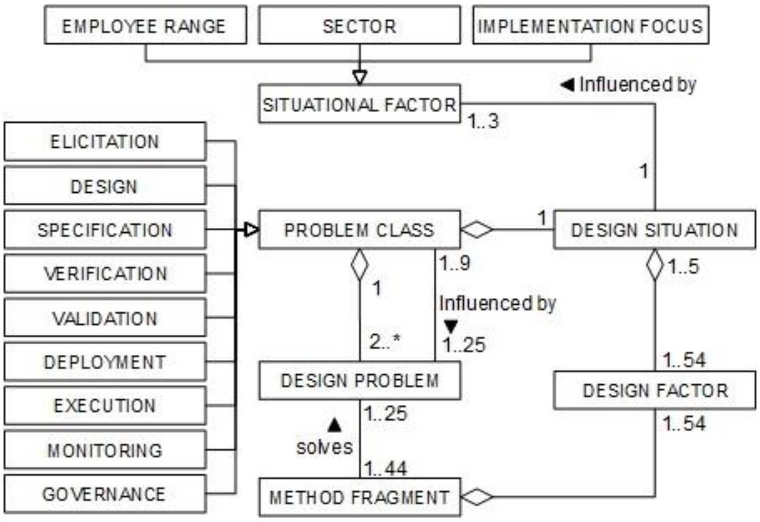


Figure 23 BRMS metamodel

BRMS construction process conclusion

The previous sections result into design situations, method fragments and configuration rules to support the creation of the BRMS implementation framework. 37 design situations were created distributed over the problem classes Elicitation, Design, Specification, Verification, Validation, Deployment, Execution, Monitoring, and Governance. These nine problem classes contain 54 design factors; 41 characterising design factors and 13 non-characterising design factors. The characterising design factors ensure with a different configuration for the characterisation of the design situations. The non-characterising design factors only have one specific value (e.g. only Yes). Nonetheless, the non-characterising design factors are added to the BRMS implementation framework for the sole reason that all the BRMS implementation cases indicate the same value (e.g. only Yes) and therefore are deemed important for the implementation of a BRMS. 25 design problems were identified from the goals and arguments why an organisation should implement a BRMS. The 25 design problems are linked to design factors with the purpose to solve the design problems. Method fragments were created to guide the selection of a specific set of design factors to solve a design problem or problems. The related design situation together with the method fragments results into a recommendation which could be used to implement a BRMS in a specific organisation of a specific size in a specific sector.

The situational artefact construction technique (Winter, 2011b) was used as a guideline for this process. Due to the small number of BRMS implementation cases, specific elements of the situational artefact construction technique could not be followed but is nonetheless still used as a guidance towards a situational artefact in the BRMS domain. In the next chapter the metamodels are created, which support the creation of the BRMS metamodel support tool and to support the comparison of BRMS implementations to-be and the BRMS which are successfully implemented.

3.4 The BRMS metamodel support tool

Methods, concepts and rules were introduced in the previous chapters and will be used for the creation of the BRMS metamodel support tool. This section will elaborate further on these artefacts and their relationship, resulting into a metamodel. The first stage of the metamodel will be the platform independent metamodel (PIM). This metamodel is not adapted to certain software or programming language requirements. The PIM will evolve into the platform dependent metamodel (PDM) which is adapted to defined software or programming language requirements.

3.4.1 Platform independent metamodel (PIM)

The PIM is created containing the elements identified in the earlier chapters. This metamodel will contribute to the actual creation of the framework. The PIM is the stepping stone towards the dependent specification of the framework (PDM), which is done in the next section. The elements of the PIM (which is shown in Figure 24). All the elements of the PIM are the elements identified in this study (SITUATIONAL FACTORS (SECTOR, EMPLOYEE, and FOCUS), DESIGN PROBLEMS, METHOD FRAGMENTS, PROBLEM CLASSES (ELICITATION, DESIGN, SPECIFICATION, VERIFICATION, VALIDATION, DEPLOYMENT, EXECUTION, MONITORING, and GOVERNANCE), DESIGN FATOR, and DESIGN SITUATIONS.

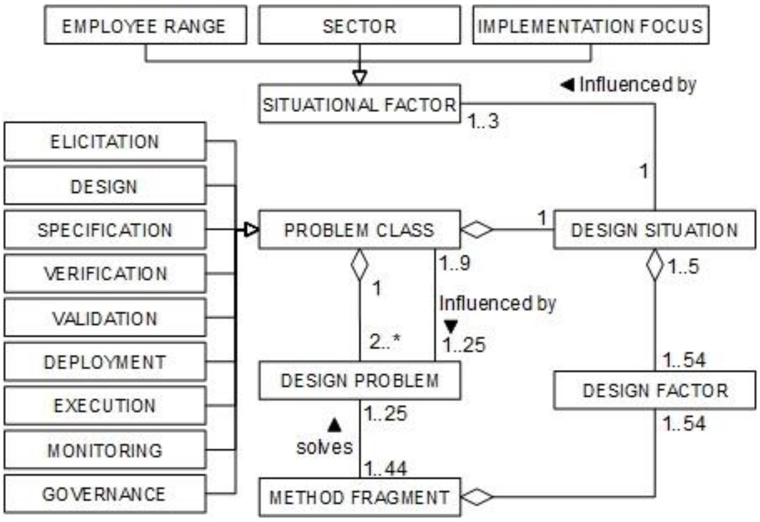


Figure 24 Platform independent metamodel (PIM)

The elements of the PIM are specified in Table 15.

Table 15 PIM specification

ID:	Element name:	Description:
1	SITUATIONAL FACTOR	The SITUATIONAL FACTORS are factors specific for a different situation. The SITUATIONAL FACTORS are a configuration of EMPLOYEE RANGE, SECTOR, and IMPLEMENTATION FOCUS and influences the specific configuration of elements included into a BRMS implementation.
2	DESIGN FACTOR	The DESIGN FACTORS are elements of a BRMS and could differ from value for each different implementation. The DESIGN FACTORS are included in the METHOD FRAGMENTS and in the DESIGN SITUATIONS.
3	DESIGN PROBLEM	The DESIGN PROBLEM indicate which design problems a specific organisation has and which should be solved by implementing elements of a BRMS.
4	METHOD FRAGMENT	The METHOD FRAGMENTS element should be implemented to solve a specific DESIGN PROBLEM. The METHOD FRAGMENTS exists of a number of DESIGN FACTORS.
5	PROBLEM CLASS	The PROBLEM CLASS element is a classification of a set of design problems, and which are identical to the known BRMS capabilities

		(ELICITATION, DESIGN, SPECIFICATION, VERIFICATION, VALIDATION, DEPLOYMENT, EXECUTION, MONITORING, and VALIDATION) . Each PROBLEM CLASS has multiple DESIGN SITUATIONS out of which exactly one can occur. The number of PROBLEM CLASSES are influenced by a specific configuration of DESIGN PROBLEMS.
6	DESIGN SITUATION	The DESIGN SITUATION element is a specific configuration of design factors specified to a certain configuration of SITUATIONAL FACTORS of the specific organisation.

The elements of the BRMS metamodel support tool also consist of numerators which contain the actual data which is a result of this research. The numerators are shown in Figure 25.

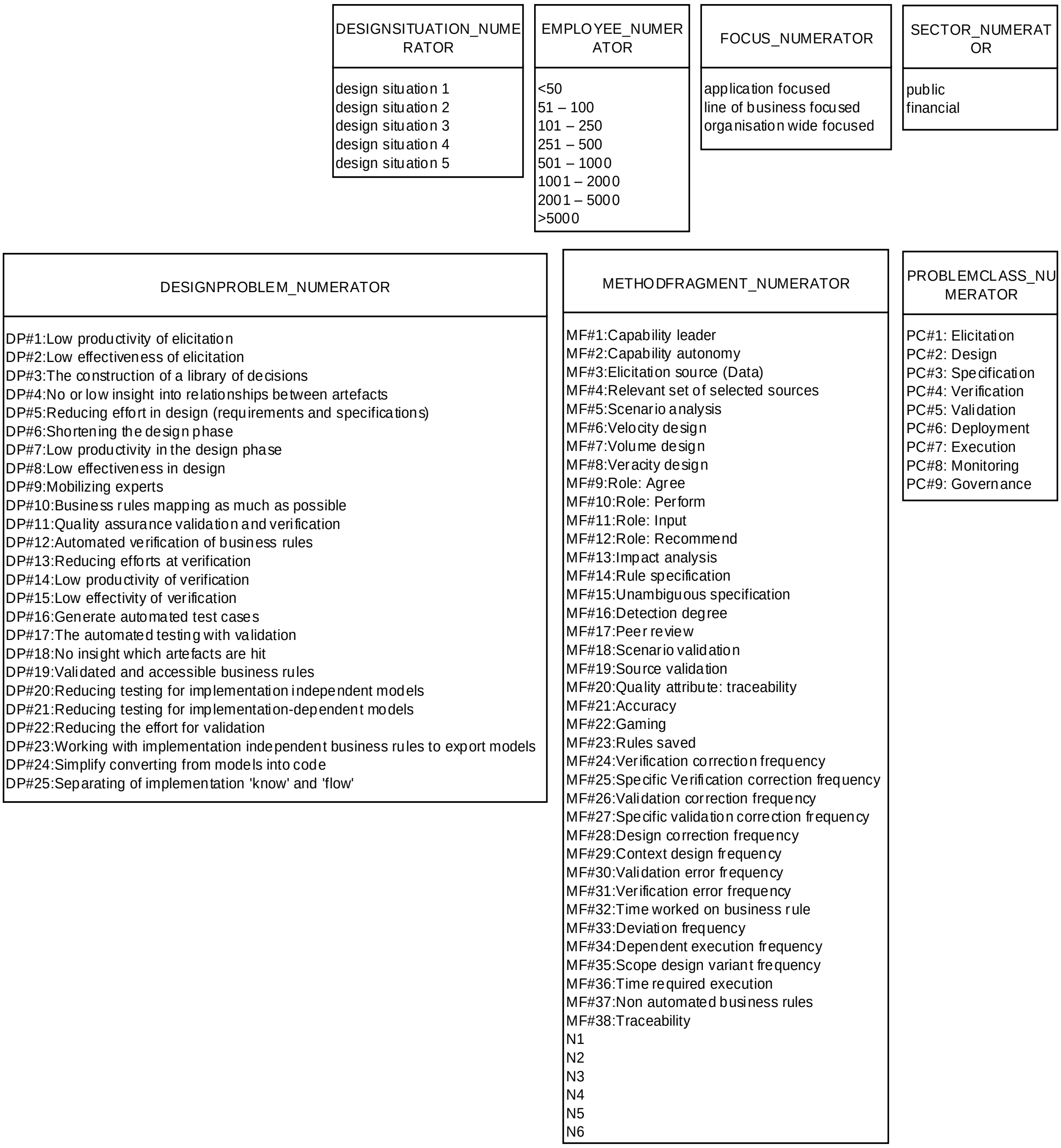


Figure 25 PIM - numerators

The numerators of the PIM are specified further in Table 16.

Table 16 PIM numerator specification

ID	Element name:	Description
8	DESIGNPROBLEM_NUMERATOR	The DESIGNPROBLEM_NUMERATOR consists of the 25 known DESIGN PROBLEMS.
9	METHODFRAGMENT_NUMERATOR	The METHODFRAGMENT_NUMERATOR consists of the 44 known METHOD FRAGMENTS.
10	PROBLEMCLASS_NUMERATOR	The PROBLEMCLASS_NUMERATOR consists of the 9 known PROBLEM CLASSES of BRMS.
11	SECTOR_NUMERATOR	The SECTOR_NUMERATOR consists of the sector situational factors FINANCIAL and PUBLIC.
12	EMPLOYEE_NUMERATOR	The EMPLOYEE_NUMERATOR consists of the employee range situational factors <50, 51 – 100, 101 – 250, 251 – 500, 501 – 1000, 1001 – 2000, 2001 – 5000, and >5000
13	FOCUS_NUMERATOR	The FOCUS_NUMERATOR consist of the different implementation focuses an ORGANISATION could have when implementing a BRMS.

3.4.2 Platform dependent metamodel (PDM)

The BRMS metamodel support tool is created using the model-driven development tool Eclipse Graphical Modelling Framework (GMF). The Eclipse implementation is focussed on delivering code which can be implemented into existing enterprise tools. The platform independent metamodel (PIM) is translated into a platform dependent metamodel (PDM) which is specified towards working with Eclipse. Additional relations are included to explicitly visualise what elements are included in other elements. The numerator data is relatively the same as with the PIM but is adapted towards working with Java, which is the final result of working with Eclipse GMF. The PDM is shown in Figure 26, and further specified in Appendix H – Specifications of the Platform Dependent Metamodel. Some elements were added and excluded in the PDM compared to the PIM because of the requirements of working with Eclipse.

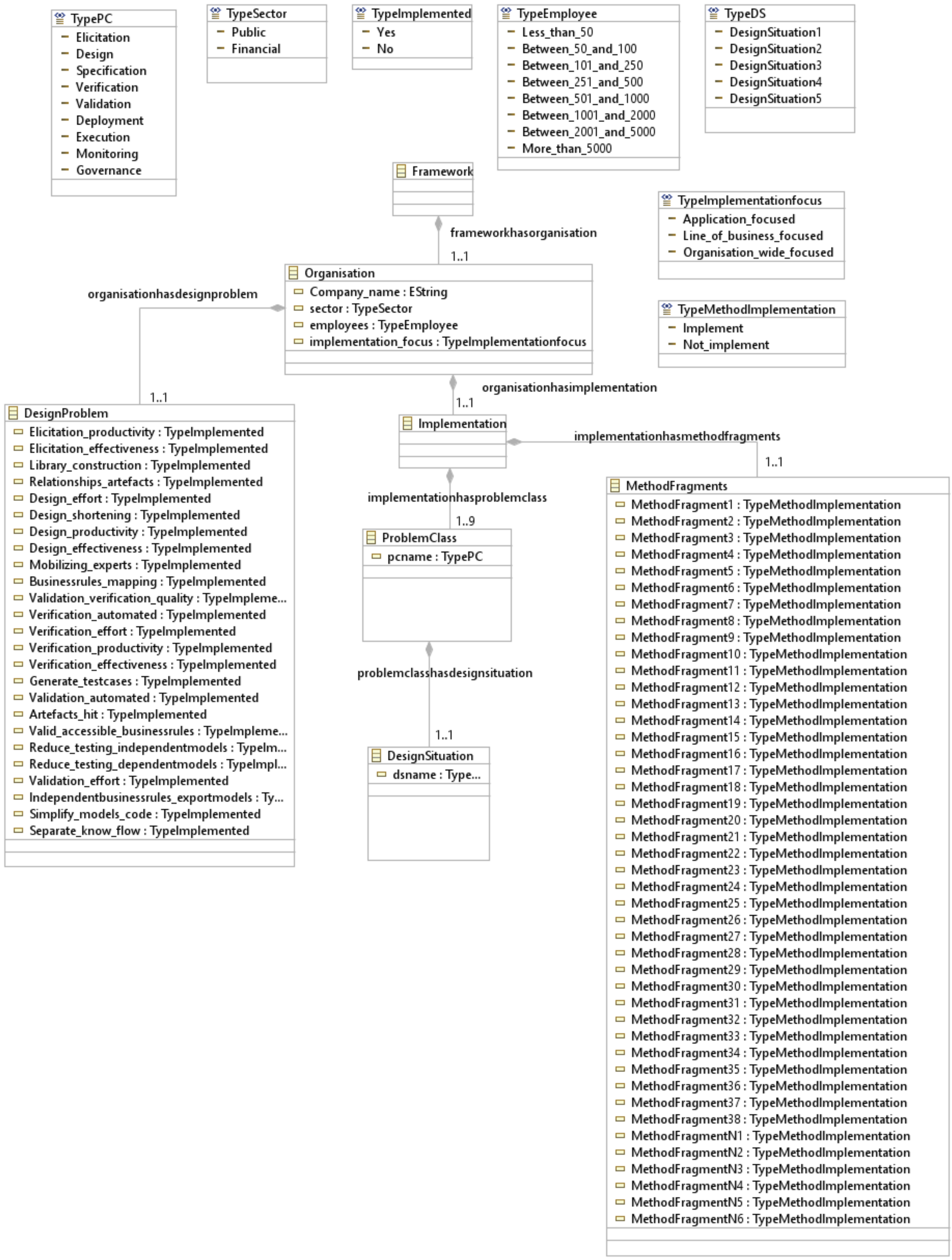


Figure 26 PDM BRMS metamodel support tool

3.4.3 *The BRMS metamodel support tool implemented*

The PDM is implemented and by utilising Eclipse GMF the BRMS metamodel support tool[2] can be created, which supports an organisation of setting up a specific configuration of their BRMS implementation. Figure 27 shows an example of how to solve a specific set of design problems when being an organisation in the public sector, with 2001-5000 employees, and with an organisation wide implementation.

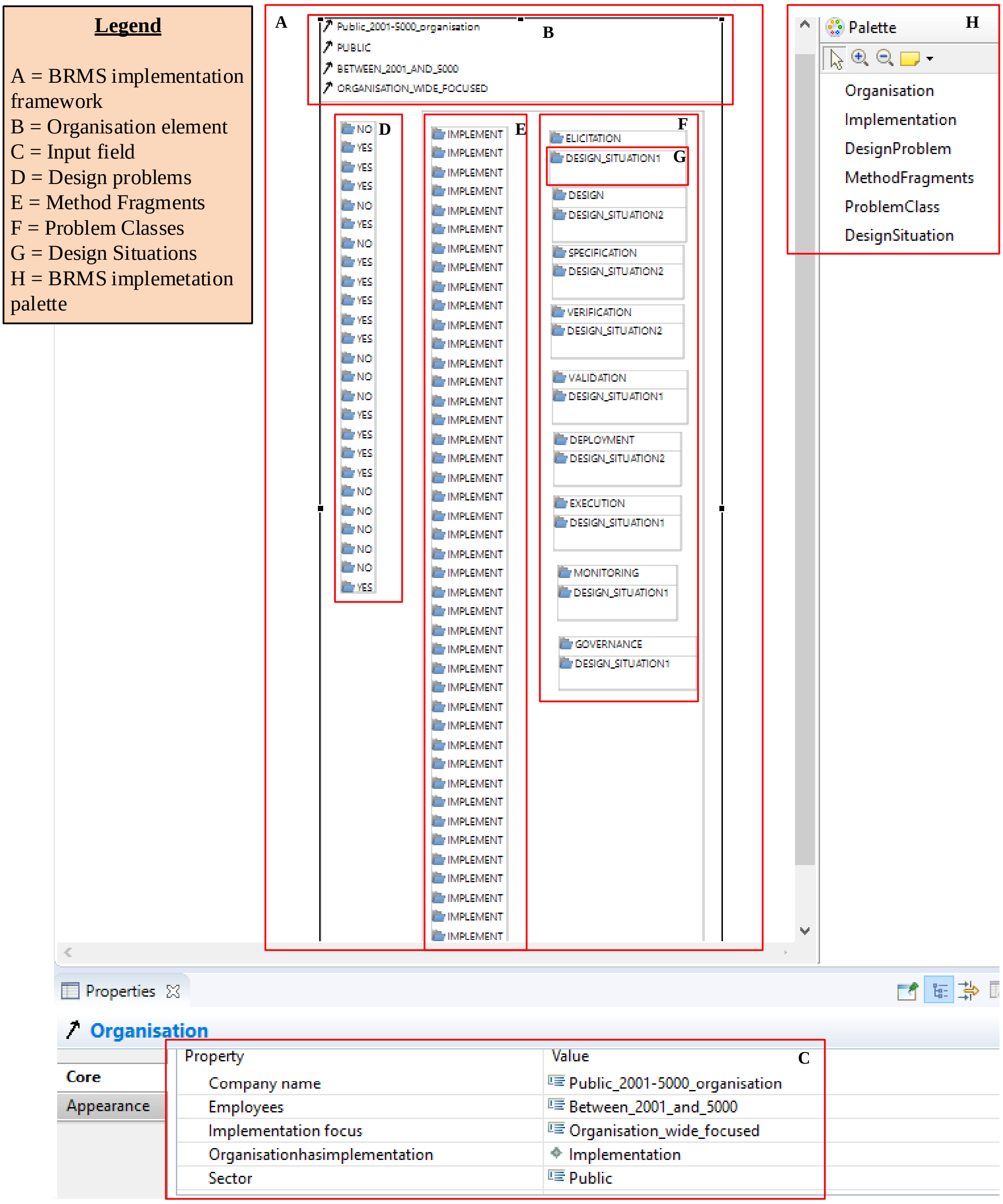


Figure 27 BRMS implementation framework tool elements

---

[2] BRMS metamodel support tool: https://www.dropbox.com/s/j2b0til1zrxbrzl/Project%20BRMS%20Framework_ECLIPSE.rar?dl=0

The BRMS implementation framework (A)

Contains all the elements which support the implementation of a BRMS.

Organisation element (B)

Contains the fields with organisation specific data (situational factors). These are the sector, number of employees working at the organisation, and the implementation focus.

Input field (C)

This element serves as an input field for all the elements containing in the BRMS implementation framework.

Design Problems (D)

The problems existing at a specific organisation which could be solved by implementing a BRMS. The design problems can be indicated with a ‘Yes’ when the design problems exist or with a ‘No’ when not.

Method Fragments (E)

The method fragments are the elements which should be implemented when aiming to solve a specific configuration of design problems.

Problem Classes (F)

The problem classes are elements which are implemented related to the configuration of design problems. The problem classes have each their own design situation specified for a specific organisation.

Design Situations (G)

The design situations are a specific setup of elements which are specified towards different organisations. Therefore, ensuring a specific solution for different organisations.

BRMS implementation framework palette (H)

The palette supports the creation of all the elements of the BRMS implementation framework. In Figure 27 a prepared example is already given. Therefore, no additional elements could be added, only when elements are deleted additional elements could be created.

3.5 How to use the BRMS implementation framework

The BRMS observation technique, the BRMS construction process, the BRMS metamodel, and the BRMS metamodel support tool are created. Thereby, resulting in the creation of the BRMS implementation framework. This section focusses on how to use the BRMS implementation framework for utilisation by organisation aimed at implementing a BRMS or for future research. Figure 28 shows a process on how to use the BRMS implementation framework. It all starts with the need of implementing a BRMS. The BRMS observation technique is first used to gather BRMS implementation cases. The next step is the BRMS construction process, which contain specific steps to analyse the BRMS implementation cases gathered in the previous step. The BRMS construction process analyses the BRMS implementation cases. The result of the BRMS observation technique and the BRMS construction process is specified into a BRMS metamodel. The BRMS metamodel represents the elements identified in the previous two steps together with their relations and multiplicities. The BRMS metamodel is a representation of the elements occurring in a BRMS implementation. The next step is supporting the elicitated data using the BRMS metamodel support tool, which is developed through a model-driven development approach (Eclipse GMF). The of this activity are instantiations of BRMS implementation cases. The instantiations of the BRMS implementation cases can be utilised by organisations to find or create the most optimal configuration of implementing a BRMS in their specific environment.

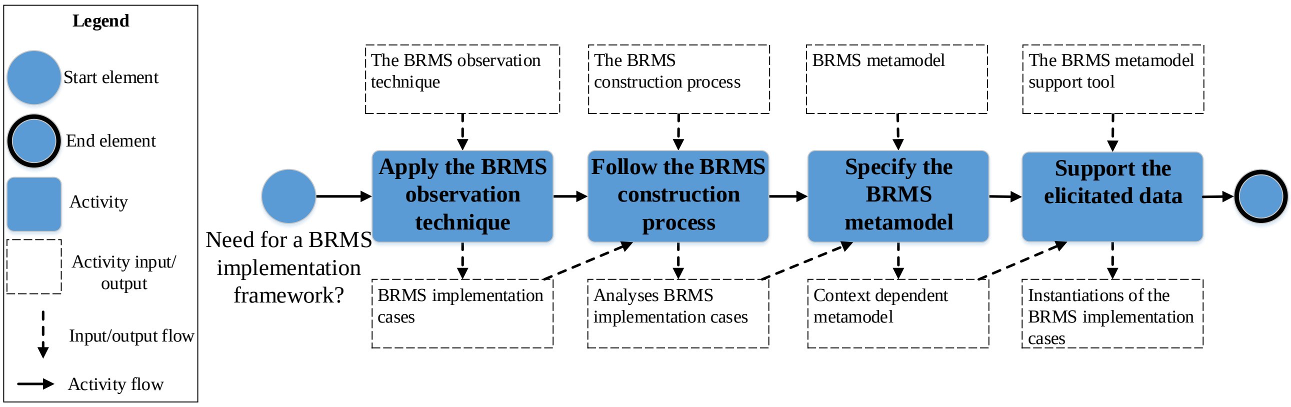


Figure 28 BRMS implementation framework process

The next chapter will focus on validating the BRMS implementation framework through expert interviews.

4 Validation of the BRMS implementation framework

This chapter contains the validation of the BRMS implementation framework. When following the initial BRMS construction process one of the first additions were made on the validation part, specific validation with people from practice. The research fields of BRM and BRMS lacking standards and frameworks and therefore comparison and validation is needed from practice (Nelson et al., 2008; Zoet, 2014). Semi-structured validation expert interviews were conducted with the focus of validating the correctness of the BRMS implementation framework and its containing elements. The experts were selected on their knowledge on the topics of BRM and BRMS. The same selection is made as with the pilot of the BRMS observation technique, which focused on specific levels of experience and knowledge in the BRM and BRMS field. The experts consisted of person 1 which is a professor lecturing and performing research in the field of BRM and BRMS, person 2 which is a lecturer and PhD with 6 years of practical and research experience in the field of BRM and BRMS, and person 3 which is a lecturer with 3 years of practical and research experience on BRMS capabilities. An interview protocol was used during the expert interviews, and this could be found in Appendix I – Expert interview protocol.

4.1 Validation expert interviews

The elements of the BRMS framework (problem classes, design factors, design situations, and design problems) are proposed to the experts and from their expertise which possible changes should be made and which elements should be included or excluded. Besides stating of elements are correct or not correct, examples from practice were asked from the expert so to conclude its validity.

The elements which are a result of statistical analysis where nonetheless discussed and when a certain exclusion of inclusion of an element was a point of discussion by the expert this was noted to take into consideration for future research. The comments of the experts influence the correctness of the framework and thereby validating the correctness of the process of creating a situational artefact in the BRM and BRMS research field. The structure of the validation expert interviews together with the demonstration of the framework is visualised in Figure 29.

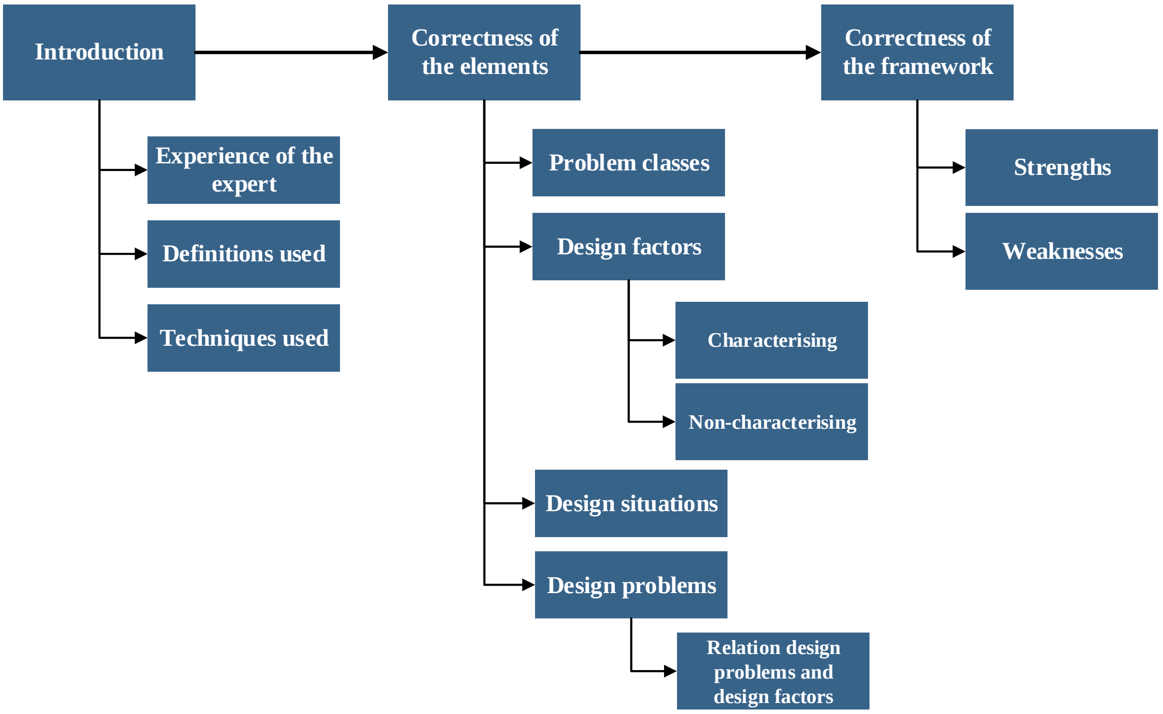


Figure 29 Validation expert interviews breakdown structure

Problem classes

The nine problem classes of the BRMS were proposed to the experts and were agreed on. Furthermore, the overlap of the names of the problem classes and the BRMS capabilities were not indicated as a weakness but as a strength because of the similarities between a problem class and a capability.

Design factors

All the characterising and non-characterising design factors were proposed. Table 12 shows all the design factors which were point of discussion and were additional comments were given.

Table 17 Validation interview - design factors

ID:	Design factor:	Value:	Description:	Comment:
19	Detection degree	Automated - Detective Automated - Preventive Manual - Detective Manual - Preventive	Detection degree indicates at which degree syntax and semantic errors are detected	All the experts indicated that the value Manual – preventive seems an unlikely value as detection degree in a BRMS. An example was given of a manual – preventive degree: ‘’When an error is made by employee A and at the same time employee B is manually detecting the error employee A made.’’ Therefore, this is noted and taking into consideration when conducting future research.
42	Elicitation source (People)	Yes	People are used as a source for elicitation.	All the experts stated that the use of people as a source for elicitation is a possibility. But the possibility was given that the use of people is broad, and for future research, this needs more explanation.
44	Source analysis	Yes	Source analysis is used during elicitation. Analysis based on sources.	All the experts stated that this could be valid but very limited. Again source analysis needs more defining for future research.
45	Impact analysis	Yes	An impact analysis checks the impact that a certain change brings forth.	All the experts stated that it is highly unlikely that everyone is doing an impact analysis during elicitation. Again this could be in the broadest sense of the word.
46	Value design	Yes	Value is one of the five V’s and represents the value of the decision to the organisation.	One expert stated that this is not the case for all the organisations included in the sample.
50	Quality attribute: Usability	No	The validation capability checks on the quality attribute usability.	The experts agree on the validity of the given value; only the experts agreed if this is wise to not include in the implementation. Usability ensures the understandability and usefulness of the business rule if this is not included other people who didn’t have the expertise to read this are left out of the process.

52	Output saved	Yes	The output is saved in the execution capability.	All the experts disagreed on the value of the output saved design factor. Experience was shared on organisations who deliberately do not save output data.
53	Version management	Yes	The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities.	The experts found it highly unlikely that all the organisations made use of version and validity management. Again in a somewhat limited form. An example was given for validity management: ‘’Organisations would state that they do validity management, but this is likely on database level and not defined in the BRMS.’’
54	Validity management	Yes	The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time.	

Design situations

The design situations were proposed to the experts, and the notion was given that the configuration of the design situations is a result of statistical analysis. Nonetheless, the experts saw this as a helpful tool to further specify the solution for specific BRMS implementation cases and validated the correctness of the design situations.

Design problems

The design problems were already validated at the start of this study with a combination of the state-of-the-art literature review, interviews with people from the BRM community, and BRMS implementation cases. Nonetheless, the experts agreed on the set of 25 design problems and indicated that the 25 design problems could overlap with other design problems which were possiblly not taken into the list of design problems. Thereby, concluding the correctness of the 25 design problems.

Overall framework

The Excel version of the framework was shown to the experts, and during this demonstration, all the working elements were demonstrated. Different configurations of the situational factors where showed together with the configuration of certain design problems, this leads to an advice of which method fragments should be implemented to solve the design problems, which problem class should be implemented, and which design situation of each problem classes should be selected.

During the expert interviews, all the elements of the framework and the framework as a whole were compared to real world examples and were validated as correct. During these expert interviews different execution paths where shown and if necessary, possible changes were made when the expert point it out as necessary. Additional explanation was given on the possible execution paths of the framework. This explanation was focused on the limited execution paths because of the low number of respondents. When the number of respondents is higher, the possible execution paths of the framework also increases. The comments of the experts on the overall working framework were positive, and no changes were proposed. This is due to the fact that the actual elements were validated in the previous sections and this demonstration sole purpose was to demonstrate the working framework to the experts.

The experts all agreed on the same strong and weaknesses of the correctness of the framework. The frameworks weakness was mainly based on the lack of a high number of BRMS implementation cases. The experts stated when the BRMS implementation cases sample would increase the power of the BRMS implementation framework would also increase. The experts complemented on the fact that such a large problem could be summarised in such and clear and useful tool to support the implementation of a BRMS.

4.1.1 Validation threats

During the validation phase, certain threats arose, which could impact the validity of the validation.

The different settings between the experts during the validation expert interviews could have impacted the different answers which were given. The length of the interviews is also an indication that there is a possible difference between the given answers of person 1 and 3 and between person 2. Therefore, the reliability of the outcome of the validation expert interviews is somewhat questionable because it is not clear if the difference in setting actual provided different answers for the validation of the BRMS implementation framework and its elements (Wohlin et al., 2012). Christmann (2009) states telephone interviews, like Skype-calls, the inter-action process may be disturbed due to the lack of non-verbal elements. Furthermore, the external factors like the presence of a third party are also unknown to the interviewer. Nonetheless, the interviews with person 1 and person 2 were conducted as a telephone interview due to the fact that the experts had a full agenda. Therefore, the choice is made for these types of interviews compared to no interviews at all.

The interview setup was also a treat towards the reliability. Proposing the elements of the framework to the experts possibly influenced the objectivity of the experts and thereby only focusing on the given elements of the framework. The knowledge of a framework supporting BRMS implementation was non-existent, and therefore the possibility exists that, without examples, the experts could not give examples of elements of a BRMS implementation framework.

The validation phase is an addition to the situational artefact construction phase, where validation is not a required step. Therefore, the threats to validity are not seen as a major impact towards the correctness of the created BRMS implementation framework. Nonetheless, the elements of discussion and the threats to validity are taken into consideration for future research.

5 Conclusion

Many organisations employ large amounts of business rules as part of their products or services to deliver added value to their customers. These business rules need to be elicited, designed, specified, verified, validated, deployed, executed, monitored, and managed in a proper way. Therefore, a Business Rules Management Solution is needed to support the configuration and implementation of capabilities. The current body of knowledge is extensive on the technical implementation of a BRMS, where BRMS stand for Business Rules Management System. However, the organisational implementation lacks knowledge on how to implement a BRMS, where BRMS stands for a Business Rules Management Solution. Therefore, support is needed to create a form of help to implement a BRMS from an organisational perspective. This is done by creating a framework. To address this problem we aimed to answer the following main research question:

''How to develop a framework that supports the organisational implementation of a business rules management solution?''

The goal of this research was to develop a framework to support the implementation of a business rules management solution from an organisational perspective. In order to achieve this goal, we explored the business rules management problem space and its neighbouring fields on techniques to create a framework. We selected the situational artefact construction technique of Winter (2011b) as a guideline for creating our framework. To be able to create a situational artefact, the state of the art of the BRM and BRMS research field needed to be measured. The state-of-the-art literature review resulted in a set of situational factors and problem classes in the BRM problem space. These situational factors and problem classes were validated by the BRM community. Thereby, providing an answer on RQ1: How to specify situational factors and problem classes in the business rules management problem space?

The state-of-the-art literature review resulted in the identification of a technique to create situational artefacts (Winter, 2011b). Using this technique as a guideline towards creating a framework to support the organisational implementation of a BRMS framework provided the answer on RQ2: How to design the business rules management solution implementation framework? One of the steps of the situational artefact technique proposed a field study to gather design solutions. This resulted into a BRMS observation technique aimed at gathering BRMS implementation cases. Successful BRMS implementation cases from 13 organisations were gathered. Analysis of the survey data using the BRMS construction process resulted into design situations, method fragments and configuration rules to support the creation of the BRMS implementation framework. 37 design situations were created distributed over the problem classes Elicitation, Design, Specification, Verification, Validation, Deployment, Execution, Monitoring, and Governance. These nine problem classes contain 54 design factors; 41 characterising design factors and 13 non-characterising design factors. The characterising design factors ensure the characterisation of the design situations of every problem class. The non-characterising design factors only have one value and therefore not characterising. The non-characterising design factors are included in the BRMS implementation framework for the sole reason that these elements have the same value in the BRMS implementation case sample. 25 design problems were identified from the goals and arguments to implement a BRMS. The 25 design problems are linked to design factors, which have as a purpose to solve these design problems. Method fragments were created to guide the selection of a specific set of design factors to solve design problems. The related design situation together with the method fragments results in a recommendation on how to implement a BRMS in a specific organisation of a specific size in a specific sector. The elements identified in the BRMS observation technique and the BRMS construction process resulted into the BRMS metamodel.

The BRMS metamodel is implemented in Eclipse Graphical Modelling Framework, which resulted into the BRMS metamodel support tool, which can be utilised as a tool for the support of the BRMS metamodel. The BRMS implementation framework was admitted to expert interviews to validate the correctness of the BRMS implementation framework. Thereby, providing an answer on RQ3: How to validate the correctness of the business rules management solution implementation framework?

5.1 Discussion

This research aimed to 1) specifying situational factors and problem classes in the BRM problem space, 2) designing the BRMS implementation framework, and 3) validating the correctness of the BRMS implementation framework. We believe that creating this framework to support BRMS implementation from an organisational perspective will contribute to the maturity of the BRM and BRMS field, creating a foundation towards other situational artefact research in general and in the BRM and BRMS field. However, we believe certain aspects of this research are susceptible to discussion.

One of the main limitations of this research is that situational artefact construction relies on large samples (60+ BRMS implementation cases at minimum) as input for the data analysis. Our sample consists of only 13 BRMS implementation cases. The effect of the small sample results in failing the requirements of performing a Principal Component Analysis (PCA), which requires a higher sample (60+ BRMS implementation cases at minimum) (Abdi & Williams, 2010; Arrindell & van der Ende, 1985; Comrey & Lee, 2013; Hogarty et al., 2005; Jolliffe, 2002; MacCallum et al., 1999). The initial quantitative PCA was replaced with a qualitative approach due to the fact of the small sample (n=13). This qualitative approach focusses on extra validation from practice compared to the initial PCA, which is not validated by people from practice. The possibility exists that important design factors could be excluded from the framework. Therefore, this qualitative PCA approach was used to not let important design factors be excluded from the framework.

Furthermore, the small sample size has an influence on the instantiation of design factors. During this research, an instantiation is made between characterising design factors and non-characterising design factors. To our knowledge, the non-characterising design factors will become characterising due to the fact that additional BRMS implementations will result in more possible configurations of the BRMS implementation framework. Therefore, providing with different values of design factors and by that reason become characterising design factors.

The low amount of experts (n=3) used for validation of the BRMS observation technique and the BRMS implementation framework are identified as a limitation and as a threat to the construct validity, and reliability. Being that the BRMS observation technique is an important element of creating the BRMS implementation framework. The possibility exists that the experts if biased, have a higher impact on the validity of the BRMS observation technique and the BRMS implementation framework when low in numbers. Thereby, a higher number of experts validating the BRMS observation technique and the BRMS implementation framework will reduce the possible impact of being biased.

Another limitation regards the actual working proof of the framework. The BRMS implementation framework is validated by experts in an expert interview. However, to support a real world BRMS implementation, additional proof is needed that the framework is valid and correct. Nonetheless, the validation of the BRMS implementation framework is performed in a controlled and rigorous way and can be seen as a contribution to the field of BRM and BRMS.

Lastly, this limitation regards the exploratory nature of this research concerning the use of situational artefact construction in an immature field. It is still not clear that using this technique is possible in immature fields and therefore possibly threats to validity could arise. However, the BRMS implementation framework and the BRMS construction process are validated with BRM experts from research and practice and deemed correct and valid.

5.2 Future research

The limitations described in the previous subsection result into possible directions for future research. One of the main limitations was the effect of the small sample size. Therefore, future research is needed with a larger and more diverse sample. Other sectors should be included in the sample. Thereby, ensuring different configurations of BRMS problem classes, which results in a more complete and correct framework. Furthermore, the larger sample will also dissolve the existence of the non-characterising design factors. These specific design factors exist due to the fact that the gathered data is limited. Therefore, studies with a larger variety (sectors, employee numbers, and implementation focus) in cases are needed d to evolve the non-characterising design factors in regular design factors. Future research needs to be focussed on an larger sample and not simply adding new BRMS implementation cases to the already existing sample. The already existing sample of BRMS implementation cases is gathered in a certain timespan and could be possibly influenced by additional BRMS implementation cases from another time. The advancement of new technologies could be a main influence in adding new BRMS implementation cases from a different timespan.

Besides the future research needed focused on a larger sample, the tool which is used to gather the BRMS implementation cases also needs more work on. Although the situational artefact survey is validated by 3 people from the BRM community, stating the importance of this survey, future research needs to be focussed on validation with a larger sample.

The tool supporting the BRMS implementation framework is rather immature and needs to be extended with automated configurations related to a set of situational factors. Now the possibility exists that BRMS implementation projects are shown focussed on a certain configuration of situational factors. This is needed to be merged into a version containing logic which will provide advice, based on a specific configuration of situational factors and design problems, which elements of a BRMS needed to be implemented to solve design problems specific to an organisation's situational factors.

Furthermore, future research is needed in the field of situational artefact construction focused on immature fields. The situational artefact construction technique is assuming that specific literature is known to extract situational factors and problem classes and that a questionnaire can be created based on a Likert-scale. The BRMS construction process (as shown in Figure 14) is an example of additional steps that should be taken when creating a situational artefact in an immature field. However, further research is needed to validate if this is the case in other immature research fields.

6 References

Aaker, D. A. (2008). *Strategic market management*. Hoboken, NJ: John Wiley & Sons.

Abdi, H., & Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, *2*(4), 433–459. http://doi.org/10.1002/wics.101

Aier, S., Riege, C., & Winter, R. (2008). Classification of enterprise architecture scenarios – an exploratory analysis. *Enterprise Modelling and Information Systems Architectures*, *3*(1), 14–23. http://doi.org/10.18417/emisa.3.1.2

Algemene Rekenkamer. (2013). Complexe of oncontroleerbare regels werken fraude soms in de hand. Retrieved July 4, 2016, from http://verantwoordingsonderzoek.rekenkamer.nl/2013/rijksbreed/themas/fraude/complexe-oncontroleerbare-regels-werken-fraude-soms-de-hand

Arnott, D., & Pervan, G. (2005). A critical analysis of Decision Support Systems research. *Journal of Information Technology*, *20*(2), 67–87.

Arrindell, W. A., & van der Ende, J. (1985). An empirical test of the utility of the observation-to-variables ratio in factor and components analysis. *Applied Psychological Measurement*, *9*(2), 165–178. http://doi.org/10.1177/014662168500900205

Bass, L., Clements, P., & Kazman, R. (2012). *Software architecture in practice* (3rd ed.). Boston, MA: Addison-Wesley.

Baumöl, U. (2005). Strategic agility through situational method construction. In *Proceedings of EURAM 2005 (European Academy of Management)*. Munchen.

Bekkers, W., van de Weerd, I., Brinkkemper, S., & Mahieu, A. (2008). The Influence of Situational Factors in Software Product Management: An Empirical Study. In *2008 Second International Workshop on Software Product Management* (pp. 43–50).

Bevan, G., & Hood, C. (2006). What's measured is what matters: Targets and gaming in the English public health care system. *Public Administration*, *84*(3), 517–538. http://doi.org/10.1111/j.1467-9299.2006.00600.x

Bogner, A., Littig, B., & Menz, W. (2009). *Interviewing experts*. *Production*. London: Palgrave Macmillan. http://doi.org/10.1057/9780230244276

Boyer, J., & Mili, H. (2011). Agile business rule development. In *Agile Business Rule Development* (pp. 49–71). Berlin, Heidelberg: Springer. http://doi.org/10.1007/978-3-642-19041-4

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology*, *38*(4), 275–280. http://doi.org/10.1016/0950-5849(95)01059-9

Bucher, T., & Winter, R. (2010). Taxonomy of business process management approaches. In *Handbook on Business Process Management* (pp. 93–114). New York, NY: Springer.

Burlton, R. (2001). *Business process management: profiting from process*. Indianapolis, IN: Sams Publishing.

Business Rules Group. (2003). *The Business Rules Manifesto*. Retrieved from http://www.businessrulesgroup.org/brmanifesto/BRManifesto.pdf

Chapin, N., Hale, J. E., Khan, K., Ramil, J. F., & Tan, W. G. (2001). Types of software evolution and software maintenance. *Journal of Software Maintenance and Evolution*, *13*(1), 3–30. http://doi.org/10.1002/smr.220

Charfi, A., & Mezini, M. (2004). Hybrid web service composition: business processes meet business rules. *Proceedings of the 2nd International Conference on Service Oriented Computing*, 30–38. http://doi.org/10.1145/1035167.1035173

Christmann, G. (2009). Expert interviews on the telephone: A difficult undertaking. In *Interviewing Experts* (p. 281). London: Palgrave Macmillan. http://doi.org/10.1057/9780230244276

Comrey, A. L., & Lee, H. B. (2013). *A first course in factor analysis* (2nd ed.). Hove: Psychology Press.

Date, C. J. (2000). *What not how: The business rules approach to application development* (1st ed.). Boston, MA: Addison-Wesley.

De Feo, J. A., & Barnard, W. (2003). *Juran Institute's six sigma: breakthrough and beyond: quality performance breakthrough methods* (1st ed.). New York, NY: McGraw-Hill Education.

Demchenko, Y., Grosso, P., De Laat, C., & Membrey, P. (2013). Addressing big data issues in scientific data infrastructure. In *2013 International Conference on Collaboration Technologies and Systems (CTS)* (pp. 48–55). San Diego,CA: IEEE. http://doi.org/10.1109/CTS.2013.6567203

Dijkstra, E. W. (1982). *Selected writings on computing: A personal perspective*. New York, NY: Springer-Verlag. http://doi.org/10.1007/978-1-4612-5695-3

Edmondson, A. C., & Mcmanus, S. E. (2007). Methodological Fit in Management Field Research. *Academy of Management Review*, *32*(4), 1155–1179.

Elias, T., Ulenbelt, P., Fokke, M., Bruins Slot, H., & Meenen, P. Van. (2014). *Parlementair onderzoek naar ICT-projecten bij de overheid*. *Nota*. The Hague. Retrieved from https://www.tweedekamer.nl/sites/default/files/field_uploads/33326-5-Eindrapport_tcm181-239826.pdf

Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster Analysis* (5th ed.). London: Wiley. http://doi.org/10.1007/BF00154794

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, *35*(2), 137–144. http://doi.org/10.1016/j.ijinfomgt.2014.10.007

Graham, I. (2007). *Business rules management and service oriented architecture a pattern language* (1st ed.). Hoboken, NJ: John Wiley & Sons.

Hardle, W., & Simar, L. (2007). *Applied multivariate statistical analysis* (2nd ed.). Berlin: Springer-Verlag. http://doi.org/10.1007/978-3-540-72244-1

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

Hogarty, K., Hines, C., Kromrey, J., Ferron, J., & Mumford, K. (2005). The Quality of Factor Solutions in Exploratory Factor Analysis: The Influence of Sample Size, Communality, and Overdetermination. *Educational and Psychological Measurement*, *65*(2), 202–226. http://doi.org/10.1177/0013164404267287

Hohwiller, J., Schlegel, D., Grieser, G., & Hoekstra, Y. (2011). Integration of BPM and BRM. In *Business Process Model and Notation* (Vol. 95, pp. 136–141). http://doi.org/10.1007/978-3-642-25160-3_12

Jeston, J., & Nelis, J. (2014). *Business process management* (3rd ed.). New York, NY: Routledge.

Jolliffe, I. T. (2002). *Principal Component Analysis, Second Edition*. *Springer Series in Statistics* (Vol. 98). New York, NY: Springer Berlin Heidelberg. http://doi.org/10.1007/b98835

Jones, T. (2012). What the Business Process Director Needs to Know About Business Rule Management.

Karlsson, F., Ågerfalk, P. J., & Hjalmarsson, A. (2001). Method configuration with development tracks and generic project types.

Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, *33*(TR/SE-0401), 28. http://doi.org/10.1.1.122.3308

Klesse, M., & Winter, R. (2007). Organizational forms of data warehousing: An explorative analysis. In *Proceedings of the 40th Annual Hawaii International Conference on System Sciences* (p. 11). http://doi.org/10.1109/HICSS.2007.417

Lahrmann, G., & Stroh, F. (2009). Towards a Classification of Information Logistics Scenarios-An Exploratory Analysis. In *2009 42nd Hawaii International Conference on System Sciences* (pp. 1–10). Big Island, HI: IEEE. http://doi.org/10.1109/HICSS.2009.464

Leist, S. (2004). Methoden zur Unternehmensmodellierung-Vergleich, Anwendungen und Diskussionen der Integrationspotenziale. *Habilitation, Institut Für Wirtschaftsinformatik, Universität St. Gallen, St. Gallen*.

Liao, S. H. (2005). Expert system methodologies and applications-a decade review from 1995 to 2004. *Expert Systems with Applications*, *28*(1), 93–103. http://doi.org/10.1016/j.eswa.2004.08.003

MacCallum, R. C., Widaman, K. F., Zhang, S. B., & Hong, S. H. (1999). Sample Size in Factor Analysis. *Psychological Methods*, *4*(1), 84–99. http://doi.org/10.1037/1082-989x.4.1.84

McAfee, A., & Brynjolfsson, E. (2012). Big Data. The management revolution. *Harvard Buiness Review*, *90*(10), 61–68. http://doi.org/10.1007/s12599-013-0249-5

McGrath, J. E. (1984). *Groups: Interaction and performance* (1st ed.). Upper Saddle River, NJ: Prentice-Hall Inc. http://doi.org/10.2307/2393041

Morgan, T. (2002). *Business Rules and Information Systems : Aligning IT with Business Goals*. Boston, MA: Addison-Wesley.

Morreim, E. H. (1991). Gaming the system. Dodging the rules, ruling the dodgers. *Archives of Internal Medicine*, *151*(3), 443–447. http://doi.org/10.1080/15265161.2012.656801

Nelson, M. L., Peterson, J., Rariden, R. L., & Sen, R. (2010). Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry. *Information and Management*, *47*(1), 30–41. http://doi.org/10.1016/j.im.2009.09.007

Nelson, M. L., Rariden, R. L., Sen, R., & Texas, A. (2008). A Lifecycle Approach towards Business Rules Management. In *41st Hawaii International Conference on System Sciences* (pp. 1–10).

Newsted, P. R., Huff, S. L., & Munro, M. C. (1998). Survey Instruments in Information Systems. *MIS Quarterly*, *22*(4), 553. http://doi.org/10.2307/249555

NVZ. (2014). NVZ-voorzitter Van Rooy: Minder maar betere regels. Retrieved July 4, 2016, from https://www.nvz-ziekenhuizen.nl/actueel/1325-nvz-voorzitter-van-rooy-minder-maar-betere-regels

Object Management Group. (2014). *Decision Model and Notation*. Retrieved from http://www.omg.org/spec/DMN/

Object Management Group (OMG). (2011). Business Process Model and Notation (BPMN) Version 2.0. *Business*, *50*(6), 508. http://doi.org/10.1007/s11576-008-0096-z

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society*, *30*(3), 286–297. http://doi.org/10.1109/3468.844354

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12), 1053–1058. http://doi.org/10.1145/361598.361623

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, *2*(11), 559–572.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management of Information Systems*, *24*(3), 45–78.

Ravesteyn, P. (2011). *Factors and competences for business process management systems implementation*. Utrecht: Utrecht University.

Ravesteyn, P., & Jansen, S. (2009). A Situational Implementation Method for Business Process Management Systems. In *Proceedings of the Fifteenth Americas Conference on Information Systems (AMCIS)* (pp. 1–9). San Francisco, CA.

Rieley, J. B. (2000). Are Your Employees Gaming the System? *National Productivity Review*, *19*(3), 1–6.

Rogers, P., & Blenko, M. (2006). Who Has the D? *Harvard Business Review*, *84*(1), 52–61.

Rolland, C., & Prakash, N. (1996). A proposal for context-specific method engineering. In *Method Engineering* (pp. 191–208). Springer.

Rosca, D., & Wild, C. (2002). Towards a flexible deployment of business rules. *Expert Systems with Applications*, *23*(4), 385–394.

Ross, R. G. (2003). *Principles of the business rule approach* (1st ed.). Boston, MA: Addison-Wesley Professional.

Ross, R. G. (2009). RuleSpeak Sentence Forms: Specifying Natural-Language Business Rules in English. *Business Rules Journal*, *10*(4).

Rula, A., Maurino, A., & Batini, C. (2016). *Data and Information Quality: Dimensions, Principles and Techniques* (1st ed.). New York, NY: Springer. http://doi.org/10.1007/978-3-319-24106-7

Simon, H. A. (1997). *The Sciences of the Artificial* (3rd ed., Vol. 33). Cambridge: MIT Press.

Smit, K., & Zoet, M. (2016). Management control system for business rules management. *International Journal on Advances in Systems and Measurements (IARIA)*, *9*(3), 210–219.

Smit, K., Zoet, M., & Berkhout, M. (2017). Functional Requirements for Business Rules Management Systems. *Twenty-Third Americas Conference on Information Systems*, 1–10.

Smit, K., Zoet, M., & Versendaal, J. (2017a). Identifying Challenges In BRM Implementations Regarding The Elicitation, Design And Specification Capabilities At Governmental Institutions. In *Twenty-Fifth European Conference on Information Systems (ECIS)* (p. 14).

Smit, K., Zoet, M., & Versendaal, J. (2017b). Identifying challenges in BRM implementations regarding the verification and validation. In *Twenty First Pacific Asia Conference on Information Systems* (p. 4).

Strong, D., & Volkoff, O. (2010). Understanding Organization-Enterprise System Fit: A Path to

Theorizing the Information Technology Artifact. *MIS Quarterly*, *34*(4), 731–756.

Tarantino, A. (2008). *Governance, risk, and compliance handbook: technology, finance, environmental, and international guidance and best practices*. Hoboken, NJ: John Wiley & Sons.

Taylor, J. (2011). *Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics* (1st ed.). Indianapolis, IN: IBM press.

van de Weerd, I. (2009). *Advancing in software product management: An incremental method engineering approach*. Utrecht: Utrecht University.

van der Aalst, W. M. P. (1998). Three Good reasons for Using a Petri-net-based Workflow Management System. In *Information and Process Integration in Enterprises: Rethinking Documents* (pp. 161–182).

van Slooten, K., & Hodes, B. (1996). Characterizing IS Development Projects. *Proceedings of the IFIP TC8, WG8.1/8.2 Working Conference on Method Engineering, 26-28 August 1996, Atlanta, USA*, (1992), 29–44. http://doi.org/10.1007/978-0-387-35080-6_3

Vanthienen, J. (2001). Ruling the business: about business rules and decision tables. *New Directions in Software Engineering*, 103–120.

Von Halle, B. (2002). *Business Rules Applied — Business Better Systems Using the Business Rules Approach*. New York, NY: John Wiley & Sons.

Von Halle, B., & Goldberg, L. (2009). *The Decision Model: A business logic framework linking business and technology*. New York, NY: Taylor and Francis Group, LLC.

Wagner, W. P., Otto, J., & Chung, Q. B. (2002). Knowledge acquisition for expert systems in accounting and financial problem domains. *Knowledge-Based Systems*, *15*(8), 439–447.

Wand, Y., & Weber, R. (1995). On the deep structure of information systems. *Information Systems Journal*, *5*(3), 203–223. http://doi.org/10.1111/j.1365-2575.1995.tb00108.x

Wang, W., Indulska, M., & Sadiq, S. (2016). To Integrate or Not to Integrate--The Business Rules Question. In *International Conference on Advanced Information Systems Engineering* (pp. 51–66).

Weber, R. (1997). *Ontological Foundations of Information Systems*. Coopers & Lybrand and the Accounting Association of Australia and New Zealand.

Wieringa, R. (2014). Empirical research methods for technology validation: Scaling up to practice. *Journal of Systems and Software*, *95*, 19–31. http://doi.org/10.1016/j.jss.2013.11.1097

Winter, R. (2010). *Unternehmensarchitekturprinzipien*. St. Gallen.

Winter, R. (2011a). Design Solution Analysis for the Construction of Situational Design Methods. In *Engineering Methods in the Service-Oriented Context* (pp. 19–33). Berlin: Springer.

Winter, R. (2011b). Problem analysis for situational artefact construction in information systems. In *Emerging themes in information systems and organization studies* (pp. 97–113). Berlin: Springer.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in software engineering* (1st ed.). Berlin: Springer-Verlag. http://doi.org/10.1007/978-3-642-29044-2

Wolfson, M., Madjd-Sadjada, Z., & James, P. (2004). Identifying National Types: A Cluster Analysis of Politics, Economics, and Conflict. *Journal of Peace Research*, *41*(5), 607–623.

Zoet, M. (2014). *Methods and Concepts for Business Rules Management*. Utrecht: Hogeschool Utrecht.

Zoet, M. (2015). Decision Management: The Choice Between Man and Machine. Retrieved January 17, 2017, from https://www.linkedin.com/pulse/decision-management-de-keuze-tussen-mens-en-machine-martijn-zoet?trk=mp-reader-card

Zoet, M., Smit, K., & de Haan, E. (2016). Performance Indicators for Business Rule Management. In *eKNOW 2016 : The Eighth International Conference on Information, Process, and Knowledge Management Performance* (pp. 1–6). Venice.

Zoet, M., Smit, K., & Leewis, S. (2015). A Classification of Modification Categories for Business Rules. In *Proceedings of the 28th Bled eConference* (pp. 155–168). Bled.

Zoet, M., & Versendaal, J. (2013). Business Rules Management Solutions Problem Space: Situational Factors. In *Pacific Asia Conference on Information Systems 2013 (PACIS)* (p. 247).

7 Appendences

Appendix A – Paper A
A Framework for the Organizational Implementation of Business Rules Management Solutions – Conference on the Practice of Enterprise Modelling (PoEM2017)

A Framework for the Organizational Implementation of Business Rules Management Solutions

Sam Leewis *s.leewis@students.uu.nl*
Utrecht University Graduate School of Natural Sciences,
Princetonplein 5, 3584 CC Utrecht, The Netherlands

Koen Smit *koen.smit@hu.nl*
Digital Smart Services HU University of Applied Sciences Utrecht,
Utrecht, The Netherlands

Martijn Zoet *martijn.zoet@zuyd.nl*
Optimizing Knowledge-Intensive Business Processes Zuyd University of Applied Sciences,
Sittard, The Netherlands

Marcela Ruiz *m.ruiz@uu.nl*
Utrecht University Graduate School of Natural Sciences,
Princetonplein 5, 3584 CC Utrecht, The Netherlands

Sjaak Brinkkemper *s.brinkkemper@uu.nl*
Utrecht University Graduate School of Natural Sciences,
Princetonplein 5, 3584 CC Utrecht, The Netherlands

Abstract

The implementation of software products is a time-consuming activity and needs a high level of expertise to be completed successfully. This is especially the case for software products related to an immature field, such as Business Rules Management (BRM) and Business Rules Management Solutions (BRMS). Support is essential to successfully guide the organizational implementation of a BRMS. Motivated by the diversity of organizational structures and their BRMS implementation contexts, we present the development of a situational-oriented BRMS implementation framework. We adopted the situational artifact construction technique to ensure that the framework can be applied in different situations. For the construction of the BRMS implementation framework, this study utilizes data of 13 BRMS implementation cases distributed over the financial and public sectors in the Netherlands. The BRMS implementation framework is a stepping stone towards further research on situational implementation methodology in the BRM domain.
Keywords: Business Rule Management, Business Rule Management Solution, Situational artifact construction, Method Engineering, Multi-case study

1. Introduction

An increasing amount of laws and regulations and the demand for automation raises the need for handling business rules in a proper way [6, 10]. For practice to be able to do so, laws and regulations need to be transformed to products and services, which could be used to create added value. Business rules are used to ensure business structure and could be applied to people, processes, organization behavior, computer systems, and to support organizations to reach their goals [12]. Thereby, giving the business rules domain an important and valuable task in practice. Business Rules Management (BRM) is the practice of managing business rules centered around the activities of elicitation, design, specification, verification, validation, deployment, execution, governance, or the monitoring of business rules [2]. A Business Rules Management Solution (BRMS) enables organizations to elicitate, design, manage and execute business rules and is a co-creation of the BRM activities described earlier [3]. Making use of business rules and the increasing amount of business rules, a BRMS is needed to keep a clear overview of all the elicited, designed, and specified business rules required in an organization [20, 25, 26, 33].

In the field of information systems, the domain of BRM is a relatively young subject of study and gained the interest from researchers the past several years [32]. The scientific world sees many opportunities in BRM, still the BRM topic is not over-researched. This is especially the case for the technical implementation of a business rules management solution. In addition, there is a lack of research in the arena of organizational implementation of BRMS [19]. Organizational implementation of BRMS is focused on the organizational aspects touched when implementing a software product. The actual paper focuses on the latest one motivated by the significant difference in scientific contributions for organizational implementations of BRMS compared to technical implementations [19].

A BRMS contains nine *capabilities* (see Figure 1, concept BRMS CAPABILITY), which an organization can configure for their own purposes to create, implement, and manage business rules [25, 32, 33]. The nine capabilities of a BRMS are as followed: 1) ELICITATION, 2) DESIGN, 3) SPECIFICATION, 4) VERIFICATION, 5) VALIDATION, 6) DEPLOYMENT, 7) EXECUTION, 8) MONITORING, and 9) GOVERNANCE [25, 33]. Previously conducted research has shown that different BRMSs have a common DESIGN PROBLEM [2, 4, 8, 15–17]. A common DESIGN PROBLEM is the difference between the goal state and the current state of a system, in this case, a BRMS, and is an indication that common PROBLEM CLASS, for which DESIGN SOLUTION can be created, exists [31]. Winter, [31] depicts a problem class as a set of comparable design problems. A PROBLEM SPACE is a collection of multiple PROBLEM CLASS. An instantiation of a PROBLEM CLASS in a specific organization is defined as a DESIGN SOLUTION. Specific for the BRMS problem space, the design solution is a specific configuration of the earlier mentioned nine capabilities [25, 33].

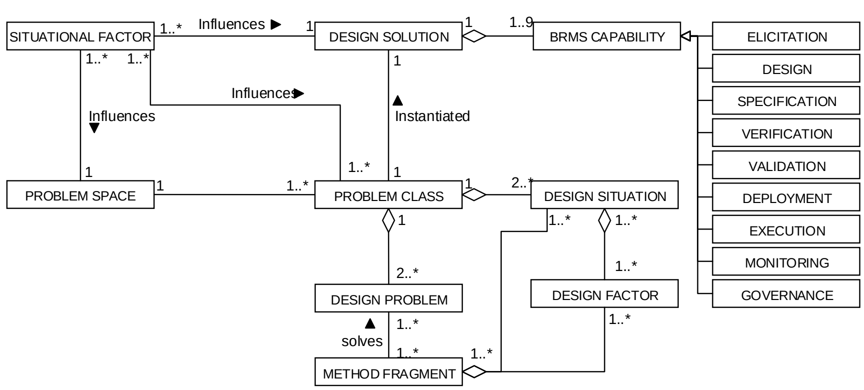


Fig. 1 Excerpt of the BRMS metamodel

SITUATIONAL FACTORS describe the context in which an information system artifact or organization has to operate in a way that the deployed artifact fits the context of the environment (see Figure 1). Situational factors might be elicited directly from the specific context in which a BRMS can be potentially implemented. Research identifying these situational factors is conducted in the situational method engineering research field [7, 14, 22, 24], with specific applications in software product management [5, 28] and business process management [8, 21].

In this research, we adopt the situational artifact strategy to design a framework for the organizational implementation of BRMS. Our framework is aiming at supporting the requirements and needs that each organization demands for the implementation of business rules [6, 10, 12]. The framework involves four main artifacts: a) the elicitation survey to extract the main inputs of a BRMS (design problems and problem classes), the elicitation survey is general enough to be applied to different contexts; b) the situational artifact construction process, to identify design situation and design factors that are related to their corresponding design problems; c) the BRMS metamodel with the specification of the concepts that should be instantiated when implementing a BRMS; and d) the BRMS metamodel support tool that implements the metamodel, it facilitates industrial transference and management of BRMS.

As a proof of concept, we have applied the BRMS framework to 13 BRMS implementation cases that belong to the financial and the public sector in the Netherlands (the BRMS implementation cases are characterized in Table 1). As a result, we have validated the feasibility of our framework by analyzing the elicited data, specifying design situations and design factors, and instantiating the BRMS metamodel. As main contributions for this paper, we describe a BRMS implementation framework and the application of the framework to 13 BRMS implementation cases.

The first section of this paper discusses the BRMS in the context of the related work, in the second section we describe the BRMS implementation framework. The third section focuses on the situational artifact construction process, and the final section contains the conclusion, lessons learned, and future work.

2. Related work

A BRMS is a set of software components for the elicitation, design, management and execution of business rules and is a composition of nine capabilities [25, 33]:

1) The *elicitation* capability captures knowledge from various sources. When there already is a business rules architecture in place, the elicitation capability also performs an impact analysis to review which artifacts are hit when a certain change occurs.
2) The *design* capability creates the non-platform specific rule system. The output of the design capability is the business rule architecture.
3) The goal of the *specification* capability is to create the business rules and fact types needed to constrain or define specific aspects of the business.
4) The *verification* capability verifies the created business rule architecture to checks for possible semantic and syntax errors.
5) The *validation* capability reviews the created value proposition. The goal of this capability is to check for possible errors in its expected behavior.
6) The *deployment* capability transforms the verified and validated value proposition to implementation-dependent executable business rules. This is not necessarily performed by a system; a subject-matter expert could also perform the transformation.
7) The *execution* capability delivers the actual value proposition. To realize the added value, human or information system actors execute the business rules.
8) The *monitoring* capability observes, evaluate and keeps a record of the full range of activities in the previously explained BRM capabilities that are conducted to realize the value proposition.
9) The governance capability consists of three sub-capabilities; version management, traceability, and validity management. The role of the version management capability is to capture and keep track of elements which are created or modified in the other eight capabilities. The goal of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time. The goal of the traceability capability is to ensure the possibility to trace created elements to their corresponding laws and regulations.

At first, a BRMS was stated as singular problem-oriented, meaning that it is designed to solve one specific problem [18, 27]. Previous research contradicts the singular problem orientation and proposes that different BRMS have a common design problem. Common design problems indicate that different problem classes exist [31]. Therefore, design solutions can be created. A problem space can contain a single or multiple problem classes. An instantiation of a specific problem class in a specific organization is defined as a design solution [30, 32]. The design solution in the BRM problem space is the configuration of the earlier mentioned nine capabilities. [25, 32]. The problem space and design solutions are influenced by situational factors as described by Winter [31]. The context in which the organization or artifact has to operate is described by the situational factors. Being that each organization has different characterizations and therefore the implementation is different compared to other implementations. The framework needs to have the possibility to be adapted to different situations, a so-called situational artifact.

Winter [31] proposed in his work a technique to create situational artifacts through a design science multicase study approach. The goal of this research is to create a BRMS implementation framework which could be used for implementing BRMS from an organizational perspective. In other words, a situational artifact is created using the situational artifact construction technique [1]. The situational artifact construction technique focusses on the following eleven steps: 1) Initial demarcation of the design problem class, 2) Identification of potential contingency factors, 3) Field study based analysis of design problems in practice, 4) Refining specifications of the design problem class, 5) Calculation of ultra-metric distances, 6) Determination of the level of generality, 7) Specification of design situations, 8) Identifying characterizing design factors, 9) Linking design factors to design problems, 10) Deriving elementary problem-solving actions, and 11) Deriving method configuration rules. In short, this technique discovers, identifies, and creates elements to design a situational artifact in any Information Science research field.

3. The BRMS implementation framework

The aim of the BRMS framework is to support the organizational implementation of business rules management solutions. This generic version (as shown in Figure 2) of the BRMS implementation framework focuses on the presentation of the elements of the framework and the relationships among the different elements. The BRMS implementation framework provides guidelines of how to implement a BRMS in a specific situation. Besides the BRMS implementation framework as a deliverable, the framework itself contains four major contributions. These contributions are the elicitation survey, The situational artifact construction process, the metamodel, and the BRMS metamodel support tool. Thereby, supporting any future research on situational artifacts in the BRM domain. The framework enables the possibility to support each implementation of a BRMS in each instantiation of a problem class. Therefore, the traceability of the content of the framework is of high importance. An example of this traceability is that the business rules management situational factor contain the source of origin of the situational factor.

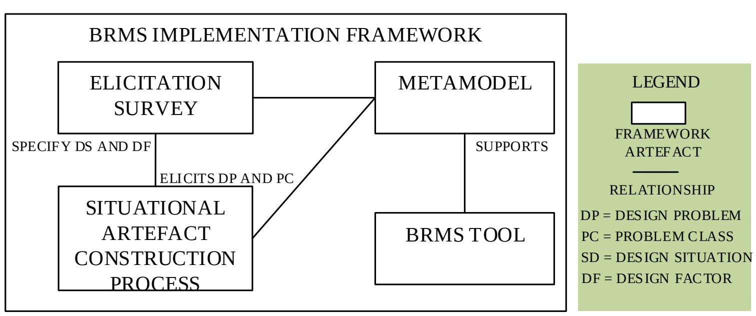


Fig. 2 BRMS implementation framework

The framework and its elements are described briefly, and the following sections will elaborate further on how these elements are actually discovered and created. This process is supported by the situational artifact construction technique of Winter [30].

4. Situational artifact construction process

The goal of situational artifact construction is to construct an artifact which could be adapted to multiple situations. In this case, supporting the implementation of a BRMS. The eleven situational artifact construction steps adopted from the work of Winter [30] are merged with new created additional steps which focus specifically on situational artifact construction in the BRMS domain. The adaptation is needed because the focus of the initial situational artifact construction is on a generic level In the BRM research field a more qualitative approach is needed because the field lacks standards as a base to perform quantitative analysis from, which is originally part of the initial situational artifact construction technique. The initial situational artifact construction steps and the additional steps are shown in Figure 3. The initial process is focused on creating a situational artifact in any given research field. The additional steps are deliverables which are needed to build a situational artifact in the field of BRM.

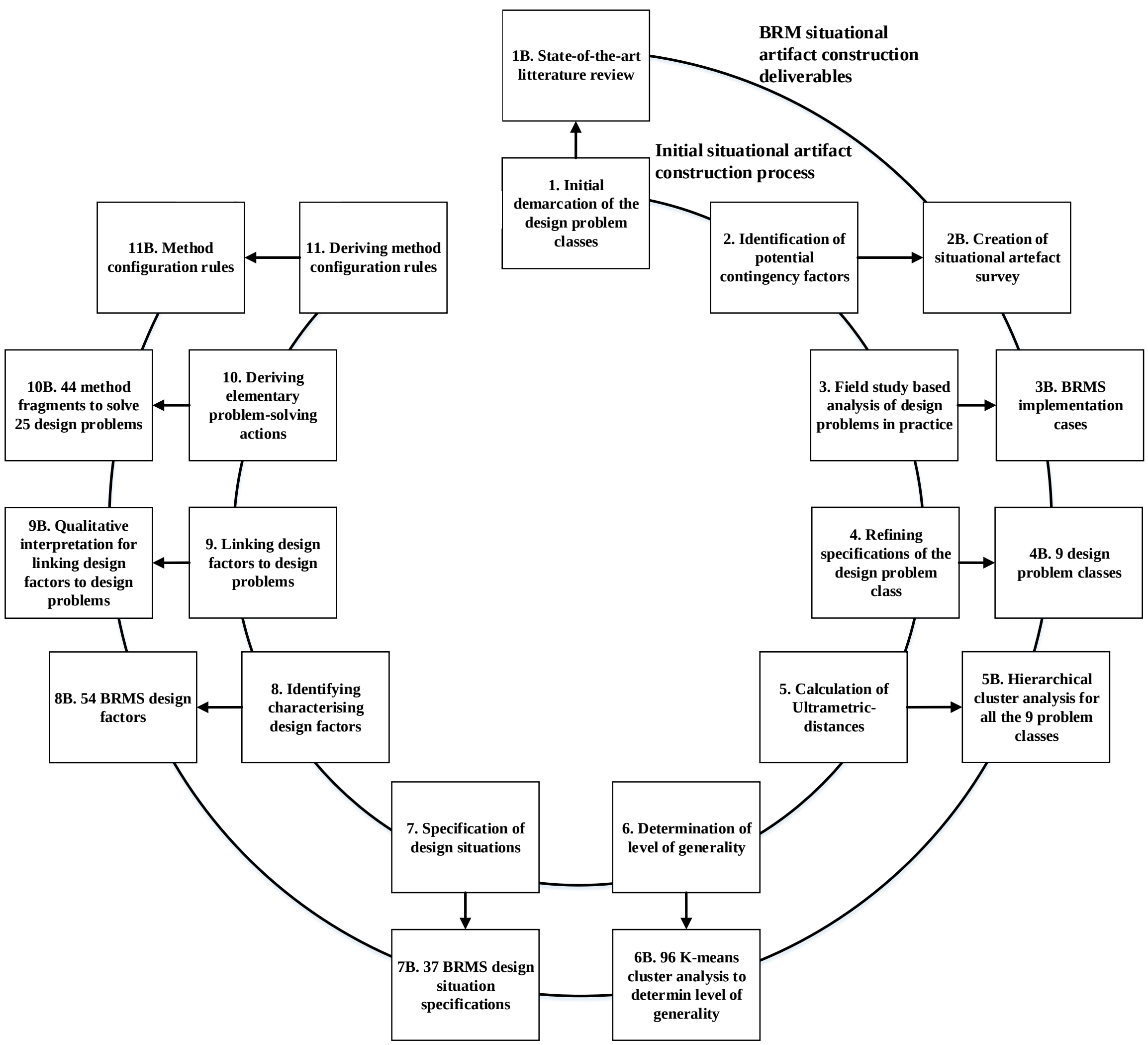


Fig. 3 Situational artifact construction process

The eleven situational artifact construction steps are elaborated further with data used from a running experiment.

4.1 Initial demarcation of the design problem class

Discovery of the BRM problem space is needed to identify the existing knowledge on creating a situational artifact in the field of BRM. The goal of this step is to discover any concepts in the field of BRM (or its neighboring fields) to support the creation of a situational artifact. A state-of-the-art literature review is conducted to create an overview of the existing body of knowledge and to discover similar research conducted in related research fields (business process management, software product management, and enterprise architecture).

4.2 Identification of potential contingency factors

The state-of-the-art literature review is also used for the identification of potential contingency factors. This literature review resulted in the creation of the survey which purpose is to gather BRMS implementation cases. Additional interviews are conducted with members of the BRM community to validate the discovered problem classes, and the contingency factors included in the situational artifact survey. The questions of the survey are constructed with a combination of existing literature and by use of experts interviews in the BRM community. The experts are chosen on their knowledge on BRMS capabilities and the different instantiations of the BRMS capabilities. The group of experts exist of a professor lecturing and conducting research in the field of BRM, a Ph.D. student lecturing and conducting research on BRM, and a graduate student with research experience in the field of BRM.

4.3 Field study based analysis of design problems in practice

The goal of this field study is to collect data on different BRMS implementations distributed over different sectors. These collections of different implementations can create an overview of clusters of situational factors that influence the different problem classes in the business rules management solution problem space. Thereby, creating a situational artifact for each different sector and possibly for more instantiations in these different sectors. Certain questions are specific for each section and can only occur when dealing with these elements, but there are questions which are more general. Therefore, these questions are frequently used in different sections. There are possibilities that sections only exist of the general questions, the reason of this is that a certain level of abstraction is selected. Therefore, no relevant questions exist to complement the more general questions. Furthermore, the questions that are created by the practical experience of the BRM expert or by use of existing literature. The situational artifact survey intends to reach people who have experience with a BRMS implementation. The respondents are distributed over a wide range of sectors, a wide range of size, and focus of implementation. Furthermore, the possibility exists that someone has experience with multiple BRMS implementation. Therefore, these respondents could fill in the questionnaire multiple times. Reducing the large amount of contingency factors into a relevant set of design factors, a principal component analysis is performed. The characterization of the 13 BRMS implementation cases are shown in Table 1.

Table 1 BRMS implementation cases

Case ID:	Employees:	Sector:	Implementation focus:
1	2001 - 5000	Public	Organisation wide
2	2001 - 5000	Public	Organisation wide
3	>5000	Public	Application focused
4	2001 - 5000	Public	Organisation wide
5	2001 - 5000	Financial	Application focused
6	2001 - 5000	Financial	Line of business focused
7	501 - 1000	Financial	Line of business focused
8	251 - 500	Financial	Line of business focused
9	>5000	Financial	Organisation wide
10	251 - 500	Public	Application focused
11	>5000	Public	Line of business focused
12	501 - 1000	Financial	Organisation wide
13	2001 - 5000	Public	Organisation wide

The survey results were submitted to a qualitative approach of a Principal Component Analysis (PCA). The main goal of a PCA is reducing a list of potential contingency factors to relevant design factors [1, 13]. The validation pilot test is used as a qualitative method for reducing the list of potential contingency factor candidates to a smaller set of relevant 'design factors.' Thereby, the existing questionnaire was already a relevant list of design factors. The survey results will also be used to refine the specifications of the design problem classes, which is shown in the next section.

4.4 Refining specifications of the design problem class

The earlier qualitative approach focusses on creating a relevant list of design factors. This step focusses on specifying and refining the design factors into more specific and refined design problem classes. The design problem classes identified in earlier steps should be refined more to ensure define the degree of homogeneity of the solutions. This will result in the excluding of ''outlier'' design solutions and thereby ensuring the degree of homogeneity. When dealing with a low number of BRMS implementation cases, the possibility exists that no 'outliers' could be identified, this results in that number of design solutions stay the same. The problem classes identified are: 1) Elicitation problem class, 2) Design problem class, 3) Specification problem class, 4) Verification problem class, 5) Validation problem class, 6) Deployment problem class, 7) Execution problem class, 8) Monitoring problem class, and 9) Governance problem class. The next step will cluster the solutions into relevant clusters which are representative towards all the gathered BRMS implementation cases.

4.5 Calculation of ultra-metric distances

Problem classes can be divided into a number of generic design situations depending on the degree of generality. The generic design situations are the specified solutions depending on the number of clusters selected in the problem class. Based on the Euclidian distance, the similarity (or dissimilarity) of two design solutions within a problem class can be represented by an ultrametric-distance. The cases and their distances are visualized using a tree-like graph. Ultrametric distances can be visualized by a graph whose Y-axis represents generality and whose X-axis represents the set of observed cases. The similarity or dissimilarity of two design solutions (or more) corresponds to the generality level of their relation. If the similarity is high, their relation is represented on a lower level of generality. Therefore, when there is a low level of similarity, their relation is represented on a higher level of generality [30]. Figure 4 shows the ultra-metric distance of the elicitation problem class. This graph contains 13 solutions (elicitation 1...13) for the elicitation problem class. Observed case 2 is represented by its own specific solution, Elicitation solution 2. This level of generality is the same for the other observed cases on this level, which are represented by their own specific solution. Observed case 1, 2, 5, 6, 11 and, 13 are represented by a more generic solution, Elicitation solution 1, 2, 5, 6, 11 and, 13. The observed cases 1 - 6, 9, and 11 - 13 are represented by an even more generic solution, Elicitation solution 1 - 6, 9, and 11 - 13. At the top level, the generic solution contains all the observed cases and is the most generic representation towards all the observed cases.

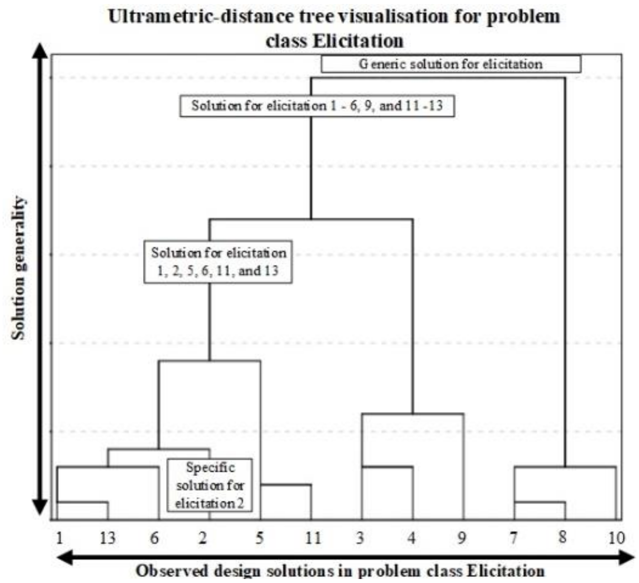


Fig. 4 Ultrametric-distance visualization

Specifying what the actual optimal number of clusters for a design solution is needed. This step will focus on what level of generality is needed to have optimal cluster consistency in a design solution.

4.6 Determination of level of generality

In order to not only visualize the generic solutions, but also characterize the generic solutions, k-means cluster analyses are applied to the implementation cases. This is to determine the optimal number of clusters for the design solution. Several k-mean cluster analyses were conducted from a minimum of 1 to a maximum of 13 clusters. More clusters is not possible because the maximum is created by the amount of respondents, which is 13. These cluster analyses were conducted for each problem class because the possibility exists that a solution contains only one capability (problem class). Therefore, a total of 96 k-means cluster analyses were conducted to determine the optimal number of clusters for each problem class. For each solution the error total, which is calculated from the distances of all implementation cases to the center of their clusters, the so-called elbow criterion is used [9, 11, 31]. The elbow criterion indicates which increased number of clusters leads to an above-average improvement in the error sum. The error sum is plotted on the y-axis, and the number of clusters is plotted on the y-axis, an elbow arises for the 4-cluster, 7-cluster and 10-cluster solution as shown in Figure 5.

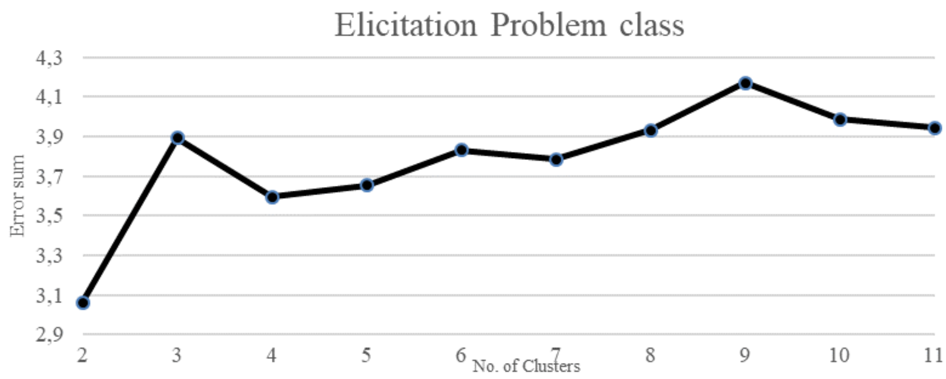


Fig. 5 Elicitation problem class elbow criterion

4.7 Specification of design situations

The design situations need more than only formal definition, which is done by the ultrametric-distance calculation, but also need semantic interpretation (i.e. by specifying the design problem types). The ultrametric-distances are used, for semantically specifying the design situations. The problem classes are specified into their preferred cluster consistency. The Elicitation problem class is specified into a 4-cluster solution, and these four different clusters are specified using a mean comparison analysis (ANOVA). This analysis aims at specifying the design situations. The goal of this is to create different design situations. Thereby, aiming at what is the exact reason why these design solutions are created into a cluster. Elements of all the design situations in a problem class, which were identical, were excluded from the design situation specification. Specifying design situations aims at showing factors which differentiate a design situation from another design situation, this will not help when identical elements do not differentiate a design situation from another design situation. These factors need to be characterized further.

4.8 Identifying characterizing design factors

Each design situation consists of characterizing design factors and need to be specified further. Every design factor has different values, these values influence the characterization of the design factor and thereby the design situation. An example of this is the design factor: Capability leader, this design factor characterizes which department is in the lead of the specific capability. The value of this capability could be IT, the business, or the central IT/Business group. The characterizing design factors are defined together with their values and related problem classes. The characterizing design factors define the design situations, but there are also design factors which do not characterize design situations but are still important for solving design problems. The non-characterizing design factors are not specifically solving design problems but are directly related to the problem classes, which in their turn solve certain design problems. The reason that these design factors are non-characterizing is that for all the design situations in a problem class these design factors have the same value.

4.9 Linking design factors to design problems

The characterized design factors described in the earlier sections need to be linked to the 25 proposed design problems, which are known in literature and out of the survey results as reasons to implement a BRMS. All the earlier conducted steps analyze the existing design solutions. These design solutions are cases of successful BRMS implementations. These design solutions were created with a certain purpose, and

therefore the design factors can be qualitatively interpreted and linked to the known design problems. The 25 known design problems were mapped against the identified and characterized design factors.
An example is design problem ''low productivity of elicitation'' this could be positively impacted by letting the system take over some tasks (capability autonomy), using data as a source when elicitating, the output of elicitation is a relevant set of selected sources, which could be reused, performing a scenario analysis which is based on business scenario's, and defining roles (the input role). In short, a certain combination of characterized design factors could solve a certain design problem

4.10 Deriving elementary problem-solving actions

The ideal possible next step would be deriving elementary problem-solving actions by comparing design solutions with design problems. Out of these elementary problem-solving actions, method fragments are created. Winter [30] focuses on a change in maturity (in different research fields), as-is to a to-be situation, which could be supported by method fragments. The focus in this study is different compared to that of Winter's work, the focus of this research lies in supporting the implementation of a BRMS. Therefore, there is a phase were is no BRMS and a phase where there is a BRMS. This is the same for each observed case, and the road to implementing a BRMS is different for each case. When implementing a BRMS in a specific organization, there is no wrong selection of design factors, only the best fit for a specific organizations situation. Nonetheless, it is still possible to create method fragments to support solving design problems when implementing a BRMS. A problem related to the elicitation problem class is given down below. Design problem #1 low productivity of elicitation, is proposed to be solved with a certain configuration of characterizing design factors. Design problem #1 could be solved with a combination of the following design factors: #5 Capability autonomy, #6 Elicitation source (Data), #7 Relevant set of selected sources, #8 Scenario analysis, and #14 Role: Input. Design factor #5, #6, #7, #8 and #14 result into method fragment #2, #3, #4, #5 and #12. The combination of these design factors and their created method fragments will evolve into a method which could solve design problem #1, this example is shown down below in Figure 6.

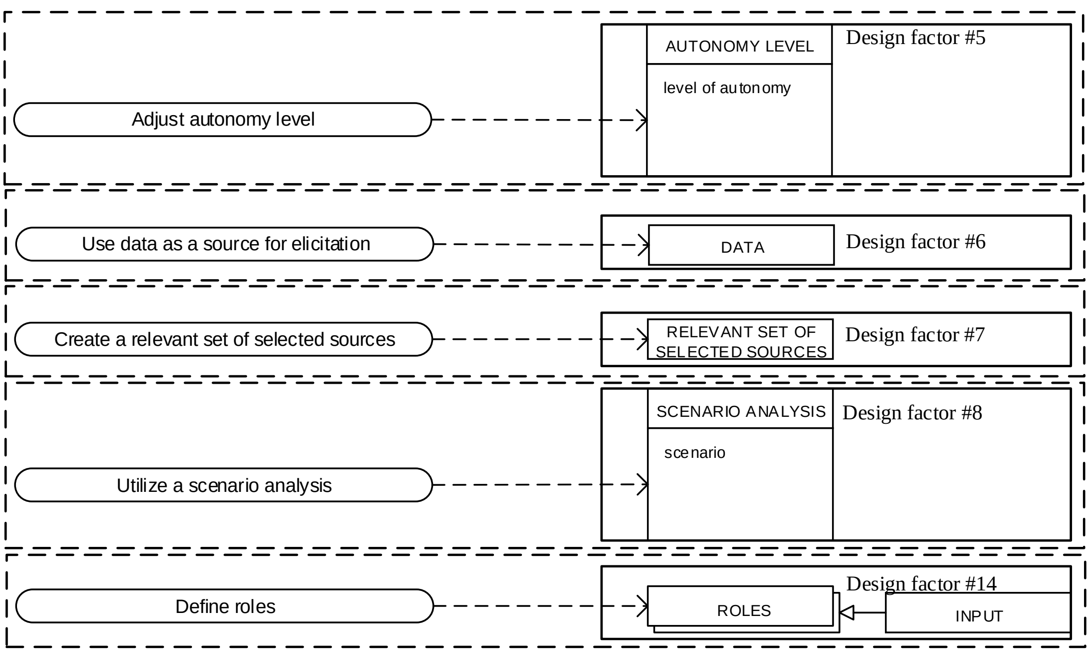


Figure 6 Created metamodel elements

4.11 Deriving method configuration rules

Based on the set of identified design problems and specified method fragments, now method configuration rules need to be derived. Basically the (reusable) method fragments identified in the previous section need to be related to their respective design situations. Configuration rules need to be designed, which guides the configuration of solutions to solve specific design problems. Merging the method fragments into one super method is not sufficient for solving the design problems. A certain combination of design problems and design factors requires additional information and attention. Therefore, characterizing design factors related to a certain problem class automatically means that a whole problem class (capability) is implemented. It is not possible to only implement certain parts of a problem class. Therefore, the whole problem class is implemented together with the related non-characterizing design factors and the related characterizing design factors. Therefore, the following notion is created:

When a method fragment is selected to solve a problem, the whole problem class related to that method fragment must be implemented.

- This includes characterizing design factors
- This includes non-characterizing design factors

Certain design situations indicate that BRMS implementation cases exist which deliberately did not had a design factor implemented even if this design factor should solve a specific design problem. The configuration of a design situation depends on design factors which identify the type of implementation. The design factors identifying the types of implementations are focused on the sector in which the organization operates, the number of employees working at this organization, and the focus of the implementation. A specific value of these three design factors gives an indication which design situation fits the organization needs. Finalizing the method configuration rules results into a situational artifact which could be specified and adapted to different situations.

5. Conclusion and lessons learned

The actual paper describes a BRMS implementation framework that involves four main artifacts: a) elicitation survey; b) situational artifact construction process; c) BRMS metamodel; and d) the BRMS metamodel support tool. We introduce each artifact and describe in detail the situational artifact construction process. We have applied the BRMS implementation framework to 13 BRMS implementation cases in the Netherlands. We have applied the survey and followed the situational artifact construction process. As a result, 37 design situations distributed over the nine capabilities of a BRMS (Design, Specification, Verification, Validation, Deployment, Execution, and Governance) have been elicitated. These nine capabilities contain 54 design factors, 41 characterizing design factors and 13 non-characterizing design factors. The characterizing design factors ensure with a different configuration for the characterization of the design situations. The non-characterizing design factors are one specific value and should always be added to the implementation. 25 design problems were identified from the goals and arguments to implement a BRMS. The 25 design problems are linked to design factors which have as a purpose to solve these design problems. The initial quantitative PCA is replaced with a qualitative approach. This qualitative approach focusses on extra validation from practice compared to the initial PCA, which is not validated with people from practice. The possibility exists that important design factors are excluded. Therefore, this additional validation was deemed important for its purpose is to not let a statistical analysis create a relevant set of design factors,

By using the metamodel, we have guided the selection of specific sets of design factors to solve design problems. The concepts of the metamodel indicate how to the different design factors for a certain organization could be supported in terms of a BRMS. The metamodel recommendations for the implementation of the BRMS involves contextual information like size and sector. We have followed a situational artifact construction strategy to design the BRMS implementation framework [30]. The 13 cases that have been analyzed by using the framework (elicitation survey, situational artifact construction process, and BRMS metamodel) demonstrate the feasibility of our research.

To facilitate the industrial transference of the current framework, we are working in the development of the BRMS metamodel support tool support. The main objective is to implement the metamodel that facilitates the specification of the main outputs from the survey (design problems and problem classes) and the situational artifact construction process (design situations and design factors). In addition, the most important functionality of the tool is to guide the selection of the most convenient design problems, which describes how the BRMS should be implemented. To build the BRMS, we follow a model-driven development paradigm were the metamodel is instantiated in a modeling editor supported with templates. Future research is needed to conduct analysis with a larger number of BRMS implementation cases and thereby further increasing the generalizability of the framework and its elements. The focus of the current framework lies now on organizations in the public and financial sector, we plan to include cases from other industries as well. In this direction, we plan to report a case study describing the use of the framework in different context. In addition, we plan to conduct an experiment to validate the usability and sensitivity of the BRMS metamodel support tool.

References

1. Abdi, H., Williams, L.J.: Principal component analysis. Wiley Interdiscip. Rev. Comput. Stat. 2 (4), 433–459 (2010)
2. Aier, S., Riege, C., Winter, R.: Classification of Enterprise Architecture Scenarios – An Exploratory Analysis. Enterp. Model. Inf. Syst. Archit. 3 (1), 14–23 (2008)
3. Arnott, D., Pervan, G.: A critical analysis of Decision Support Systems research. J. Inf. Technol. 20 (2), 67–87 (2005)
4. Baumöl, U.: Strategic agility through situational method construction. In: Proceedings of the European Academy of Management Annual Conference. (2005)
5. Bekkers, W., van de Weerd, I., Brinkkemper, S., Mahieu, A.: The Influence of Situational Factors in Software Product Management: An Empirical Study. In: 2008 Second International Workshop on Software Product Management. pp. 43–50. (2008)
6. Boyer, J., Mili, H.: Agile business rule development. In: Agile Business Rule Development. pp. 49–71. Springer, Berlin, Heidelberg (2011)
7. Brinkkemper, S.: Method engineering: Engineering of information systems development methods and tools. Inf. Softw. Technol. 38 (4), 275–280 (1996)
8. Bucher, T., Winter, R.: Taxonomy of business process management approaches. In: Handbook on Business Process Management. pp. 93–114. Springer, New York (2010)
9. Everitt, B.S., Landau, S., Leese, M., Stahl, D.: Cluster Analysis. Arnold, London (2011)
10. Graham, I.: Business rules management and service oriented architecture a pattern language. John Wiley & Sons, New Jersey (2007)
11. Hardle, W., Simar, L.: Applied Multivariate Statistical Analysis. Springer, Berlin (2003)
12. Hay, D., Healy, K.A., Hall, J.: Defining business rules: What are they really. Final Rep. 34 (2000)
13. Jolliffe, I.T.: Principal Component Analysis, Second Edition. Springer Berlin Heidelberg, New York (2002)
14. Karlsson, F., Ågerfalk, P.J., Hjalmarsson, A., others: Method configuration with development tracks and generic project types. (2001)

15. Klesse, M., Winter, R.: Organizational forms of data warehousing: An explorative analysis. In: Proceedings of the 40th Annual Hawaii International Conference on System Sciences. p. 11. (2007)
16. Lahrmann, G., Stroh, F.: Towards a Classification of Information Logistics Scenarios-An Exploratory Analysis. In: System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on. pp. 1–10. (2009)
17. Leist, S.: Methoden zur Unternehmensmodellierung-Vergleich, Anwendungen und Diskussionen der Integrationspotenziale. Habilit. Inst. für Wirtschaftsinformatik, Univ. St. Gall. St. Gall. (2004)
18. Liao, S.H.: Expert system methodologies and applications-a decade review from 1995 to 2004. Expert Syst. Appl. 28 (1), 93–103 (2005)
19. Nelson, M.L., Peterson, J., Rariden, R.L., Sen, R.: Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry. Inf. Manag. 47 (1), 30–41 (2010)
20. Nelson, M.L., Rariden, R.L., Sen, R., Texas, A.: A Lifecycle Approach towards Business Rules Management. In: 41st Hawaii International Conference on System Sciences. pp. 1–10. (2008)
21. Ravesteyn, P., Jansen, S.: A Situational Implementation Method for Business Process Management Systems. AMCIS 2009 Proc. 9 (1), 1–9 (2009)
22. Rolland, C., Prakash, N.: A proposal for context-specific method engineering. In: Method Engineering. pp. 191–208. Springer (1996)
23. Rosca, D., Wild, C.: Towards a flexible deployment of business rules. Expert Syst. Appl. 23 (4), 385–394 (2002)
24. van Slooten, K., Hodes, B.: Characterizing IS Development Projects. Proc. IFIP TC8, WG8.1/8.2 Work. Conf. Method Eng. 26-28 August 1996, Atlanta, USA. (1992), 29–44 (1996)
25. Smit, K., Zoet, M.: Management Control System for Business Rules Management. 10 (2017)
26. Taylor, J.: Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics. (2011)
27. Wagner, W.P., Otto, J., Chung, Q.B.: Knowledge acquisition for expert systems in accounting and financial problem domains. Knowledge-Based Syst. 15 (8), 439–447 (2002)
28. Van de Weerd, I.: Advancing in software product management: An incremental method engineering approach. Utrecht University (2009)
29. Van den Weerd, I., Brinkkemper, S.: Meta-Modeling for Situational Analysis and Design Methods. Handb. Res. Mod. Syst. Anal. Des. Technol. Appl. 35 (2008)
30. Winter, R.: Design Solution Analysis for the Construction of Situational Design Methods. In: Engineering Methods in the Service-Oriented Context. pp. 19–33. Springer, Berlin (2011)
31. Winter, R.: Problem analysis for situational artifact construction in information systems. In: Emerging themes in information systems and organization studies. pp. 97–113. Springer, Berlin (2011)
32. Zoet, M.: Methods and Concepts for Business Rules Management. Hogeschool Utrecht, Utrecht (2014)
33. Zoet, M., Versendaal, J.: Business Rules Management Solutions Problem Space: Situational Factors. In: PACIS. p. 247. (20

Appendix B - BRMS implementation questionnaire

This is the English translation of the BRMS implementation questionnaire. This questionnaire also is translated into Dutch and Spanish.

This questionnaire will focus on your experience of a business rules management solution. This questionnaire is to be completed for each deployment you have done. Filling out this questionnaire will take about 30 minutes of your time. The information you provide will only be used for this research. The results of the questionnaire will be treated as confidential data.

What is the name of the organisation where the BRMS is implemented?
Anonymous
Other...

In which sector is the BRMS implemented?
Public
Financial
Other...

How many employees work in the organisation where the implementation is carried out?
< 50
50 - 100
101 - 250
251 - 500
501 - 1000
1001 - 2000
2001 - 5000
> 5000

What is the scope of the BRMS implementation?
Application focused
Line of business focused
Organisation-wide

Definitions

In this survey the following definition of a business rule is used:
''a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business''

Elicitation
The elicitation capability is a bifold function. The first function is to determine the knowledge which realises the value proposition of the business rules. This knowledge needs to be captured from various sources including but not limited to laws and regulations. The second function is the to initiate an impact analysis, this is only done when a business rule architecture is already in place.

Design
The output of the design capability is the business rule architecture. The business rule architecture contains a combination of derivation structures and context designs.

Specification
The specification capability specifies the content of each separate context design. The function of this capability is to create the business rules and fact types needed to constrain or define particular aspects of the business.

Verification
The verification capability verifies the created business rule architecture to check for semantic and syntax errors.

Validation
The validation capability reviews the created value proposition. The goal of this capability is to check for possible errors in its expected behaviour.

Deployment
The deployment capability transforms the verified and validated value proposition to implementation-dependent executable business rules. The actor that utilises the value proposition is not necessarily a system, a subject-matter expert could also utilise this.

Execution
The output of the deployment capability is then executed in the execution capability, which delivers the actual value proposition. To realise the added value, human or information system actors execute the business rules.

Monitoring
The monitoring capability observes, checks and keeps a record of not only the execution of the value proposition but also the full range of activities in the previously explained BRM capabilities that are conducted to realise the value proposition.

Governance
The governance capability consists of three sub-capabilities; version management, traceability, and validity management.

- **Version management**

The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities.

- **Validity management**

The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time.

- **Traceability**

The purpose of the traceability capability is to ensure the possibility to trace created elements to their corresponding laws and regulations.

Characterization of Business Rules Management
This section will characterize business rule management in the organisation where the implementation is carried out or is still carried out. Statements are presented and true of false should be answered on the basis of how the organisation look at these statements.

The goals of BRM are (from the organisation's viewpoint):
Improving the productivity of elicitation
To improve the effectiveness of elicitation
The construction of a library of decisions
Giving insight into relationships between artefacts
Reducing effort in design (requirements and specifications)
Shortening the design phase
The improvement of the productivity in the design
To improve the effectiveness in design
Mobilizing experts
Business rules mapping as much as possible
Quality assurance validation and verification
Automated verification of business rules
Reducing efforts at verification
Verify the improvement of the productivity
Verify the improvement of the effectivity
Generate automated test cases
The automated testing with validation
Give insight which artefacts are hit
Validated and accessible business rules
Reducing testing for implementation independent models
Reducing testing for implementation-dependent models
Reducing the effort for validation
Working with implementation independent business rules to export models
Simplify converting from models into code
Separating of implementation 'know' and 'flow'

Answer:
True
False

Elicitation
The elicitation capability is a bifold function. The first function is to determine the knowledge which realises the value proposition of the business rules. This knowledge needs to be captured from various sources including but not limited to laws and regulations. The second function is the to initiate an impact analysis, this is only done when a business rule architecture is already in place.

Has this capability been part of the BRMS implementation?
Yes
No

Elicitation IST
This section addresses how the elicitation capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

From which source(s) are business rules elicitated?
People
Documents
Data

The end result of the elicitation capability is a set of selected relevant sources?
Yes
No

Which type(s) of analysis have been applied?
Source analysis, elicitation based on sources
Scenario analysis, elicitation based on actual business scenarios

In the case of a change in business rules, is there an impact analysis conducted?
Yes
No

Elicitation SOLL
This section focuses on the situation of how you wanted to see the implementation of the elicitation capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all

The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

From which source(s) are business rules elicitated?
People
Documents
Data

The end result of the elicitation capability is a set of selected relevant sources?
Yes
No

Which type(s) of analysis have been applied?
Source analysis, elicitation based on sources
Scenario analysis, elicitation based on actual business scenarios

In the case of a change in business rules, is there an impact analysis conducted?
Yes
No

Design
The output of the design capability is the business rule architecture. The business rule architecture contains a combination of derivation structures and context designs.

Has this capability been part of the BRMS implementation?
Yes
No

Design IST
This section addresses how the design capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives

The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Which of the following 5 V's are used in the design?
Value represents the value of the decision to the organisation
Velocity represents the speed at which the decision is to be taken
Volume represents the number of times a decision is taken in a given time period
Variety represents the variety of execution paths of a decision which is to be taken
Veracity represents the accuracy at which the decision is to be taken

Which of the following roles are defined (also known as a RAPID)?
Recommend
Agree
Perform
Input
Decide

In the case of a change in business rules, is there an impact analysis conducted?
Yes
No

Design SOLL
This section focuses on the situation of how you wanted to see the implementation of the design capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans.

Which of the following 5 V's are used in the design?
Value represents the value of the decision to the organisation
Velocity represents the speed at which the decision is to be taken
Volume represents the number of times a decision is taken in a given time period
Variety represents the variety of execution paths of a decision which is to be taken
Veracity represents the accuracy at which the decision is to be taken

Which of the following roles are defined (also known as a RAPID)?
Recommend
Agree
Perform
Input
Decide

In the case of a change in business rules, is there an impact analysis conducted?
Yes
No

Specification
The specification capability specifies the content of each separate context design. The function of this capability is to create the business rules and fact types needed to constrain or define particular aspects of the business.

Has this capability been part of the BRMS implementation?
Yes
No

Specification IST
This section addresses how the specification capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

How are the business rules specified?
In models
In text

Are the business rules specified in a unambiguous manner?
In this case, unambiguous means that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine
Yes
No

Specification SOLL
This section focuses on the situation of how you wanted to see the implementation of the specification capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans.

How are the business rules specified?
In models
In text

Are the business rules specified in a unambiguous manner?
In this case, unambiguous means that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine
Yes
No

Verification
The verification capability verifies the created business rule architecture (decisions, business rules, facts etc.) to check for semantic and syntax errors.

Has this capability been part of the BRMS implementation?
Yes
No

Verification IST
This section addresses how the verification capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Wat is the degree of automation?
Detection degree according to the Governance, Risk and Compliance detection matrix.
Automated - Detective
Automated - Preventive
Manual - Detective
Manual - Preventive

Verification SOLL
This section focuses on the situation of how you wanted to see the implementation of the verification capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative

The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans.

Wat is the degree of automation?
Detection degree according to the Governance, Risk and Compliance detection matrix.
Automated - Detective
Automated - Preventive
Manual - Detective
Manual - Preventive

Validation
The validation capability reviews the created value proposition. The goal of this capability is to check for possible errors in its expected behaviour.

Has this capability been part of the BRMS implementation?
Yes
No

Validation IST
This section addresses how the validation capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans.

What kind of validation is applied?
Peer review, a colleague checks if the artefacts are in concurrence with its sources
Scenario validation, scenario-based validation applies predefined test sets to check the behaviour
Source validation, Source-based validation focuses on validation based on the actual sources

During validation, the following attributes are checked:
Traceability, provide an audit trail of access to the business rule and of any changes made to the business rule
Completeness, which data (elements) need to be registered regarding the objects within the process
Accuracy indicates the degree to which the stored data reflects the reality concerning an object
Usability, the learnability and ease of use of the business rule
Other...

Validation SOLL
This section focuses on the situation of how you wanted to see the implementation of the validation capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

What kind of validation is applied?
Peer review, a colleague checks if the artefacts are in concurrence with its sources
Scenario validation, scenario-based validation applies predefined test sets to check the behaviour
Source validation, Source-based validation focuses on validation based on the actual sources

During validation, the following attributes are checked:
Traceability, provide an audit trail of access to the business rule and of any changes made to the business rule
Completeness, which data (elements) need to be registered regarding the objects within the process
Accuracy indicates the degree to which the stored data reflects the reality concerning an object
Usability, the ease of use and learnability of the business rule
Other...

Deployment
The deployment capability transforms the verified and validated value proposition to implementation-dependent executable business rules. The actor that utilises the value proposition is not necessarily have to be a system, this could also be utilised by a subject-matter expert.

Has this capability been part of the BRMS implementation?
Yes
No

Deployment IST
This section addresses how the deployment capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Deployment SOLL
This section focuses on the situation of how you wanted to see the implementation of the deployment capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time

The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Execution
The execution capability processes the output of the deployment capability and is then executed, thereby delivering the actual value proposition. The realisation of the added value is conducted by executing the business rules by human or information system actors.

Has this capability been part of the BRMS implementation?
Yes
No

Execution IST
This section addresses how the execution capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Is 'gaming' taken into consideration?
A system where gaming is possible means a system where users can generate the desired result, by means of testing.
Yes
No

What data is stored during execution?
Input data
Output data
Executed rules

Execution SOLL
This section focuses on the situation of how you wanted to see the implementation of the execution capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Is 'gaming' taken into consideration
A system where gaming is possible means a system where users can generate the desired result.
Yes
No

What data is stored in the execution capability?
Input data
Output data
Executed rules

Monitoring
The monitoring capability observes, checks and keeps a record of not only the execution of the value proposition but also the full range of activities in the previously explained BRM capabilities that are conducted to realise the value proposition.

Has this capability been part of the BRMS implementation?
Yes
No

Monitoring IST
This section addresses how the monitoring capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Which KPI's are being evaluated?
The frequency of corrections per selected context design emerging from the verification process
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule
The frequency of corrections per selected context design emerging from the validation process per type of validation error
The frequency of corrections per selected context architecture emerging from the design process per scope design
The frequency of instantiations per selected context design
The frequency per selected type of validation error
The frequency per selected type of verification error
The number of time units required to define, verify, and validate a single business rule
The frequency of deviations between an implementation dependent context design and an implementation independent context design
The frequency of executions of an implementation dependent business rule
The frequency of execution variants of a scope design
The number of time units required for the execution per execution variant
The amount of business rules that cannot be automated

Answers:
Not evaluated
Constantly
Daily
Weekly
Monthly
Quarterly
Yearly

Do you measure any additional KPI's? (open question)

Monitoring SOLL
This section focuses on the situation of how you wanted to see the implementation of the monitoring capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Which KPI's are being evaluated?
The frequency of corrections per selected context design emerging from the verification process
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule
The frequency of corrections per selected context design emerging from the validation process per type of validation error
The frequency of corrections per selected context architecture emerging from the design process per scope design
The frequency of instantiations per selected context design
The frequency per selected type of validation error
The frequency per selected type of verification error
The number of time units required to define, verify, and validate a single business rule
The frequency of deviations between an implementation dependent context design and an implementation independent context design
The frequency of executions of an implementation dependent business rule
The frequency of execution variants of a scope design
The number of time units required for the execution per execution variant
The amount of business rules that cannot be automated

Answers:
Constantly
Daily
Weekly
Monthly

Quarterly
Yearly
Not evaluated

Do you want to measure any additional KPI's? (open question)

Governance
The governance capability consists of three sub-capabilities; version management, traceability, and validity management.

- **Version management**

The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities.

- **Validity management**

The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time.

- **Traceability**

The purpose of the traceability capability is to ensure the possibility to trace created elements to their corresponding laws and regulations.

Has this capability been part of the BRMS implementation?
Yes
No

Governance IST
This section addresses how the governance capability has been implemented.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Governance consists of the following capabilities:

Version management
Validity management
Traceability

Governance SOLL
This section focuses on the situation of how you wanted to see the implementation of the governance capability. Without any restriction of knowledge and money.

Who is in the lead of this capability?
IT
Business
Central IT/Business group

What is the autonomy of this capability?
The degree of autonomy from low (not autonomous) to high (autonomously) of this capability. The ten degrees of autonomy are adopted from the Sheridan model (Parasuraman, Sheridan, & Wickens, 2000).
The computer does not help, people must do it all
The computer provides a complete set of action alternatives
The computer narrows down the choice to a few action alternatives
The computer suggests one action alternative
The computer runs one action suggestion if the human approves
Man can veto before it runs automatically in a limited time
The computer runs automatically and informs people where necessary
The computer informs the human after execution only when it is requested
The computer informs the human after execution if the computer which decides
The computer decides everything independently and ignores humans

Governance consists of the following capabilities:
Version management
Validity management
Traceability

Contact information
The information you provide will only be used for this research. The results of the questionnaire will be treated as confidential data. If you are interested in the research, please leave your contact information. This data is used solely for the purpose to keep you up to date of this investigation.

What is your e-mailadress?

Appendix C – Ultrametric-distance calculations

This appendix contains the ultrametric-distance calculations for the other problem classes (design, specification, verification, validation, deployment, execution, monitoring and governance) besides that of the elicitation problem class.

Ultrametric-distance Design problem class

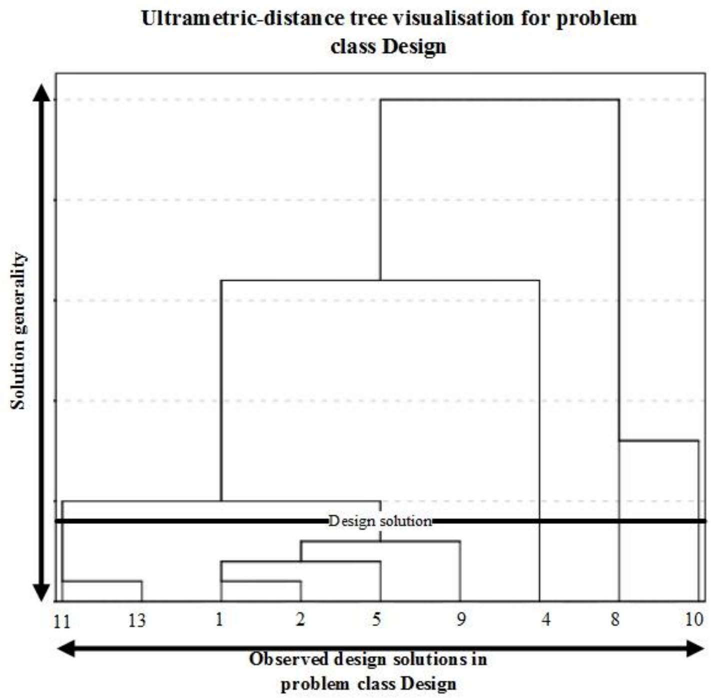


Figure 30 Ultrametric-distance tree visualisation for problem class design

Ultrametric-distance Specification problem class

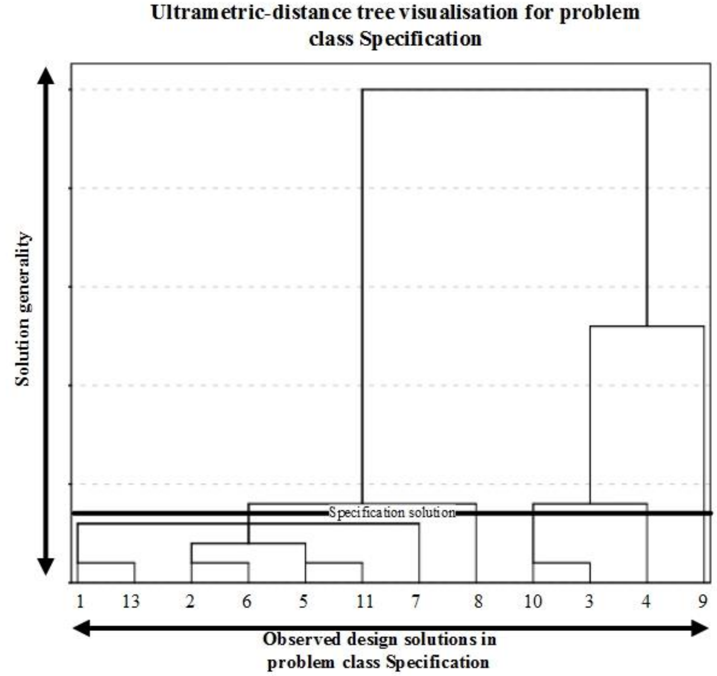


Figure 31 Ultrametric-distance tree visualisation for problem class specification

Ultrametric-distance Verification problem class

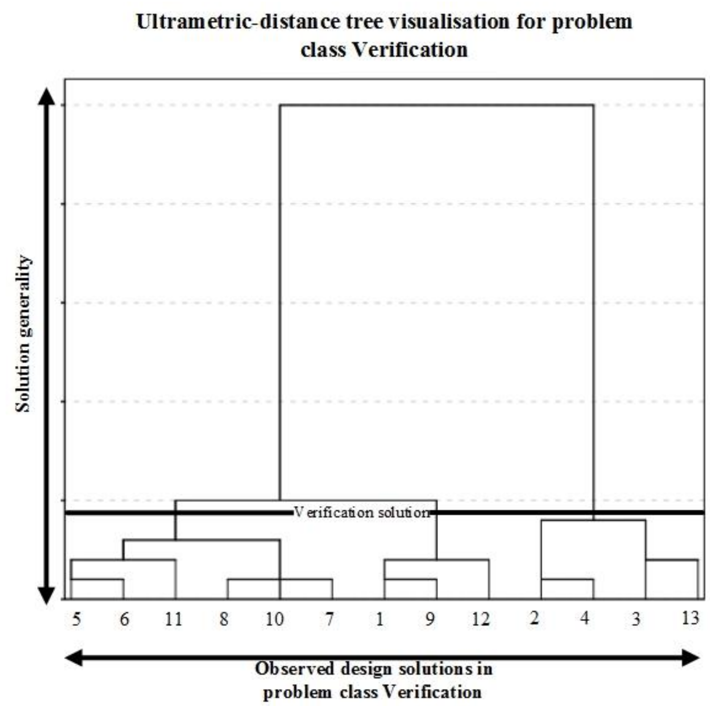


Figure 32 Ultrametric-distance tree visualisation for problem class verification

Ultrametric-distance Validation problem class

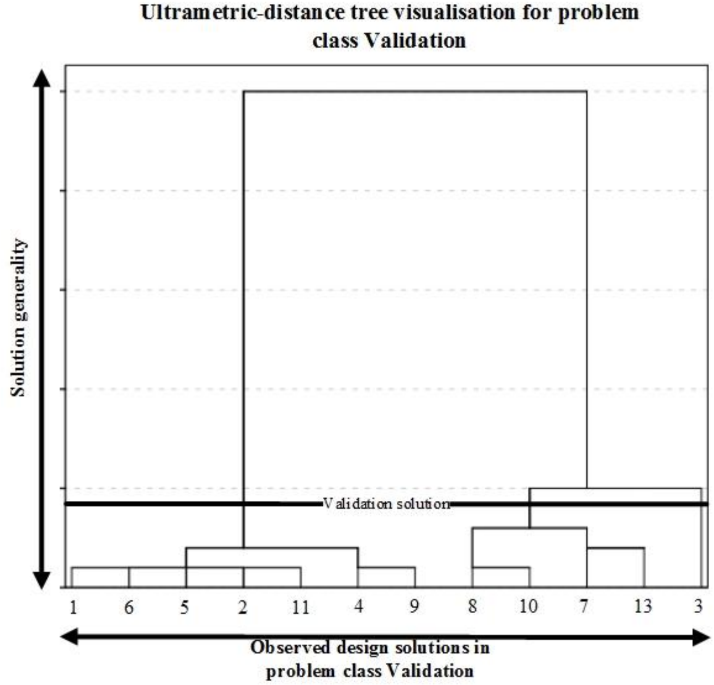


Figure 33 Ultrametric-distance tree visualisation for problem class validation

Ultrametric-distance Deployment problem class

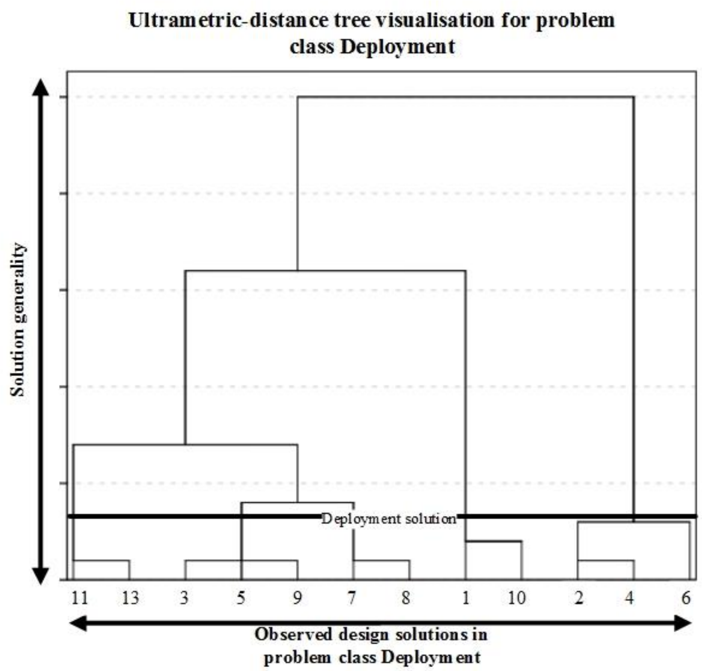


Figure 34 Ultrametric-distance tree visualisation for problem class deployment

Ultrametric-distance Execution problem class

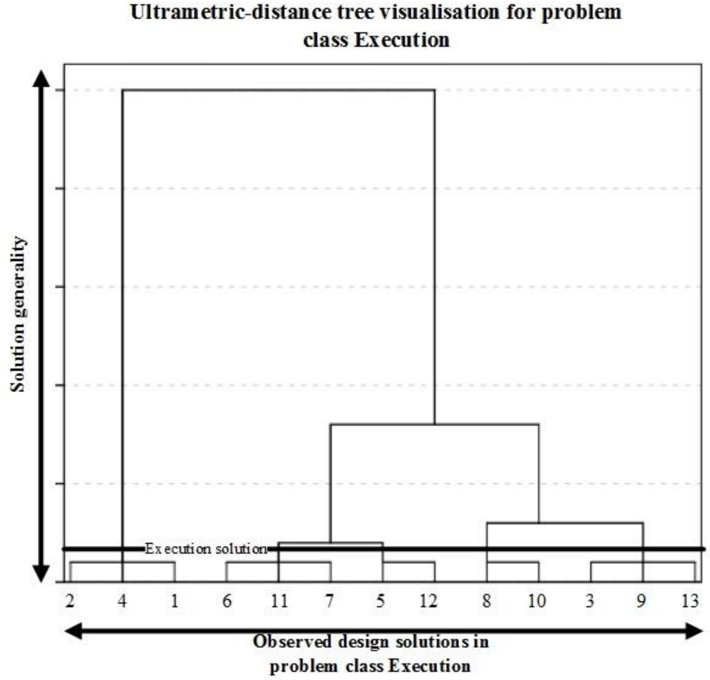


Figure 35 Ultrametric-distance tree visualisation for problem class execution

Ultrametric-distance Monitoring problem class

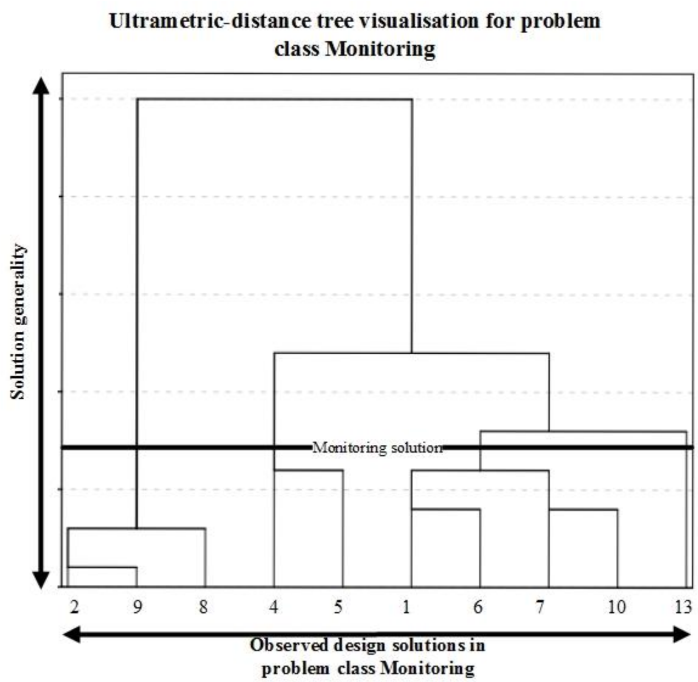


Figure 36 Ultrametric-distance tree visualisation for problem class monitoring

Ultrametric-distance Governance problem class

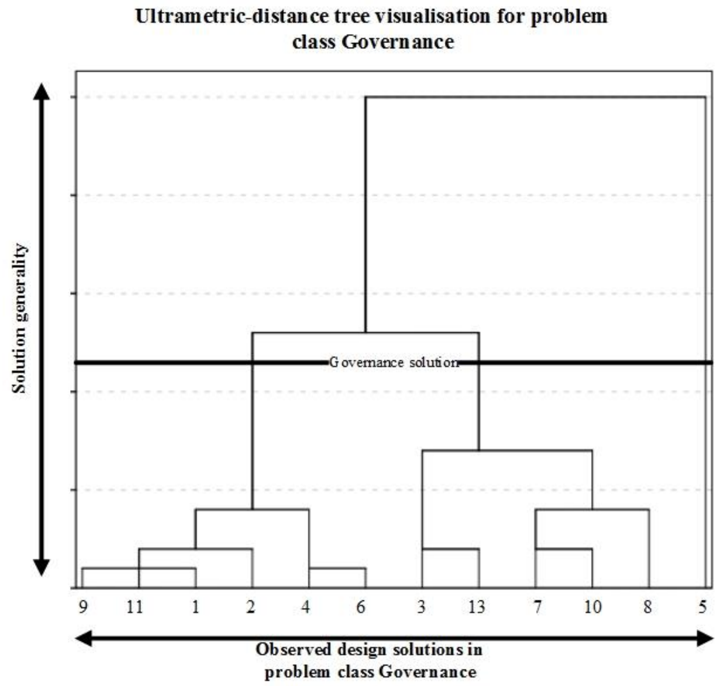


Figure 37 Ultrametric-distance tree visualisation for problem class governance

Appendix D – Generality level determinations

This appendix contains the k-means cluster analysis to determine the level of generality of the design solution. These cluster analysis are conducted for the other problem classes (design, specification, verification, validation, deployment, execution, monitoring and governance) besides that of the elicitation problem class.

Generality level determination Design problem class

Figure 38 K-means cluster analysis for generality determination (design)

Elbow: 5 or 8

Generality level determination Specification problem class

Figure 39 K-means cluster analysis for generality determination (specification)

Elbow: 5 or 9

Generality level determination Verification problem class

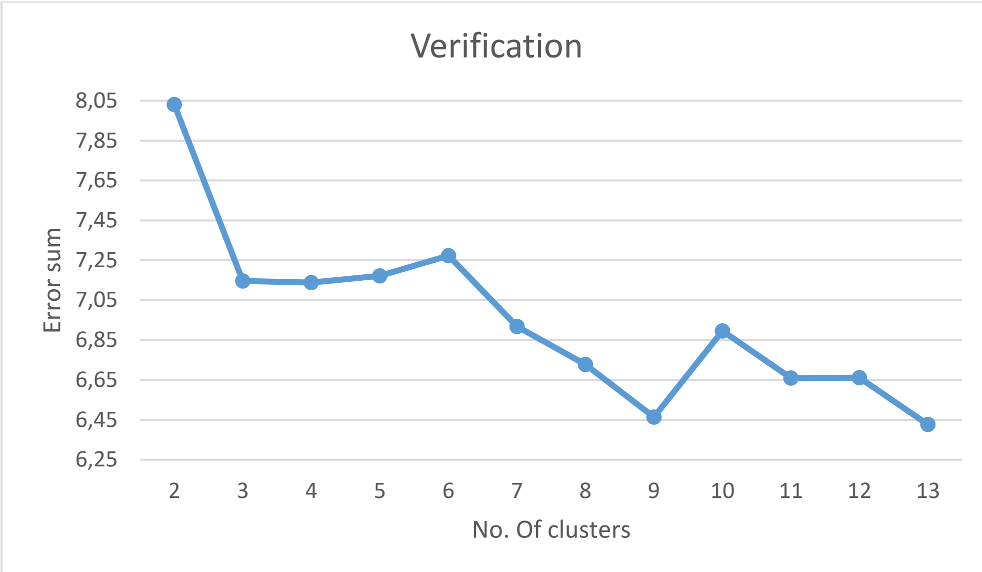


Figure 40 K-means cluster analysis for generality determination (verification)

Elbow: 3,9 or 11

Generality level determination Validation problem class

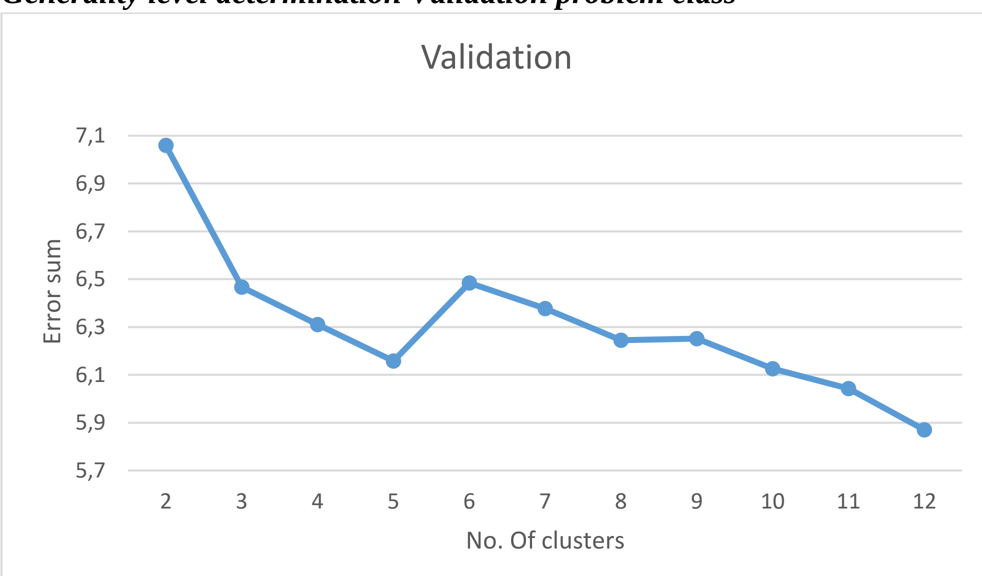


Figure 41 K-means cluster analysis for generality determination (validation)

Elbow: 3,5 or 8

Generality level determination Deployment problem class

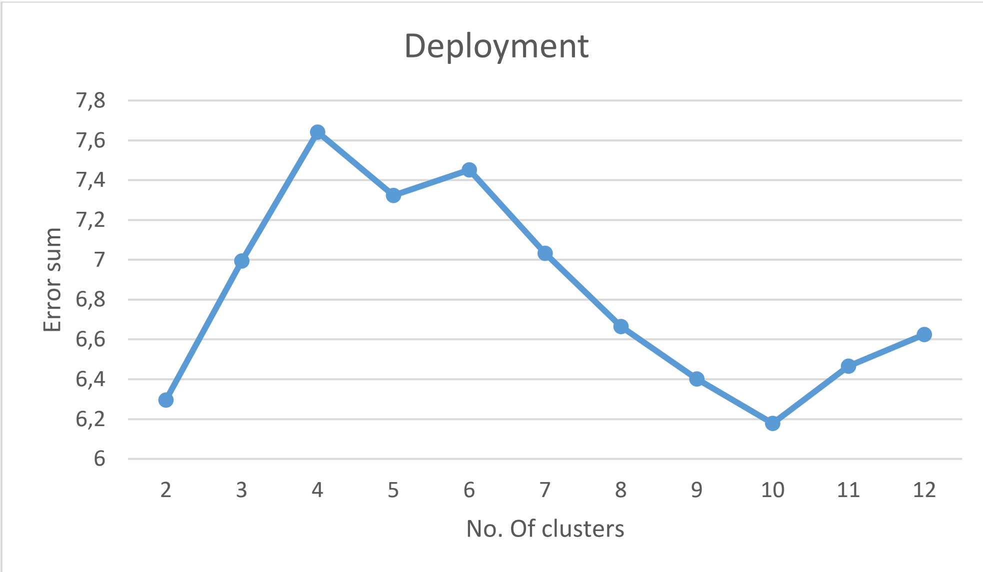


Figure 42 K-means cluster analysis for generality determination (deployment)

Elbow: 5 or 10

Generality level determination Execution problem class

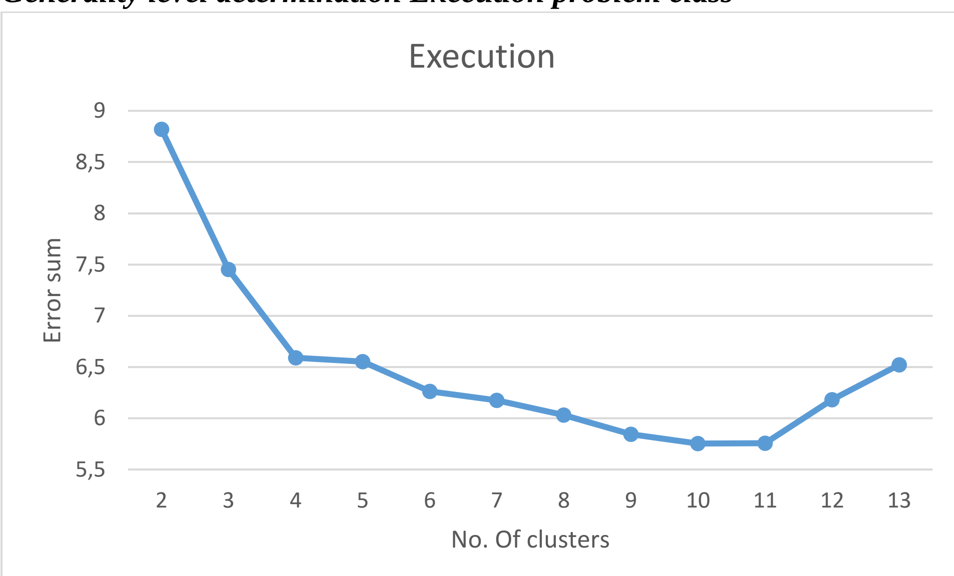


Figure 43 K-means cluster analysis for generality determination (execution)

Elbow: 4 or 11

Generality level determination Monitoring problem class

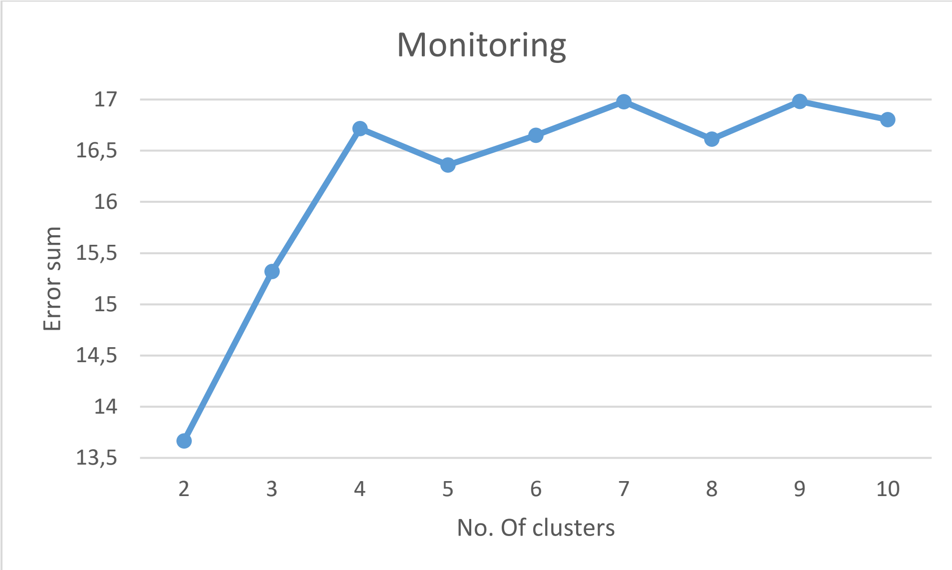


Figure 44 K-means cluster analysis for generality determination (monitoring)

Elbow: 4,7 or 9

Generality level determination Governance problem class

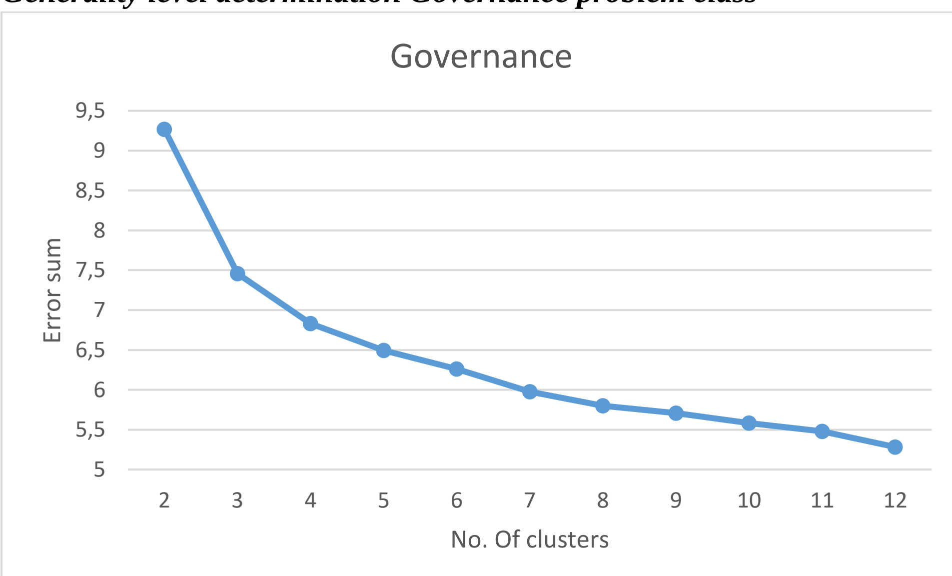


Figure 45 K-means cluster analysis for generality determination (governance)

Elbow: 3

Appendix E – Design situations

This appendix contains the design situations for all the nine problem classes (elicitation, design, specification, verification, validation, deployment, execution, monitoring, and governance).

Elicitation design situations

Design situation 1

Public sector focused
Organisations with 2001 – 5000 employees
Organisation implementation focused
Fully manual elicitation focused (autonomy level 1)
Database data is used for elicitating business rules
Outcome of the capability is not a relevant set of selected sources
Scenario analysis is conducted

Design situation 2

Public sector focused
Organisations with more than 5000 employees
Line of business implementation focused
Capability is supported with a complete set of action alternatives (autonomy level 2)
No database data is used for elicitating business rules
Outcome of the capability is a relevant set of selected sources
No scenario analysis is conducted

Design situation 3

Financial sector focused
Organisations with more than 5000 employees
Line of business implementation focused
Capability is supported with a narrowed down set of action alternatives (autonomy level 3)
No database data is used for elicitating business rules
Outcome of the capability is not a relevant set of selected sources
No scenario analysis is conducted

Design situation 4

Financial sector focused
Organisations with 251 -500 employees
Line of business implementation focused
Fully manual elicitation focused (autonomy level 1)
No database data is used for elicitating business rules
Outcome of the capability is a relevant set of selected sources
Scenario analysis is conducted

Design design situations

Design situation 1

Financial sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
The business is the capability leader
Capability is supported with a narrowed down set of action alternatives (autonomy level 3)
Velocity is not taken into consideration in the design capability
Volume is taken into consideration in the design capability
Veracity is taken into consideration in the design capability
The agree role is defined in the design capability
The perform role is defined in the design capability
The input role is defined in the design capability
The recommend role is defined in the design capability
An impact analysis is conducted in the design capability

Design situation 2

Public sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
The business is the capability leader
Capability is supported but a Man can veto before it runs automatically in a limited time (autonomy level 6)
Velocity is taken into consideration in the design capability
Volume is taken into consideration in the design capability
Veracity is not taken into consideration in the design capability
The agree role is defined in the design capability
The perform role is defined in the design capability
The input role is not defined in the design capability
The recommend role is not defined in the design capability
An impact analysis is conducted in the design capability

Design situation 3

Financial sector focused
Organisations with 251 – 500 employees
Line of business implementation focused
The Central IT/Business group is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
Velocity is taken into consideration in the design capability
Volume is not taken into consideration in the design capability
Veracity is not taken into consideration in the design capability
The agree role is not defined in the design capability
The perform role is not defined in the design capability
The input role is defined in the design capability
The recommend role is not defined in the design capability
An impact analysis is not conducted in the design capability

Design situation 4

Public sector focused
Organisations with 251 – 500 employees
Application implementation focused
The Central IT/Business group is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
Velocity is not taken into consideration in the design capability
Volume is not taken into consideration in the design capability
Veracity is not taken into consideration in the design capability
The agree role is not defined in the design capability
The perform role is not defined in the design capability
The input role is defined in the design capability
The recommend role is not defined in the design capability
An impact analysis is not conducted in the design capability

Design situation 5

Public sector focused
Organisations with >5000 employees
Organisation wide implementation focused
The business is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
Velocity is not taken into consideration in the design capability
Volume is not taken into consideration in the design capability
Veracity is not taken into consideration in the design capability
The agree role is not defined in the design capability
The perform role is defined in the design capability
The input role is defined in the design capability
The recommend role is defined in the design capability
An impact analysis is conducted in the design capability

Specification design situations

Design situation 1

Public sector focused
Organisations with 2001 – 5000 employees
Line of business implementation focused
The business is the capability leader
The capability is supported and The computer provides a complete set of action alternatives (autonomy level 2)
The business rules are specified in models
The business rules are specified unambiguous

Design situation 2

Public sector focused
Organisations with 2001 – 5000 employees

Organisation wide implementation focused
The business is the capability leader
The capability is supported Man can veto before it runs automatically in a limited time (autonomy level 6)
The business rules are specified in text
The business rules are not specified unambiguous

Design situation 3

Financial sector focused
Organisations with 2001 – 5000 employees
Line of business implementation focused
The central IT/business group is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
The business rules are specified in text
The business rules are specified unambiguous

Design situation 4

Financial sector focused
Organisations with 501 – 1000 employees
Line of business implementation focused
The IT is the capability leader
The capability is supported and The computer provides a complete set of action alternatives (autonomy level 2)
The business rules are specified in models
The business rules are not specified unambiguous

Design situation 5

Financial sector focused
Organisations with 251 – 500 employees
Line of business implementation focused
The business is the capability leader
The capability is supported and The computer runs automatically and informs people where necessary (autonomy level 7)
The business rules are specified in models
The business rules are specified unambiguous

Verification design situations

Design situation 1

Financial sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
The capability is supported and The computer decides everything independently and ignores humans (autonomy level 10)
The detection degree of the verification capability is automated – detective

Design situation 2

Public sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
The computer provides a complete set of action alternatives (autonomy level 2)
The detection degree of the verification capability is manual - preventive

Design situation 3

Financial sector focused
Organisations with 1001 – 2000 employees
Line of business implementation focused
The capability is supported and The computer runs automatically and informs people where necessary (autonomy level 7)
The detection degree of the verification capability is automated – detective

Validation design situations

Design situation 1

Public sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
Peer reviews are used for validation
Scenario validation is used for validation
Source validation is not used for validation
Traceability is not checked in the validation capability
Accuracy is checked in the validation capability

Design situation 2

Public sector focused
Organisations with 5000> employees
Application focused implementation focused
The capability is supported The computer provides a complete set of action alternatives (autonomy level 2)
Peer reviews are used for validation
Scenario validation is not used for validation
Source validation is not used for validation
Traceability is checked in the validation capability
Accuracy is not checked in the validation capability

Design situation 3

Financial sector focused
Organisations with 501 - 1000 employees
Line of business implementation focused

The capability is supported The computer runs automatically and informs people where necessary (autonomy level 7)
Peer reviews are not used for validation
Scenario validation is used for validation
Source validation is used for validation
Traceability is checked in the validation capability
Accuracy is not checked in the validation capability

Deployment design situations

Design situation 1

Public sector focused
Organisations with 1001 - 2000 employees
Line of business implementation focused
Business is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)

Design situation 2

Public sector focused
Organisations with 2001 - 5000 employees
Organisation wide implementation focused
Business is the capability leader
The capability is supported The computer provides a complete set of action alternatives (autonomy level 2)

Design situation 3

Financial sector focused
Organisations with >5000 employees
Line of business implementation focused
IT is the capability leader
The capability is supported The computer runs automatically and informs people where necessary (autonomy level 7)

Design situation 4

Financial sector focused
Organisations with 501 - 1000 employees
Line of business implementation focused
Business is the capability leader
The capability is supported Man can veto before it runs automatically in a limited time (autonomy level 6)

Design situation 5

Public sector focused
Organisations with >5000 employees

Organisation wide implementation focused
IT is the capability leader
The capability is supported The computer decides everything independently and ignores humans (autonomy level 10)

Execution design situations

Design situation 1

Public sector focused
Organisations with 2001 - 5000 employees
Organisation wide implementation focused
Business is the capability leader
The capability is supported The computer provides a complete set of action alternatives (autonomy level 2)
Gaming is not taken into consideration in the execution capability
The executed rules are saved in the execution capability

Design situation 2

Public sector focused
Organisations with 5000> employees
Line of business implementation focused
Business is the capability leader
The capability is supported The computer runs automatically and informs people where necessary (autonomy level 7)
Gaming is not taken into consideration in the execution capability
The executed rules are not saved in the execution capability

Design situation 3

Financial sector focused
Organisations with 1001 - 2000 employees
Line of business implementation focused
Central IT/business group is the capability leader
The capability is supported The computer decides everything independently and ignores humans (autonomy level 10)
Gaming is not taken into consideration in the execution capability
The executed rules are saved in the execution capability

Design situation 4

Financial sector focused
Organisations with 2001 - 5000 employees
Line of business implementation focused
IT is the capability leader
The capability is supported The computer decides everything independently and ignores humans (autonomy level 10)
Gaming is taken into consideration in the execution capability
The executed rules are not saved in the execution capability

Design situation 5

Financial sector focused
Organisations with 251 – 500 employees
Line of business implementation focused
Central IT/business group is the capability leader
The capability is supported The computer runs automatically and informs people where necessary (autonomy level 7)
Gaming is not taken into consideration in the execution capability
The executed rules are saved in the execution capability

Monitoring design situations

Design situation 1

Financial sector focused
Organisations with 1001 – 2000 employees
Line of business implementation focused
Business is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)

What is evaluated?	Design situation 1
The frequency of corrections per selected context design emerging from the verification process	Not evaluated
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error	Not evaluated
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule	Not evaluated
The frequency of corrections per selected context design emerging from the validation process per type of validation error	Not evaluated
The frequency of corrections per selected context architecture emerging from the design process per scope design	Not evaluated
The frequency of instantiations per selected context design	Not evaluated
The frequency per selected type of validation error	Not evaluated
The frequency per selected type of verification error	Not evaluated
The number of time units required to define, verify, and validate a single business rule	Yearly
The frequency of deviations between an implementation dependent context design and an implementation independent context design	Not evaluated
The frequency of executions of an implementation dependent business rule	Not evaluated
The frequency of execution variants of a scope design	Not evaluated
The number of time units required for the execution per execution variant	Yearly
The amount of business rules that cannot be automated	Yearly

Design situation 2

Financial sector focused
Organisations with 1001 – 2000 employees
Organisation wide implementation focused

Central IT/business group is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)

What is evaluated?	Design situation 2
The frequency of corrections per selected context design emerging from the verification process	Not evaluated
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error	Not evaluated
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule	Not evaluated
The frequency of corrections per selected context design emerging from the validation process per type of validation error	Not evaluated
The frequency of corrections per selected context architecture emerging from the design process per scope design	Not evaluated
The frequency of instantiations per selected context design	Not evaluated
The frequency per selected type of validation error	Not evaluated
The frequency per selected type of verification error	Not evaluated
The number of time units required to define, verify, and validate a single business rule	Not evaluated
The frequency of deviations between an implementation dependent context design and an implementation independent context design	Not evaluated
The frequency of executions of an implementation dependent business rule	Not evaluated
The frequency of execution variants of a scope design	Not evaluated
The number of time units required for the execution per execution variant	Not evaluated
The amount of business rules that cannot be automated	Not evaluated

Design situation 3

Financial sector focused
Organisations with 2001 – 5000 employees
Line of business implementation focused
Central IT/business group is the capability leader
The capability is supported The computer provides a complete set of action alternatives (autonomy level 2)

What is evaluated?	Design situation 3
The frequency of corrections per selected context design emerging from the verification process	Weekly
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error	Yearly
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule	Yearly
The frequency of corrections per selected context design emerging from the validation process per type of validation error	Weekly
The frequency of corrections per selected context architecture emerging from the design process per scope design	Weekly
The frequency of instantiations per selected context design	Weekly
The frequency per selected type of validation error	Yearly
The frequency per selected type of verification error	Yearly
The number of time units required to define, verify, and validate a single business rule	Monthly

The frequency of deviations between an implementation dependent context design and an implementation independent context design	Constantly
The frequency of executions of an implementation dependent business rule	Weekly
The frequency of execution variants of a scope design	Weekly
The number of time units required for the execution per execution variant	Monthly
The amount of business rules that cannot be automated	Yearly

Design situation 4

Public sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
Central IT/business group is the capability leader
The capability is supported The computer narrows down the choice to a few action alternatives (autonomy level 3)

What is evaluated?	Design situation 4
The frequency of corrections per selected context design emerging from the verification process	Daily
The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error	Constantly
The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule	Constantly
The frequency of corrections per selected context design emerging from the validation process per type of validation error	Not evaluated
The frequency of corrections per selected context architecture emerging from the design process per scope design	Quarterly
The frequency of instantiations per selected context design	Not evaluated
The frequency per selected type of validation error	Not evaluated
The frequency per selected type of verification error	Not evaluated
The number of time units required to define, verify, and validate a single business rule	Not evaluated
The frequency of deviations between an implementation dependent context design and an implementation independent context design	Not evaluated
The frequency of executions of an implementation dependent business rule	Yearly
The frequency of execution variants of a scope design	Not evaluated
The number of time units required for the execution per execution variant	Not evaluated
The amount of business rules that cannot be automated	Not evaluated

Governance design situations

Design situation 1

Public sector focused
Organisations with 2001 – 5000 employees
Organisation wide implementation focused
Central IT/business group is the capability leader
The capability is supported The computer provides a complete set of action alternatives (autonomy level 2)

Traceability is used in the governance capability

Design situation 2

Public sector focused
Organisations with 1001 – 2000 employees
Line of business implementation focused
business is the capability leader
The capability is not supported The computer does not help, people must do it all (autonomy level 1)
Traceability is used in the governance capability

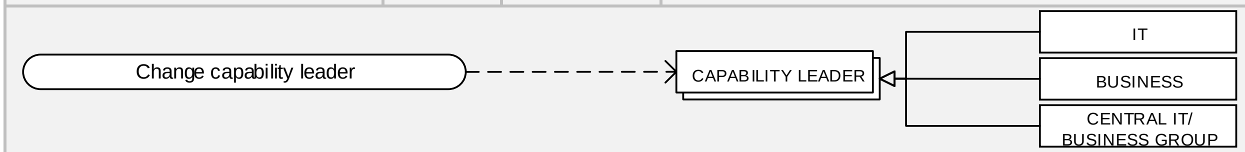
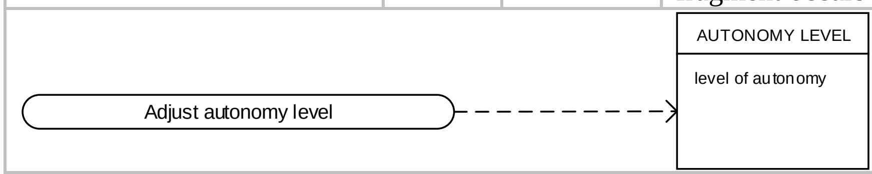
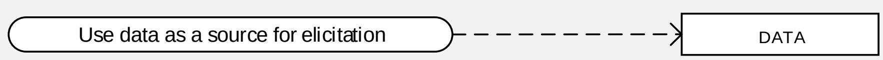
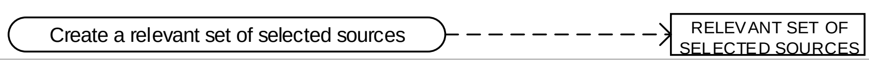
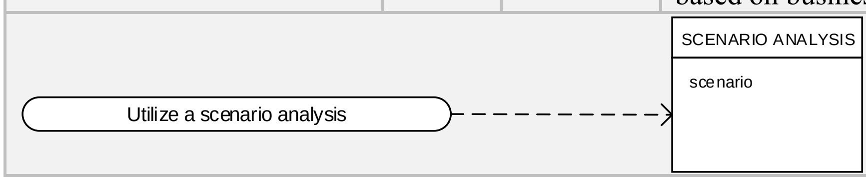
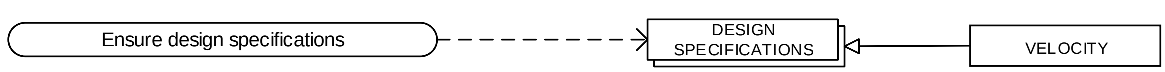
Design situation 3

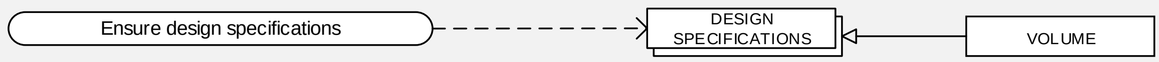
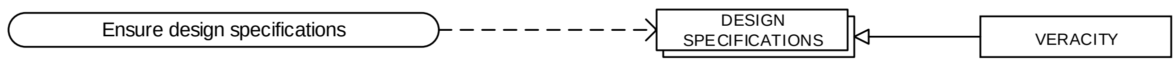
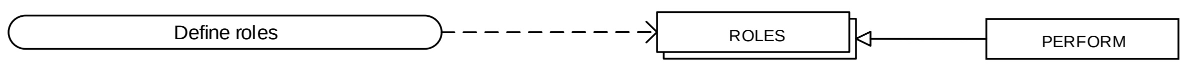
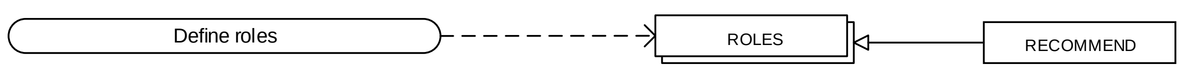
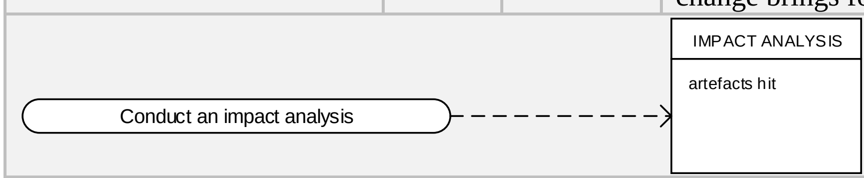
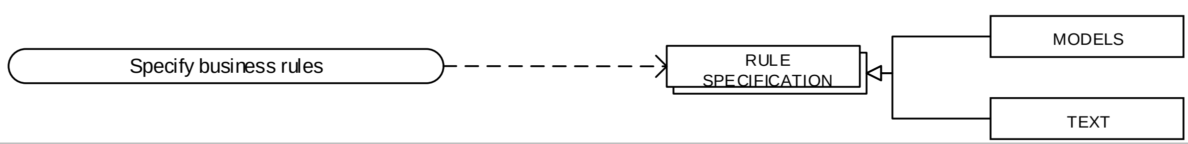
Public sector focused
Organisations with 2001 – 5000 employees
Application focused implementation focused
IT is the capability leader
The capability is supported and The computer informs the human after execution if the computer which decides (autonomy level 9)
Traceability is not used in the governance capability

Appendix F – Characterising design factor method fragments

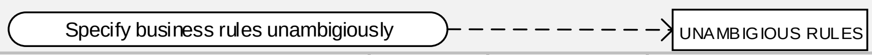
This appendix contains the combination of characterising design factors and method fragments.

Table 18 Method fragments created

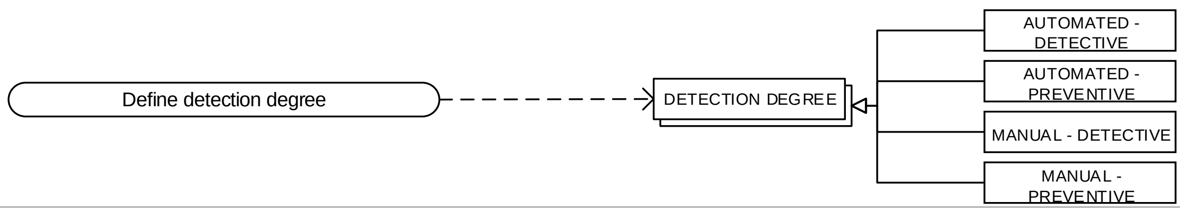
Design factor	Design factor#	Method fragment#	Description:
Capability leader	**#4**	**#1**	A specific group which is leader of the capability. This method fragment is used in the following problem classes: Design, Specification, Deployment, Execution, and Governance.
			
Capability autonomy	**#5**	**#2**	The level of autonomy of the capability. This method fragment occurs in all the nine problem classes.
			
Elicitation source (Data)	**#6**	**#3**	Database data is used as an source for elicitation.
			
Relevant set of selected sources	**#7**	**#4**	The output of elicitation is a relevant set of selected sources.
			
Scenario analysis	**#8**	**#5**	Scenario analysis is used during elicitation. Elicitation based on business scenario's.
			
Velocity design	**#9**	**#6**	Velocity is one of the 5v's and represents the speed at which the decision is to be taken.
			

Volume design	**#10**	**#7**	Volume is one of the 5v's and represents the number of times a decision is taken in a given time period.
			
Veracity design	**#11**	**#8**	Veracity is one of the 5v's and represents the accuracy at which the decision is to be taken.
			
Role: Agree	**#12**	**#9**	The Agree (RAPID) role is defined in the design problem class.
			
Role: Perform	**#13**	**#10**	The Perform (RAPID) role is defined in the design problem class.
			
Role: Input	**#14**	**#11**	The Input (RAPID) role is defined in the design problem class.
			
Role: Recommend	**#15**	**#12**	The Recommend (RAPID) role is defined in the design problem class.
			
Impact analysis	**#16**	**#13**	An impact analysis is checks the impact that a certain change brings forth
			
Rule specification	**#17**	**#14**	Rules are specified in either models or in text.
			

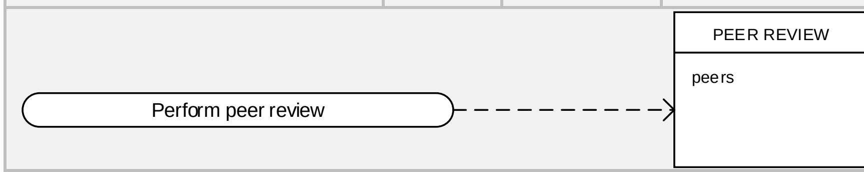
Unambiguous specification	**#18**	**#15**	Unambiguous specification is that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine.



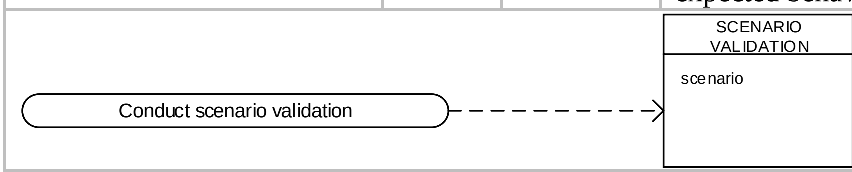
Detection degree	**#19**	**#16**	Detection degree indicates at which degree syntax and semantic errors are detected.



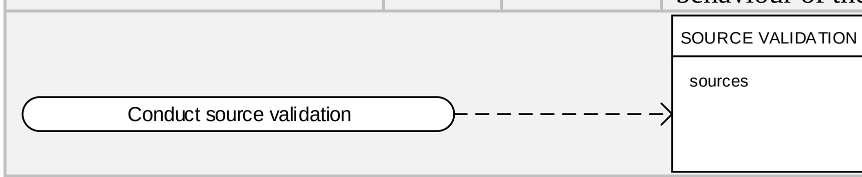
Peer review	**#20**	**#17**	Peer reviews are conducted to validate the expected behaviour of the business rules.



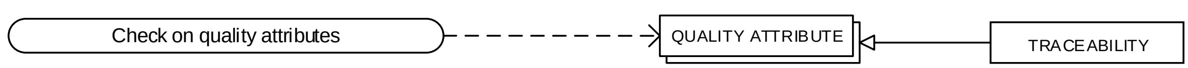
Scenario validation	**#21**	**#18**	Scenario validation is conducted to validate the expected behaviour of the business rules.



Source validation	**#22**	**#19**	Source validation is conducted to validate the expected behaviour of the business rules.

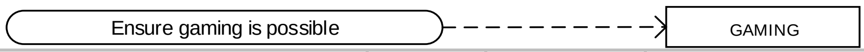


Quality attribute: Traceability	**#23**	**#20**	The validation capability checks on the quality attribute traceability.

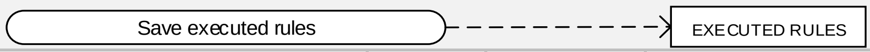


Quality attribute: Accuracy	**#24**	**#21**	The validation capability checks on the quality attribute accuracy.

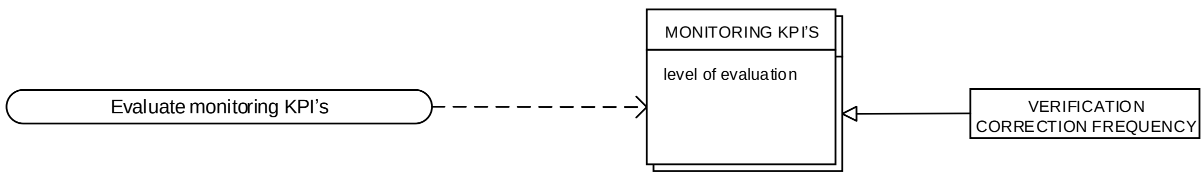
Gaming	**#25**	**#22**	Gaming is made possible for the purpose of testing.



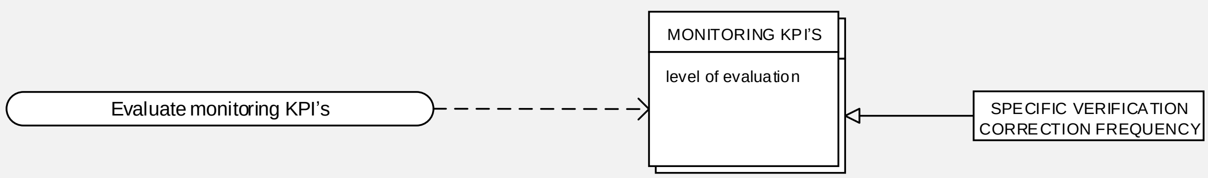
Rules saved	**#26**	**#23**	The executed rules are saved in the execution capability.



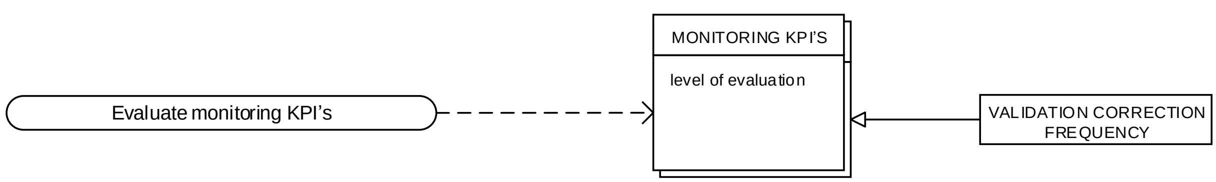
Verification correction frequency	**#27**	**#24**	The frequency of corrections per selected context design emerging from the verification process.



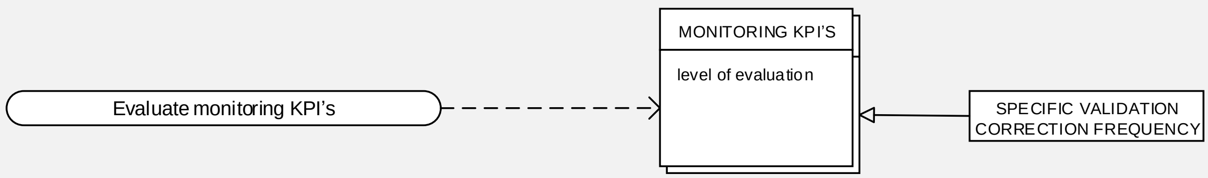
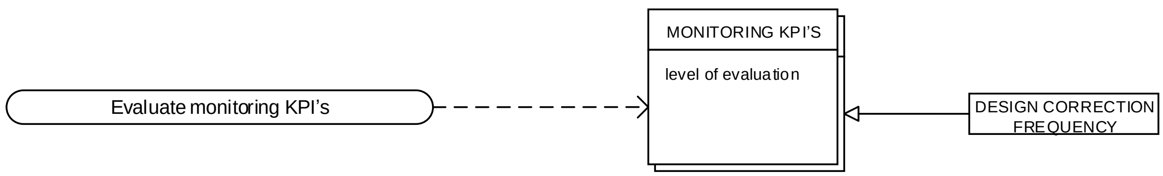
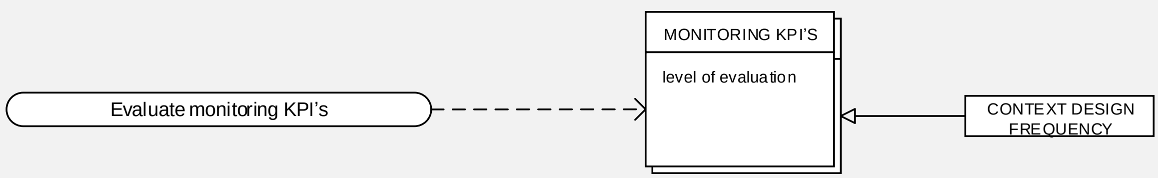
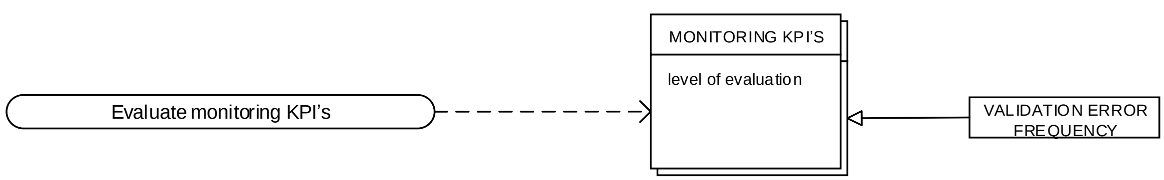
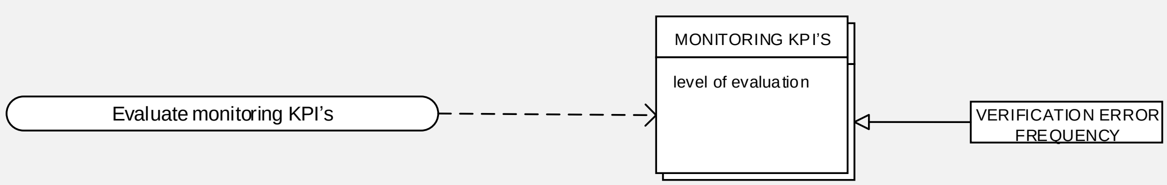
Specific Verification correction frequency	**#28**	**#25**	The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error.

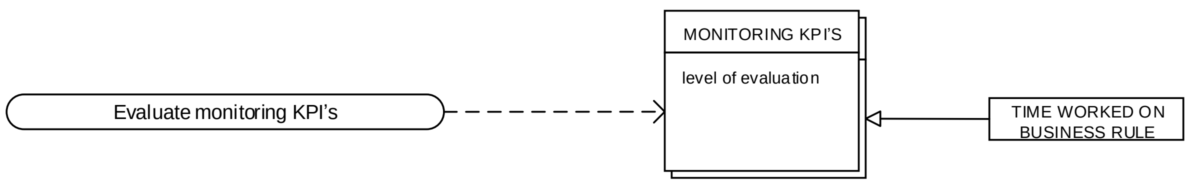
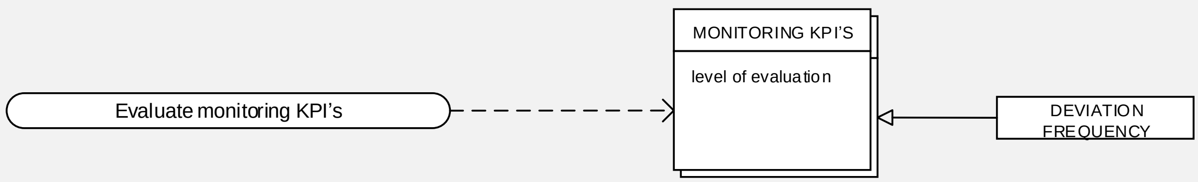
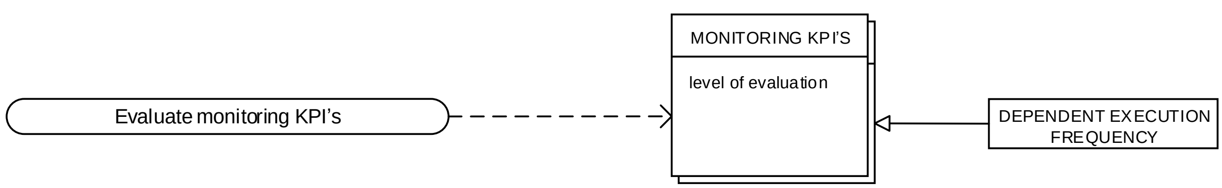
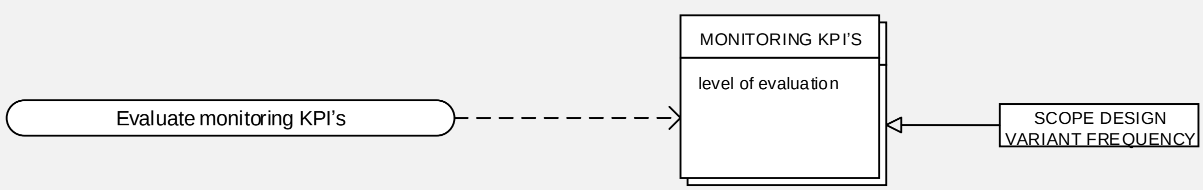


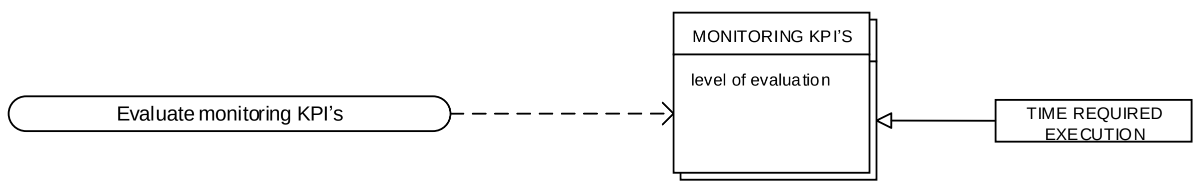
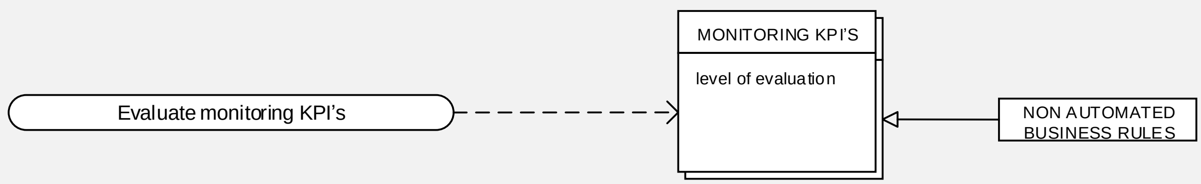
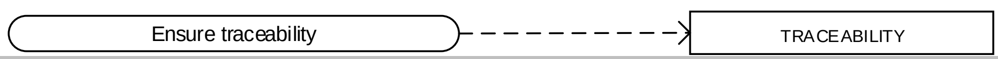
Validation correction frequency	**#29**	**#26**	The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule.



Specific validation correction frequency	**#30**	**#27**	The frequency of corrections per selected context design emerging from the validation process per type of validation error.

			
Design correction frequency	**#31**	**#28**	The frequency of corrections per selected context architecture emerging from the design process per scope design.
			
Context design frequency	**#32**	**#29**	The frequency of instantiations per selected context design.
			
Validation error frequency	**#33**	**#30**	The frequency per selected type of a validation error.
			
Verification error frequency	**#34**	**#31**	The frequency per selected type of a verification error.
			

<table>
<tr><td>Time worked on business rule</td><td>#35</td><td>#32</td><td>The number of time units required to define, verify, and validate a single business rule.</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td>Deviation frequency</td><td>#36</td><td>#33</td><td>The frequency of deviations between an implementation dependent context design and an implementation independent context design.</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td>Dependent execution frequency</td><td>#37</td><td>#34</td><td>The frequency of executions of an implementation dependent business rule.</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td>Scope design variant frequency</td><td>#38</td><td>#35</td><td>The frequency of execution variants of a scope design.</td></tr>
<tr><td colspan="4">
</td></tr>
<tr><td>Time required execution</td><td>#39</td><td>#36</td><td>The number of time units required for the execution per execution variant.</td></tr>
</table>

			
Non automated business rules	**#40**	**#37**	The amount of business rules that cannot be automated.
			
Traceability	**#41**	**#38**	Traceability ensures the possibility to trace created elements to their corresponding sources.
			

Appendix G – Non-characterising design factor method fragments

This appendix contains the combination of non-characterising design factors and method fragments.

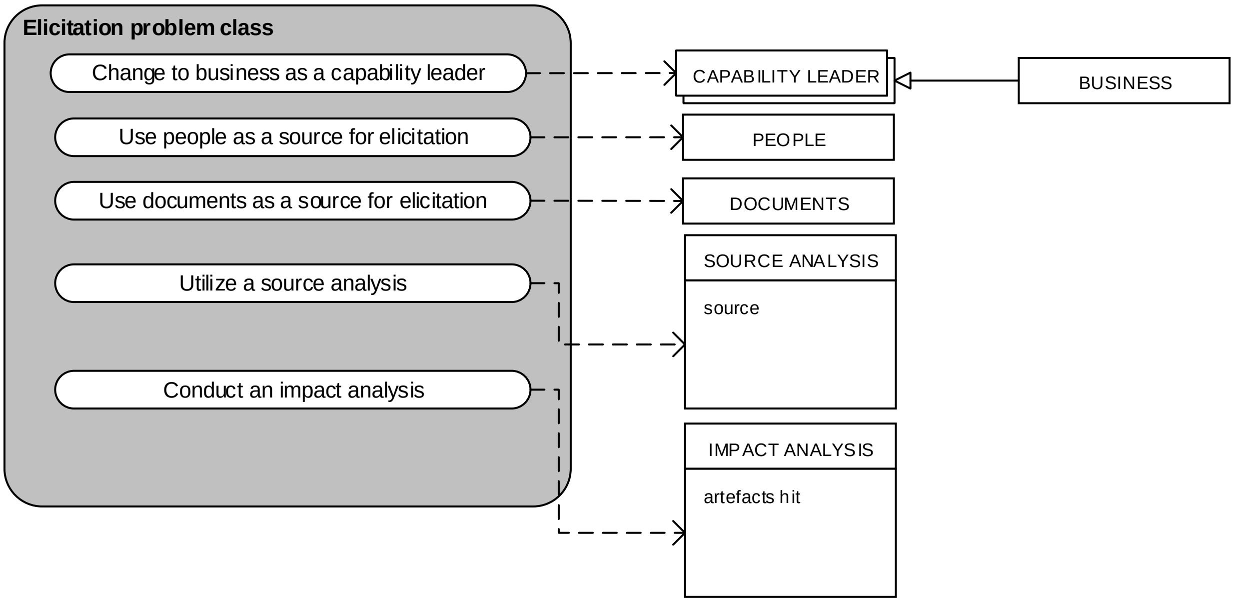


Figure 41 Non-characterising design factor method fragments elicitation (N1)

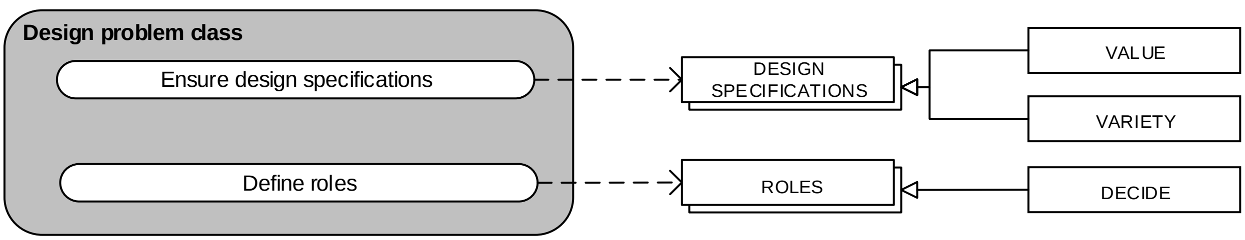


Figure 42 Non-characterising design factor method fragments design (N2)

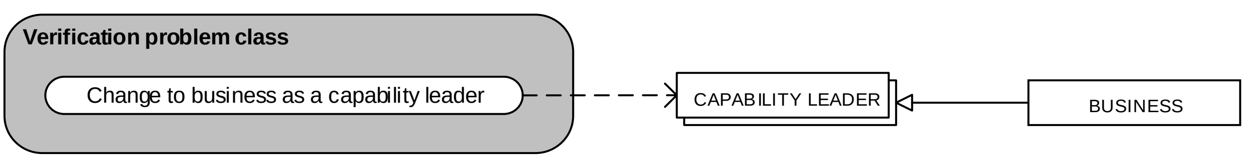


Figure 43 Non-characterising design factor method fragments verification (N3)

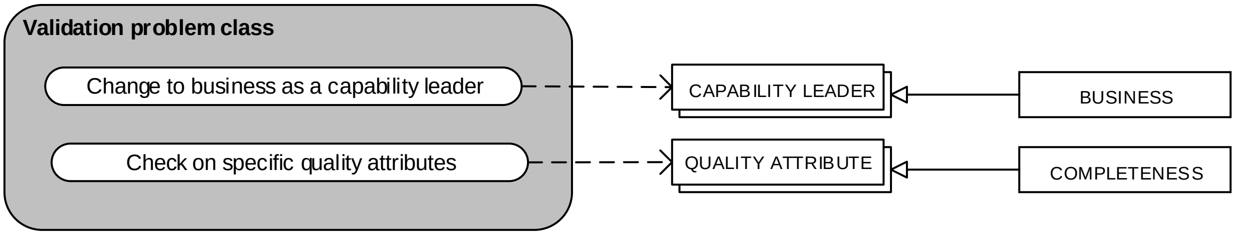


Figure 44 Non-characterising design factor method fragments validation (N4)

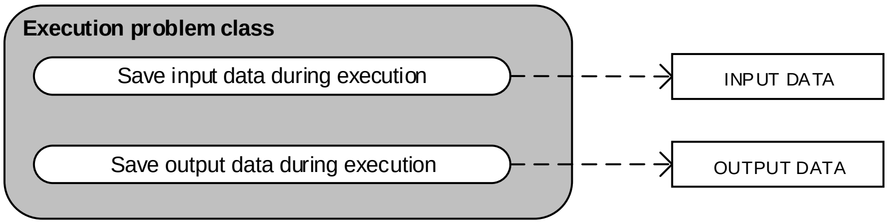


Figure 45 Non-characterising design factor method fragments execution (N5)

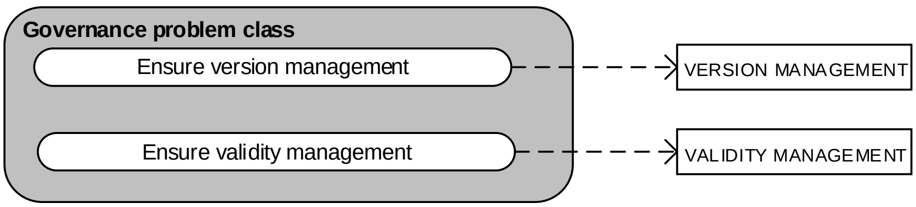


Figure 46 Non-characterising design factor method fragments governance (N6)

Appendix H – Specifications of the Platform Dependent Metamodel

This appendix contains the specifications of the Platform Dependent Metamodel (PDM).

Framework, Organisation, and Implementation metaclass

Figure 46 shows the PDM containing the metaclasses Framework, Organisation, and Implementation. Organisation contains all the input elements which are the situational factors (Sector, Employee range, Implementation focus) of the organisation. The overall Framework exists always of exactly one Organisation, and the Organisation always consists of exactly one Implementation. The attributes are further specified in Table 19.

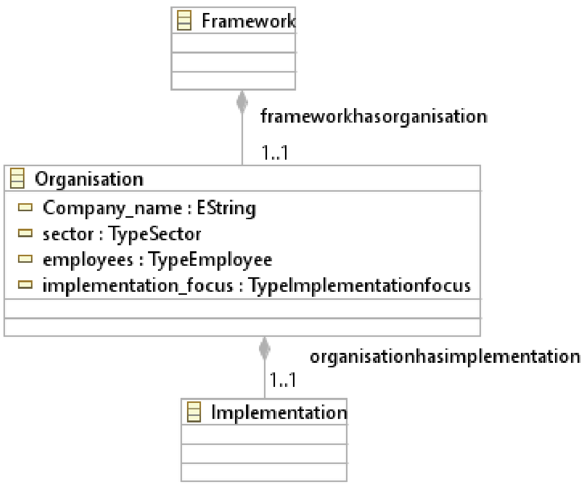


Figure 46 Portion of the PDM containing the metaclasses Framework, Organisation, and Implementation

Table 19 shows the attributes of the Framework, Organisation, and Implementation metaclasses.

Table 19 Attributes of the metaclasses Framework, Organisation, and Implementation

Element:	**Type:**	**Description:**
Frameworkhasorganisation	Aggregation	The Framework always consists of exactly 1 Organisation.
Company_name	Estring	This element represents the Company name as an input field.
Sector	TypeSector	This element represents the Sector in which an specific organisation operates in.
Employees	TypeEmployee	This element represents the number of Employees working at a specific organisation.
Implementation_focus	TypeImplementationfocus	This element represents the focus of the BRMS implementation of a specific organisation.
organisationhasimplementation	Aggregation	The Organisation always consist of exactly 1 Implementation.

DesignProblem metaclass

The DesignProblem metaclass represents 25 design problems which could exists as a reason to implement a BRMS (as shown in Figure 47).

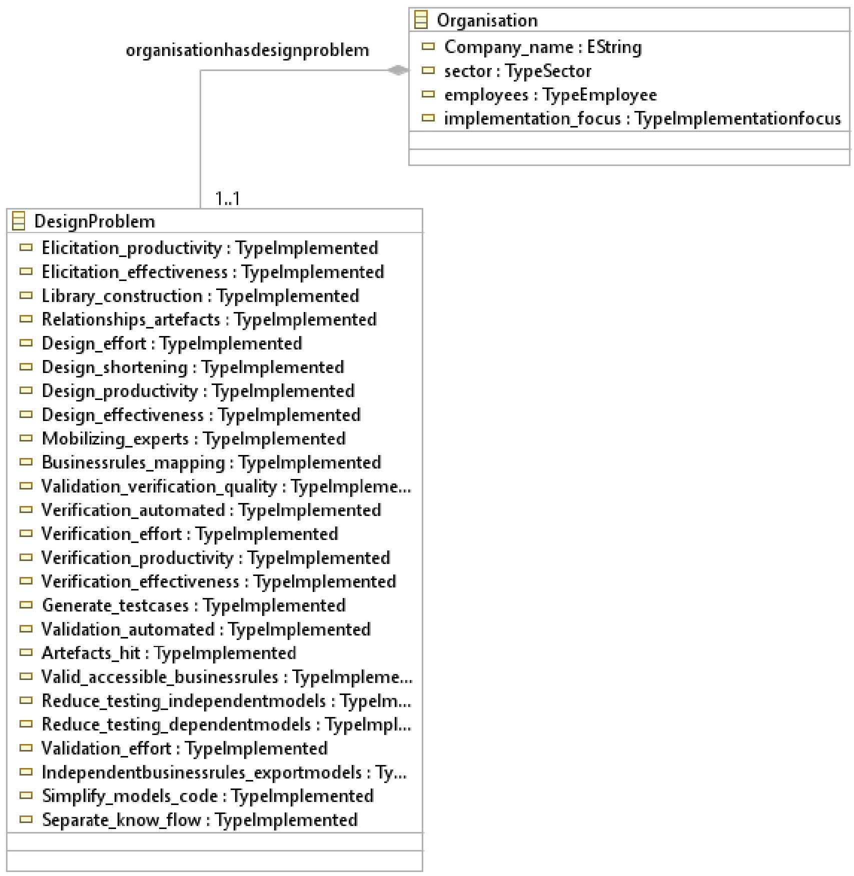


Figure 47 Portion of the PDM containing the metaclass DesignProblem

Table 20 shows the attributes of the DesignProblem metaclass.

Table 20 Attributes of the metaclass DesignProblem

Element:	**Type:**	**Description**
Elicitation_productivity	TypeImplemented	This element indicates Design Problem #1: Low productivity of elicitation.
Elicitation_effectiveness	TypeImplemented	This element indicates Design Problem #2: Low effectiveness of elicitation.
Library_construction	TypeImplemented	This element indicates Design Problem #3: The construction of a library of decisions.
Relationships_artefacts	TypeImplemented	This element indicates Design Problem #4: No or low insight into relationships between artefacts.

Design_effort	TypeImplemented	This element indicates Design Problem #5: Reducing effort in design (requirements and specifications).
Design_shortening	TypeImplemented	This element indicates Design Problem #6: Shortening the design phase.
Design_productivity	TypeImplemented	This element indicates Design Problem #7: Low productivity in the design phase.
Design_effectiveness	TypeImplemented	This element indicates Design Problem #8: Low effectiveness in design.
Mobilizing_experts	TypeImplemented	This element indicates Design Problem #9: Mobilizing experts.
Businessrules_mapping	TypeImplemented	This element indicates Design Problem #10: Business rules mapping as much as possible.
Validation_verification_quality	TypeImplemented	This element indicates Design Problem #11: Quality assurance validation and verification.
Verification_automated	TypeImplemented	This element indicates Design Problem #12: Automated verification of business rules.
Verification_effort	TypeImplemented	This element indicates Design Problem #13: Reducing efforts at verification.
Verification_productivity	TypeImplemented	This element indicates Design Problem #14: Low productivity of verification.
Verification_effectiveness	TypeImplemented	This element indicates Design Problem #15: Low effectivity of verification.
Generate_testcases	TypeImplemented	This element indicates Design Problem #16: Generate automated test cases.
Validation_automated	TypeImplemented	This element indicates Design Problem #17: Automated testing with validation.
Artefacts_hit	TypeImplemented	This element indicates Design Problem #18: No insight which artefacts are hit.
Valid_accessible_businessrules	TypeImplemented	This element indicates Design Problem #19: Validated and accessible business rules.
Reduce_testing_independentmodels	TypeImplemented	This element indicates Design Problem #20: Reducing testing for implementation independent models.
Reduce_testing_dependentmodels	TypeImplemented	This element indicates Design Problem #21: Reducing testing for implementation-dependent models.
Validation_effort	TypeImplemented	This element indicates Design Problem #22: Reducing effort for validation.
Independentbusinessrules_exportmodels	TypeImplemented	This element indicates Design Problem #23: Working with implementation independent business rules to export models.
Simplify_models_code	TypeImplemented	This element indicates Design Problem #24: Simplify converting from models into code.

Separate_know_flow	TypeImplemented	This element indicates Design Problem #25: Separating of implementation 'know' and 'flow'.
organisationhasdesignproblem	Aggregation	The organisation always consists of exactly 1 Design Problem, which in its turn exists of 25 different elements.

MethodFragments metaclass

The MethodFragments metaclass represents a configuration of 1 – 44 method fragments which needed to be implemented to solve a specific design problem (as shown in Figure 48)

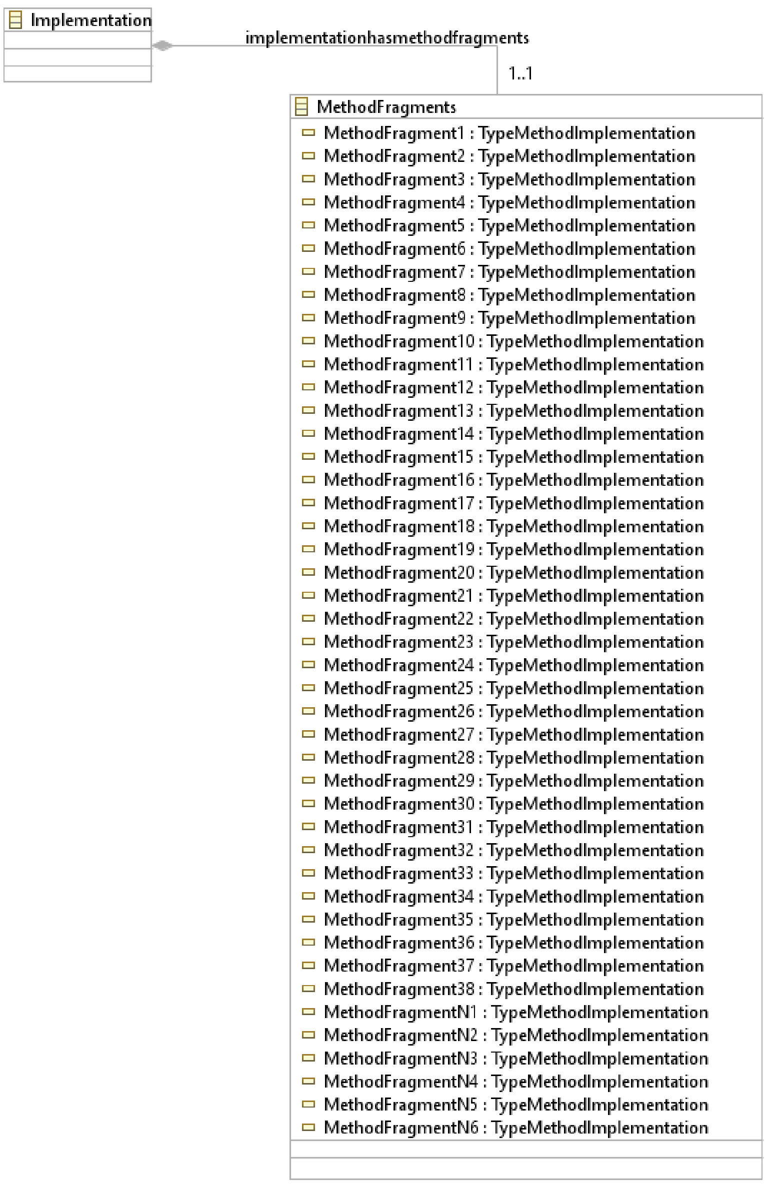


Figure 48 Portion of the PDM containing the metaclass MethodFragments

Table 21 shows the attributes of the MethodFragments metaclass.

Table 21 Attributes of the metaclass MethodFragments

Element:	Type:	Description
MethodFragment1	TypeMethodImplementation	This element represents MethodFragment1.
MethodFragment2	TypeMethodImplementation	This element represents MethodFragment2.
MethodFragment3	TypeMethodImplementation	This element represents MethodFragment3.
MethodFragment4	TypeMethodImplementation	This element represents MethodFragment4.
MethodFragment5	TypeMethodImplementation	This element represents MethodFragment5.
MethodFragment6	TypeMethodImplementation	This element represents MethodFragment6.
MethodFragment7	TypeMethodImplementation	This element represents MethodFragment7.
MethodFragment8	TypeMethodImplementation	This element represents MethodFragment8.
MethodFragment9	TypeMethodImplementation	This element represents MethodFragment9.
MethodFragment10	TypeMethodImplementation	This element represents MethodFragment10.
MethodFragment11	TypeMethodImplementation	This element represents MethodFragment11.
MethodFragment12	TypeMethodImplementation	This element represents MethodFragment12.
MethodFragment13	TypeMethodImplementation	This element represents MethodFragment13.
MethodFragment14	TypeMethodImplementation	This element represents MethodFragment14.
MethodFragment15	TypeMethodImplementation	This element represents MethodFragment15.
MethodFragment16	TypeMethodImplementation	This element represents MethodFragment16.
MethodFragment17	TypeMethodImplementation	This element represents MethodFragment17.
MethodFragment18	TypeMethodImplementation	This element represents MethodFragment18.
MethodFragment19	TypeMethodImplementation	This element represents MethodFragment19.
MethodFragment20	TypeMethodImplementation	This element represents MethodFragment20.
MethodFragment21	TypeMethodImplementation	This element represents MethodFragment21.

MethodFragment22	TypeMethodImplementation	This element represents MethodFragment22.
MethodFragment23	TypeMethodImplementation	This element represents MethodFragment23.
MethodFragment24	TypeMethodImplementation	This element represents MethodFragment24.
MethodFragment25	TypeMethodImplementation	This element represents MethodFragment25.
MethodFragment26	TypeMethodImplementation	This element represents MethodFragment26.
MethodFragment27	TypeMethodImplementation	This element represents MethodFragment27.
MethodFragment28	TypeMethodImplementation	This element represents MethodFragment28.
MethodFragment29	TypeMethodImplementation	This element represents MethodFragment29.
MethodFragment30	TypeMethodImplementation	This element represents MethodFragment30.
MethodFragment31	TypeMethodImplementation	This element represents MethodFragment31.
MethodFragment32	TypeMethodImplementation	This element represents MethodFragment32.
MethodFragment33	TypeMethodImplementation	This element represents MethodFragment33.
MethodFragment34	TypeMethodImplementation	This element represents MethodFragment34.
MethodFragment35	TypeMethodImplementation	This element represents MethodFragment35.
MethodFragment36	TypeMethodImplementation	This element represents MethodFragment36.
MethodFragment37	TypeMethodImplementation	This element represents MethodFragment37.
MethodFragment38	TypeMethodImplementation	This element represents MethodFragment38.
MethodFragmentN1	TypeMethodImplementation	This element represents MethodFragment N1.
MethodFragmentN2	TypeMethodImplementation	This element represents MethodFragment N2.
MethodFragmentN3	TypeMethodImplementation	This element represents MethodFragment N3.
MethodFragmentN4	TypeMethodImplementation	This element represents MethodFragment N4.
MethodFragmentN5	TypeMethodImplementation	This element represents MethodFragment N5.
MethodFragmentN6	TypeMethodImplementation	This element represents MethodFragment N6.

implementationhasmethodfragments	Aggregation	The implementation always consists of exactly 1 Method Fragment, which in its turn consist of 44 different elements.

Implementation, ProblemClass, and DesignSituation metaclass

The Implementation, ProblemClass, and DesignSituation metaclasses represents a number of problem classes (1-9) with every problem classes a design situation which should be implemented to solving a specific set of design problems (as shown in Figure 49).

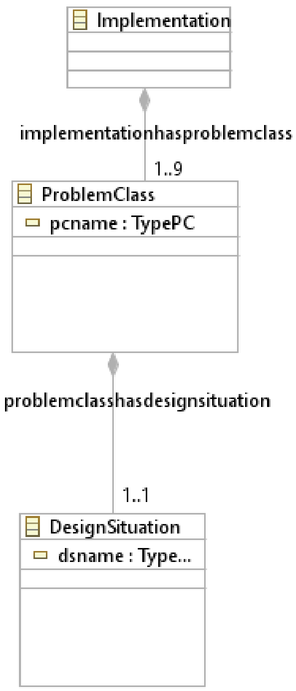


Figure 49 Portion of the PDM containing the metaclasses Implementation, ProblemClass, and DesignSituation

Table 22 shows the attributes of the Implementation, ProblemClass, and DesignSituation metaclasses.

Table 22 Attributes of the metaclasses Implementation, ProblemClass, and DesignSituation

Element:	Type:	Description
Implementationhasproblemclass	Aggregation	The implementation always has 1 or a maximum of 9 problem classes.
Pcname	TypePC	This element represents the Problem Class which is included into a BRMS implementation.
Problemclasshasdesignsituation	Aggregation	The Problem Class always has exactly 1 Design Situation.
Dsname	TypeDS	This element represent the Desing Situation which is included into a BRMS implementation.

TypePC enumeration metaclass

The TypePC metaclass represents all the nine problem classes (as shown in Figure 50) of the BRM problem space.

Figure 50 Portion of the PDM containing the metaclass TypePC

Table 23 shows the attributes of the TypePC metaclass.

Table 23 Attributes of the metaclass TypePC

Element:	**Description:**
Elicitation	This element represents the elicitation problem class.
Design	This element represents the design problem class.
Specification	This element represents the specification problem class.
Verification	This element represents the verification problem class.
Validation	This element represents the validation problem class.
Deployment	This element represents the deployment problem class.
Execution	This element represents the execution problem class.
Monitoring	This element represents the monitoring problem class.
Governance	This element represents the governance problem class.

TypeSector enumeration metaclass

The TypesSector metaclass represents all the sectors (Public and Financial) included into the BRMS implementation framework (as shown in Figure 51).

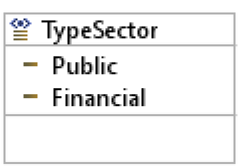


Figure 51 Portion of the PDM containing the metaclass TypeSector

Table 24 shows the attributes of the TypeSector metaclass.

Table 24 Attributes of the metaclass TypeSector

Element:	Description:
Public	This element represents the Sector Public.
Financial	This element represents the Sector Financial.

TypeImplemented enumeration metaclass

The TypeImplemented metaclass represents if a Design Problem is present (Yes) in a specific organisation or when a Design Problem is not present (No) in a specific organisation (as shown in Figure 52).

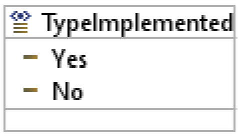


Figure 52 Portion of the PDM containing the metaclass TypeImplemented

Table 25 shows the attributes of the TypeImplemented metaclass.

Table 25 Attributes of the metaclass TypeImplemented

Element:	Description:
Yes	This elements represents the Yes for when a Design Problem is present at a specific organisation.
No	This elements represents the No for when a Design Problem is not present at a specific organisation.

TypeEmployee enumeration metaclass

The TypeEmployee metaclass represents the number of employees (<50…>5000) working at a specific organisation (as shown in Figure 53).

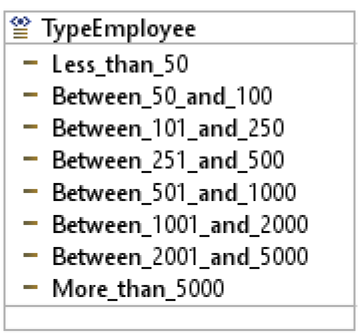


Figure 53 Portion of the PDM containing the metaclass TypeEmployee

Table 26 shows the attributes of the TypeEmployee metaclass.

Table 26 Attributes of the metaclass TypeEmployee

Element:	Description:
Less_than_50	This element represents the employee range <50.
Between_50_and_100	This element represents the employee range 50 – 100.
Between_101_and_250	This element represents the employee range 101 – 250.
Between_251_and_500	This element represents the employee range 251 – 500.
Between_501_and_1000	This element represents the employee range 501 – 1000.
Between_1001_and_2000	This element represents the employee range 1001 – 2000.
Between_2001_and_5000	This element represents the employee range 2001 – 5000.
More_than_5000	This element represents the employee range >5000.

TypeDS enumeration metaclass

The TypeDS metaclass represents the Design Situations which represents a design situation for a problem class which is included into a BRMS implementation for an organisation (as shown in Figure 54).

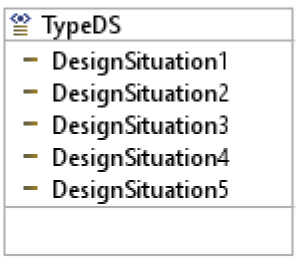


Figure 54 Portion of the PDM containing the metaclass TypeDS

Table 27 shows the attributes of the TypeDS metaclass

Table 27 Attributes of the metaclass TypeDS

Element:	Description:
DesignSituation1	This element represents a design situation for a specific problem class which is included in BRMS implementation for a specific organisation.
DesignSituation2	This element represents a design situation for a specific problem class which is included in BRMS implementation for a specific organisation.
DesignSituation3	This element represents a design situation for a specific problem class which is included in BRMS implementation for a specific organisation.
DesignSituation4	This element represents a design situation for a specific problem class which is included in BRMS implementation for a specific organisation.
DesignSituation5	This element represents a design situation for a specific problem class which is included in BRMS implementation for a specific organisation.

TypeDS enumeration metaclass

The TypeDS metaclass represents the focus (Application, Line of business, or Organisation wide focused) of the BRMS implementation of the organisation (as shown in Figure 55).

Figure 55 Portion of the PDM containing the metaclass TypeImplementationfocus

Table 28 shows the attributes of the TypeImplementationfocus metaclass.

Table 28 Attributes of the metaclass TypeImplementationfocus

Element:	**Description:**
Application_focused	This element represents that the BRMS implementation of a specific organisation is Application focused.
Line_of_business_focused	This element represents that the BRMS implementation of a specific organisation is Line of business focused.
Organisation_wide_focused	This element represents that the BRMS implementation of a specific organisation is Organisation wide focused.

TypeMethodImplementation enumeration metaclass

The TypeMethodImplementation represents if a specific Method Fragment needs to be implemented (Implement) or not implemented (Not_implement) when having a certain configuration of Design Problems (as shown in Figure 56).

Figure 56 Portion of the PDM containing the metaclass TypeMethodImplementation

Table 29 shows the attributes of the TypeMethodImplementation metaclass.

Table 29 Attributes of the metaclass TypeMethodImplementation

Element:	**Description:**
Implement	This element represents that a specific Method fragment should be implemented to solve a specific Design problem.
Not_implement	This element represents that a specific Method fragment should not be implemented to solve a specific Design problem.

Appendix I – Expert interview protocol

Interview protocol

Interviewee name:

Interviewee position:

Interviewee experience:

Interview timestamp:

Interview duration:

Interviewee experience

Interviewee experience in the BRM(S) field (This is to validate the actual statement of being an expert, otherwise no EXPERT interview could be conducted).

Elements

The focus of validating the BRMS framework and its elements lies on the QA correctness. Some elements of the framework are output of statistical analysis, these specific elements are excluded.

These elements are:

- The configuration of design situations
- The existing set of design factors and its values

Introduction

Defining BRM and BRMS

BRM: Business Rules Management (BRM) is the practice of managing business rules centered around the activities of elicitation, design, specification, verification, validation, deployment, execution, governance, or the monitoring of business rules.

BRMS: Business Rules Management Solution (BRMS) enables organizations to elicitate, design, manage and execute business rules and is a co-creation of the BRM activities described earlier.

Triggers (valid or not?)

Organisational and technical implementation:

The scientific world sees many opportunities in BRM, but the BRM topic is not over-researched. This is especially the case for the technical implementation of a business rules management solution. In addition, there is a lack of research in the arena of organizational implementation of BRMS. Organizational implementation of BRMS is focused on the organizational aspects touched when implementing a software product.

MENTIONING: This research is focused on the contradiction of the singular problem orientation.

Problem Classes

Previous research contradicts the singular problem orientation and proposed that different BRMS have a common design problem. Common design problems indicate that problem classes exist. Therefore, design solutions can be created.

- Elicitation
- Design
- Specification
- Verification
- Validation
- Deployment
- Execution
- Monitoring
- Governance

Design factors

What are design factors?

			Problem classes:										
ID:	**Design factor:**	**Value:**	**Elicitation**	**Design**	**Specification**	**Verification**	**Validation**	**Deployment**	**Execution**	**Monitoring**	**Governance**	**Correct:**	**Comments:**
1	Sector implementation	Public or Financial	X	X	X	X	X	X	X	X	X		
2	Employee range	< 50, 50 - 100, 101 - 250, 251 - 500, 501 - 1000, 1001 - 2000, 2001 - 5000, or >5000	X	X	X	X	X	X	X	X	X		
3	Scope focus	Application focused, Line of business focused, or Organisation wide focused	X	X	X	X	X	X	X	X	X		
4	Capability leader	IT, Business, or Central IT/Business group		X	X			X	X	X	X		
5	Capability autonomy	Autonomy level 1 - 10	X	X	X	X	X	X	X	X	X		

ID:	Problem class:	Design factor:	Value:	Description:	Correct:	Comments:
6	Elicitation	Elicitation source (Data)	Yes/No	Database data is used as an source for elicitation.		
7	Elicitation	Relevant set of selected sources	Yes/No	The output of elicitation is a relevant set of selected sources.		
8	Elicitation	Scenario analysis	Yes/No	Scenario analysis is used during elicitation. Elicitation based on business scenario's.		
9	Design	Velocity design	Yes/No	Velocity is one of the 5v's and represents the speed at which the decision is to be taken.		
10	Design	Volume design	Yes/No	Volume is one of the 5v's and represents the number of times a decision is taken in a given time period.		
11	Design	Veracity design	Yes/No	Veracity is one of the 5v's and represents the accuracy at which the decision is to be taken.		
12	Design	Role: Agree	Yes/No	The Agree (RAPID) role is defined in the design problem class.		
13	Design	Role: Perform	Yes/No	The Perform (RAPID) role is defined in the design problem class.		
14	Design	Role: Input	Yes/No	The Input (RAPID) role is defined in the design problem class.		
15	Design	Role: Recommend	Yes/No	The Recommend (RAPID) role is defined in the design problem class.		
16	Design	Impact analysis	Yes/No	An impact analysis is checks the impact that a certain change brings forth		
17	Specification	Rule specification	In models or in text	Rules are specified in either models or in text.		
18	Specification	Unambiguous specification	Yes/No	Unambiguous specification is that the language in which the business rules are written can automatically be transformed to a business rule in a rule engine.		

19	Verification	Detection degree	Automated - Detective Automated - Preventive Manual - Detective Manual – Preventive	Detection degree indicates at which degree syntax and semantic errors are detected.		
20	Validation	Peer review	Yes/No	Peer reviews are conducted to validate the expected behaviour of the business rules.		
21	Validation	Scenario validation	Yes/No	Scenario validation is conducted to validate the expected behaviour of the business rules.		
22	Validation	Source validation	Yes/No	Source validation is conducted to validate the expected behaviour of the business rules.		
23	Validation	Quality attribute: Traceability	Yes/No	The validation capability checks on the quality attribute traceability.		
24	Validation	Quality attribute: Accuracy	Yes/No	The validation capability checks on the quality attribute accuracy.		
25	Execution	Gaming	Yes/No	Gaming is made possible for the purpose of testing.		
26	Execution	Rules saved	Yes/No	The executed rules are saved in the execution capability.		
27	Monitoring	Verification correction frequency	Constantly Daily Weekly Monthly Quarterly Yearly Not evaluated	The frequency of corrections per selected context design emerging from the verification process.		
28	Monitoring	Specific Verification correction frequency		The frequency of corrections per selected context design, emerging from the verification process, per business analyst and per type of verification error.		
29	Monitoring	Validation correction frequency		The frequency of corrections per selected context design emerging from the validation process per complexity level of a business rule.		
30	Monitoring	Specific validation correction frequency		The frequency of corrections per selected context design emerging from the validation process per type of validation error.		

31	Monitoring	Design correction frequency		The frequency of corrections per selected context architecture emerging from the design process per scope design.		
32	Monitoring	Context design frequency		The frequency of instantiations per selected context design.		
33	Monitoring	Validation error frequency		The frequency per selected type of a validation error.		
34	Monitoring	Verification error frequency		The frequency per selected type of a verification error.		
35	Monitoring	Time worked on business rule		The number of time units required to define, verify, and validate a single business rule.		
36	Monitoring	Deviation frequency		The frequency of deviations between an implementation dependent context design and an implementation independent context design.		
37	Monitoring	Dependent execution frequency		The frequency of executions of an implementation dependent business rule.		
38	Monitoring	Scope design variant frequency		The frequency of execution variants of a scope design.		
39	Monitoring	Time required execution		The number of time units required for the execution per execution variant.		
40	Monitoring	Non automated business rules		The amount of business rules that cannot be automated.		
41	Governance	Traceability	Yes/No	Traceability ensures the possibility to trace created elements to their corresponding laws and regulations. Furthermore, the traceability capability creates a foundation for impact analysis when, for example, new laws are needed to be processed into value propositions.		

Non-characterising design factors

ID:	Problem class:	Design factor:	Value:	Description:	Valid:	Comments:
42	Elicitation	Elicitation source (People)	Yes	People are used as a source for elicitation.		
43	Elicitation	Elicitation source (Documents)	Yes	Documents are used as a source for elicitation.		
44	Elicitation	Source analysis	Yes	Source analysis is used during elicitation. Analysis based on sources.		
45	Elicitation	Impact analysis	Yes	An impact analysis is checks the impact that a certain change brings forth.		
46	Design	Value design	Yes	Value is one of the five V's and represents the value of the decision to the organisation.		
47	Design	Variety design	Yes	Variety is one of the five V's and represents the variety of execution paths of a decision which is to be taken.		
48	Design	Role: Decide	Yes	The Decide (RAPID) role is defined in the design problem class.		
49	Validation	Completeness	Yes	The validation capability checks on the quality attribute completeness.		
50	Validation	Usability	No	The validation capability checks on the quality attribute usability.		
51	Execution	Input saved	Yes	The input is saved in the execution capability.		
52	Execution	Output saved	Yes	The output is saved in the execution capability.		
53	Governance	Version management	Yes	The purpose of the version management capability is capturing and keeping track of regarding the elements which are created of modified in the other eight capabilities.		
54	Governance	Validity management	Yes	The purpose of validity management is to create the possibility to provide a specific version of a value proposition, at any given moment of time.		

Design situations

(THESIS)

Design problems

Design factors solving design problems (Excel)

Additional Comments:

Framework

Missing:

Not necessary:

Strong:

Weak: