Context: Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the value networks in which the enterprise acts. This can be supported with the help of intent-driven systems. The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of them. Only then are different machine actors able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement.

Objective: When building a business support system it is critical to separate the business model of the business support system itself from the business models used by the enterprise which is using the business support system. The core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents. The business studio is a software that supports the realization of business models used by the enterprise by configuring the capabilities provided by the business support system. The aim is to find out how we can support the design of a business studio which is based on intent-driven systems.

Method: We are using the design science framework as our research framework. During our design science study we have used the following research methods: systematic literature review, case study, quasi-experiment, and action research.

Results: We have produced two design artifacts as a start to be able to support the design of a business studio. These artifacts are the models and quasi-experiment in Chapter 3, and the action research in Chapter 4. The models found during the case study have proved to be a valuable artifact for the stakeholder. The results from the quasi-experiment and the action research are seen as new problem solving knowledge by the stakeholder.

Conclusion: The synthesis shows a need for further research regarding semantic interchange of information, actor interaction in intent-driven systems, and the governance of intent-driven systems.
Towards Intent-Driven Systems

Johan Silvander
Towards Intent-Driven Systems

Johan Silvander

Doctoral Dissertation in Computer Science
Towards Intent-Driven Systems

Johan Silvander
Abstract

Context: Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the value networks in which the enterprise acts. This can be supported with the help of intent-driven systems. The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of them. Only then are different machine actors able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement.

Objective: When building a business support system it is critical to separate the business model of the business support system itself from the business models used by the enterprise which is using the business support system. The core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents. The business studio is a software that supports the realization of business models used by the enterprise by configuring the capabilities provided by the business support system. The aim is to find out how we can support the design of a business studio which is based on intent-driven systems.

Method: We are using the design science framework as our research framework. During our design science study we have used the following research methods: systematic literature review, case study, quasi experiment, and action research.

Results: We have produced two design artifacts as a start to be able to support the design of a business studio. These artifacts are the models and quasi-experiment in Chapter 3, and the action research in Chapter 4. The models found during the case study have proved to be a valuable artifact for the stakeholder. The
results from the quasi-experiment and the action research are seen as new problem solving knowledge by the stakeholder.

Conclusion: The synthesis shows a need for further research regarding semantic interchange of information, actor interaction in intent-driven systems, and the governance of intent-driven systems.
Acknowledgments

To everyone who have contributed to the one I am today, for good and for worse.
No one mentioned and no one forgotten.
Preface

Papers in This Thesis

This compilation thesis includes the following five papers 1.


Chapter 5: Johan Silvander and Mikael Svahnberg, “Towards Executable Business Rules”, Intended for a conference (Accepted as an appendix to Chapter 3).


Contribution Statement

Johan Silvander is the lead author of all the papers in this thesis. As a lead author, he took the main responsibility in designing the studies, collecting and analyzing data, and reporting the findings in peer-reviewed publications. Furthermore, he

---

1Papers marked with “Under Review” are currently considered for publication. Papers marked with “Under Submission” have been submitted but were not reviewed yet at the time of writing.
is the sole author of Chapter 1, the overview. The co-authors’ contribution are described below.

Chapter 2: Mikael Svalnberg contributed with valuable methodology support during the whole study. He reviewed and commented on intermediate versions and the final draft off the paper.

Chapter 3: Magnus Wilson contributed with valuable comments during the analysis of the results. Magnus Wilson, Krzysztof Wnuk and Mikael Svalnberg reviewed and commented on intermediate versions and the final draft off the paper.

Chapter 4: Magnus Wilson wrote parts of the Introduction and Background sections. Magnus Wilson and Krzysztof Wnuk reviewed and commented on intermediate versions and the final draft off the paper.

Chapter 5: Mikael Svalnberg reviewed and commented on intermediate versions and the final draft off the paper.

Chapter 6: Mikael Svalnberg reviewed and commented on intermediate versions and the current draft off the paper.

Related Papers Not Included in This Thesis

- Magnus Wilson, Krzysztof Wnuk, Johan Silvander and Tony Gorschek, “Towards effective and efficient business modeling - A systematic literature review” journal (Under Submission).

Other Papers Not Included in This Thesis


Funding

The research presented in this paper was partly funded by the Swedish Knowledge Foundation, the project “Professional Licentiate of Engineering Research School”, conducted at Blekinge Institute of Technology, Software Engineering Research Lab - Sweden.
# Contents

Abstract i  
Acknowledgments iii  
Preface iv  

1 Overview  
1.1 Introduction .................................................. 1  
1.1.1 The use of Pask’s conversation theory ................. 3  
1.1.2 Background and related work ......................... 5  
1.1.3 Research questions ...................................... 8  
1.2 Methodology .................................................. 9  
1.2.1 Research settings and data collection .................. 9  
1.2.2 Limitations ............................................... 11  
1.3 Research Framework and Research Execution ............. 11  
1.3.1 Research framework ..................................... 11  
1.3.2 Research execution ..................................... 13  
1.4 Conclusion and Further Research .......................... 16  
1.4.1 Conclusion .............................................. 16  
1.4.2 Produced design artifacts .............................. 17  
1.4.3 Suggested new design problems ....................... 18  

2 A Systematic Literature Review on Intent-Driven Systems 20  
2.1 Introduction .................................................. 21  
2.1.1 Background ............................................... 24  
2.1.2 Research questions ..................................... 25  
2.1.3 Summary ................................................. 27  
2.2 Review Methodology ........................................ 28  
2.2.1 Pilot study .............................................. 28
6 Uncovering Implicit Rules in Medicine Diagnosis

6.1 Introduction ........................................ 84
6.2 Methodology ........................................ 85
6.3 Results .............................................. 86
6.4 Analysis .............................................. 89
6.5 Discussion ........................................... 90
6.6 Conclusion and Future Work .......................... 91

A Appendix to Chapter 2

A.1 Investigating the Need for a Systematic Literature Review .... 93
A.2 Pilot Study .......................................... 94
A.3 Data Extraction Protocol ................................ 95
  A.3.1 Data extraction protocol for RQ1:1-RQ1:6 ............... 95
  A.3.2 Data extraction for P1 ................................ 95
  A.3.3 Data extraction for P2 ................................ 95
A.4 Comments on Primary Studies ............................ 98

B Appendix to Chapter 3

B.1 Interview Instrument .................................... 103
B.2 Study Execution ........................................ 104
  B.2.1 Phase 1: Initial meeting (February 2015) .............. 104
  B.2.2 Phase 2: Defining Why & What (February 2015) ....... 104
  B.2.3 Phase 3: External review (March 2015) ............... 105
  B.2.4 Phase 4: Defining How (April 2015) .................. 105
  B.2.5 Phase 5: Context frame (April 2015) ................. 105
  B.2.6 Phase 6: Presentation to Ericsson research (May 2015) 106
  B.2.7 Phase 7: Composite context frame & value networks (June 2015) .............................................. 106
  B.2.8 Phase 8: The model in Figure 3.6 (August 2015) ....... 106
  B.2.9 Phase 9: Business interactions (September 2015) .... 106
  B.2.10 Phase 10: Verification of changes to the information due to review comments (October 2016) .................. 107
B.3 Proof-of-concept ....................................... 108
  B.3.1 Introduction ....................................... 108
  B.3.2 Methodology ....................................... 109
Chapter 1

Overview

1.1 Introduction

Knowledge creation in enterprises is obtained through interactions between actors [68]. Intents are the aim and purpose resulting from knowledge creation regarding internal or external influences. The SECI-model [68] and Pask’s conversation theory [78] are models which can be used for knowledge creation when enterprises are interacting with each other.

We define intent as a subject or type of possible behavior, i.e. something that can be interpreted to have significance. Any actor can have intents. When an actor publicly declare an intent in a certain context, it becomes a stated intent. A stated intent could be a declaration of capabilities an actor promise to provide, or a requirement an actor try to impose on another actor. A true cooperation between actors is based on the capabilities promised by the interacting actors. A decision about whether an actor has kept its promise or not, can be done by the promising actor as well as any actor which is observing the behavior of the promising actor.

Definition 1.1: Our definition of intent

Today’s enterprises are part of value networks. A value network refers to “A set of connections between organizations and/or individuals interacting with each other to benefit the entire group [45]”. Enterprises in a value network can be seen as parts in a compositional system and are by themselves compositional
Chapter 1. Overview

systems, also known as a system of systems. Compositionality refers to “The evident ability of humans to represent entities as hierarchies of parts, with these parts themselves being meaningful entities, and being reusable in a near infinite assortment of meaningful combinations [39]”. Cyber-physical systems can be seen as a vital part of a compositional system.

The construction of a compositional system requires methods to achieve a holistic collective benefit through the individual systems’ participation and cooperation when each system adopts a solution that maximizes its own self-interest [29]. The construction of such a system has to support changes of a system’s policies and rules in a way that are effective and efficient [33]. Achieving the changes in an effective and efficient way requires knowledge about the affected intents and the correlation between intents. The correlations between intents are affected by factors like value network structure, decision process and the actors’ responsibilities.

The actors in a compositional system may be humans or machines. By using software agents as machine actors enterprises can bring customers closer to suppliers of products and services, support their continued demand for change, inject further intelligence into enterprises and simplify the environment for both customers and employees [3]. Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the compositional systems the enterprise acts in.

An actor’s intent has to be communicated to other actors. During the interaction about the intent the actors have to prove their understanding of the intent in order to gain a common understanding and knowledge about the intent. This interaction can continue in several steps and might re-shape the original intent. Together with Ericsson we are using Pask’s conversation theory [78] as a model to describe intent-driven systems.

The remaining sections in the overview of this thesis is organized as follows. In Section 1.1.1 we present how we are using Pask’s conversation theory, followed by the presentation of the background and related work in Section 1.1.2. The research questions are presented in Section 1.1.3. The research methodology is presented in Section 1.2, followed by a presentation of the research framework, and the execution of the research in Section 1.3. Finally, in Section 1.4, the conclusion and further research are presented.
We define an intent-driven system as a compositional system of actors where the actors declare, negotiate, and asses intents made by actors. The intents can be declared, negotiated, or assessed, on-behalf of an actor’s stakeholder or an actor’s self-interest.

The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of them. Only then are different machine actors able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents.

Definition 1.2: Our definition of intent-driven systems

1.1.1 The use of Pask’s conversation theory

The minimal structure of the conversation theory is shown in Figure 1.1. Figure 1.1 illustrates how two actors (A and B) interact on the same domain D with the help of a common language L. In order to construct, express, and validate a topic, the two layers of procedures with feedback loops (FB) and feedforward loops (FF) exist. The P(0) layer procedures operate upon the domain in order to bring about or explain topic relations. The P(1) layer operate on P(0) procedures in order to construct or reconstruct them.

Figure 1.1: The minimal structure of the conversation theory.

Two key components in Pask’s conversation theory [78] are language and domain. A language L is defined as:
Chapter 1. Overview

“L may be a natural, written or spoken, symbolic language, but it need not be. It may be a system of symbolic behaviors such as dance or actions such as key pressing. It may be formalized, as in, mathematics and higher level programming languages, but it need not be. It must however have many of the qualities of a natural language, with possibilities to express and interpret commands, questions, answers, obediences, explanations, or descriptions.” [56].

A domain could be described as:

“A domain is a collection of topic’s, and a topic is essentially a relation. This may be a very concrete relation (a relation between alphabetic characters and the keyboard positions in typing) or it may be an abstract relation (a relation between smugglers and the countries they operate in): To learn or solve a problem is to ‘bring about’ such a topic relation.” [56].

Pask [78] stresses the fact that the different actors have obtained their specific domain information through several different interactions which makes the model recursive. This means that each actor obtains its specific domain information in different contexts. We define a context frame as the total domain information for a specific domain an actor has obtained. A context description is what gives a context frame a scope (boundary) and defines a meaning (semantics).

The idea of an intent-driven system as chains of interacting loops with one or several cycles is shown in Figure 1.2. Figure 1.2 shows how two human actors (A and B) interacts about an intent, using language L regarding domain D. The interaction results in some sort of common understanding. Each human actor interacts with its respective machine actor to translate the intent from domain D to domain $D'$, using language $L_{hm}$. The two machine actors ($A'$ and $B'$) interact on the intent, using Language $L'$ regarding domain $D'$, and the outcome is fed back to their respective human actor, using language $L_{mh}$. One or both of the human actors may not be satisfied with the outcome. This will render the start of a new cycle. This indicates that actor interaction between several actors of different types are vital for intent-driven systems. The languages are built upon semantics and a collaboration between actors might need a interchange of semantics. Since an intent-driven system is a compositional system, governance is needed to guide the different actors to achieve a holistic collective benefit.

When building a business support system it is critical to separate the business model of the business support system itself from the business models used by the enterprise which is using the business support system. The business studio will support the realization of business models used by the enterprise by configure the capabilities provided by the business support system.
Chapter 1. Overview

The idea of a business studio is shown in Figure 1.3. Figure 1.3 shows how two human actors (A and B) interacts with a business studio, which is the dotted part in the figure. The human actor A represents the enterprise using the business support system an the human actor B represents the owner of the business support system. Actor A and actor B interact using language L to formulate common understanding of the needed capabilities. Actor B interacts with the component (B’) in the business support system to configure basic capabilities according to the understanding obtained during the interaction with actor A. The interactions between actor B and component B’ are regarded as a configuration of the business studio, and is not part of the business studio itself. Actor A interacts with component A’ to obtain its business intents. Component A’ is responsible to translate the business intents and composing the needed functionality with the help of the basic capabilities (B’). Since the business studio is not a monolithic actor, and there are more actors than one in an enterprise, the business studio will be a composition of what is described in Figure 1.2.

1.1.2 Background and related work

The research project is done in collaboration with Ericsson and is based on the design science methodology [42]. At the time of the project, Ericsson was in an early pre-study phase of a business studio. The business studio is part of a business support system and will act in the area of planning and monitoring business intents. The idea with the business studio is to deliver support for a 360-degree view of an enterprise’s business. The view includes both the actual execution of an enterprise’s business and the intended changes to this execution.
Chapter 1. Overview

The business studio will support and govern the decisions and actions needed to maintain or change the way an enterprise does its business. Ericsson’s customers will be able to buy the business studio as a product or a service. With the help of the business studio, Ericsson’s customers will get support and knowledge about how to configure, monitor, and redesign their products and business. The software could be used in different business areas, for example charging, billing, customer relationship management, partner relationship management, product management, order management, etc. The business studio supports the idea of continuous business-requirement engineering. Intent-driven systems are one of the cornerstones for the business studio product.

The business intent realization builds upon collaborations in the form of interactions between the different actors. The interactions between the actors are negotiated since each actor has its own view of a business intent. The negotiation may result in a, to some extent, desired outcome. The outcome of an interaction between human actors results in conclusions stated in natural language. From a software engineering perspective the conclusions can be made executable in the form of policies and rules but this would require Natural Language Processing, suggested in [110], and a formal way of expressing the policies and rules, e.g. using Semantics of Business Vocabulary and Rules (SBVR) [71]. Since SBVR is business agnostic semantics and ontologies are needed to give meaning to the policies and rules. The appropriate semantics and ontologies are not available in today’s enterprises [43].

Combinations of interactions between more than one real world actor exists in the literature [23, 44, 52], but it is not evident that the combinations presented in the literature can be used in other domains due to the tight coupling between the realization and the information in each solution. Interactions are not only
Chapter 1. Overview

taking place between individual actors. The interactions between groups are an integral part of the SECI-model which is used for knowledge creation [68]. The interactions between groups are supported in Pask’s conversation theory [78].

When constructing a compositional system, it is necessary to understand the information in a system’s context frames as well as having temporal separation [30], and for systems that are part of a compositional system to coordinate and cooperate in uncertain environments [25]. In a compositional system, multi-context capabilities are needed [77] since an actor’s context frame might have to act in several roles.

Intent-driven systems have a need for governance since new or modified business intents introduce changes in the correlations between business intents. To govern all the business intents requires each involved party to share a common understanding of the used governance model. Since several actors have to collaborate to fulfill a business intent it is not realistic to assume the use of one homogeneous governance model. To achieve a common understanding of heterogeneous governance models, collaboration between different governance models could be used. This collaboration requires appropriate semantics and ontologies which are not available today [43]. Governance models are discussed in, e.g. Wiesner et al. [110], Beigi et al. [7], and Lewis et al. [55].

Creating software components that can be orchestrated and bringing value to the relevant stakeholders in business ecosystems, and timely respond to frequent changes remains the main challenge. This is partly addressed by Software product lines [12] and industrial Product-Service systems [62] which focused on changeability [88], as ways to create flexible, adaptable and efficient component-based software architectures. To obtain the possibility of a high level of automation of decision and action selection [76] the information acquisition has to be effective and efficient. This requires the possibility to automatically apply rules and constraints during the information acquisition to improve the sender’s and the receiver’s understanding of the information.

To fully support business flexibility, we need to better understand and define the business context. Modeling context is also critical for developing context-aware software systems [15]. Baldauf et al. [5] summarized context-aware systems including methods to achieve context-awareness. Despite several similarities, context-aware software systems focus on dynamically discoverable services rather than dynamically changing business opportunities.

Supporting business flexibility requires support for agile business policies and agile business rules [16] which are used to govern how an enterprise does its business [69]. It is desired to have a common governance structure and a standardized way
Chapter 1. Overview

of handling the business rules. Rosca et al. have contributed valuable knowledge in the area of common governance of business rules [90, 91].

1.1.3 Research questions

According to Runeson et al. [94] research questions are statements about the knowledge that is being sought, or is expected to be discovered, during the study. The discovery or attainment of this knowledge demonstrates that the study has achieved its intended objectives.

In design science a research question is a knowledge-question supporting a design problem. The knowledge-question-answering activity returns knowledge to the design-problem-solving activity. Since design problems can create new problems, this generates an iteration over design problems and knowledge questions in design science [109].

In order to understand how we can support the design of a business studio we formulate a research question (mRQ: What design artifacts are needed in order to construct a business studio which is based on intent-driven systems?) which will serve as the main knowledge question for this design science study.

We formulate two knowledge questions in order to understand the state-of-the-art with respect to intent-driven systems. These knowledge questions are stated as research questions and are investigated in Chapter 2. The first research question (RQ1: What methods/techniques supporting intent-driven systems have been presented in literature?) is used to find evidence in the literature, of methods/techniques supporting intent-driven systems, i.e. supporting the construction, expressing and validation of business intents. RQ1 is divided into six sub-questions which are mapped to intent-driven systems with the help of Figure 1.4. Each pair of research questions in Figure 1.4 are covering how a business intent can be constructed, expressed, and validated. RQ1.1, RQ1.3, and RQ1.5 is used to investigate how a business intent can be constructed and expressed. RQ1.2, RQ1.4, and RQ1.6 is used to investigate how a business intent is validated.

The second research question (RQ2: What evidences for enabling flexible realizations of intent-driven systems have been presented in literature?) is used to find evidence in the literature of how the different methods/techniques could be used together to enable flexible realizations of intent-driven systems. We focus on four aspects; semantics, interchange of semantics, governance, and actor interaction.

The synthesis of RQ1 and RQ2 do not provide all the needed answers to construct a business studio. The aspects of semantics, interchange of semantics, governance,
and actor interaction are not provided to the extent that a business studio can be constructed. Instead we formulate a new knowledge question which is stated as a research question (RQ3: How can support for continuous changes to business intents be realized in business support systems?) and investigated in Chapter 3. The focus of the knowledge question is on the aspects of semantics, interchange of semantics, governance, and actor interaction.

The research questions RQ1-RQ3 support Ericsson in their early pre-study phase of a business studio. The results from Chapter 2, 3, and 4 generate design artifacts and new design problems which make it possible for Ericsson to produce a first proof-of-concept of a business studio. This will make it possible for us to fully answer mRQ in the future.

![Diagram of RQ1's sub-questions to the intent-driven systems model](image)

Figure 1.4: Mapping of RQ1’s sub-questions to the intent-driven systems model.

1.2 Methodology

In this section we discuss the research methodology used in this thesis. In Section 1.2.1 we discuss the research settings and data Collection, and Section 1.2.2 describes the limitations of this thesis.

1.2.1 Research settings and data collection

Each chapter details research settings and methods for data collection. However, we would like to highlight the main points for each chapter.

The systematic literature review in Chapter 2 use a methodology based on Kitchenham and Charters’ guidelines [53]. We start to investigate the need for a systematic literature review and conduct a pilot study. During the pilot study we refine the search strings, specify a study selection approach, develop a search strategy, define study quality assessment, define a data extraction process, and create the review protocol.
Chapter 1. Overview

Since the ideas span a wider area than computer science we select SCOPUS as
the primary source, with subsequent additional searches in ISI/Web of science,
ACM and IEEE Xplorer. In all, we investigate eleven different subject areas
(Multidisciplinary, Economics, Environmental Science, Social Sciences, Arts and
Humanities, Psychology, Engineering, Mathematics, Business, Decision Sciences,
or Computer Science.).

A reference management system\(^1\) is used to import the papers and the first full
text reading selection was done in the tool. The selected papers are then imported
to a tool named Atlas.ti\(^2\) supporting the grounded theory approach \([20]\). With
the help of Atlas.ti, the coding is done as an iterative process. We apply open
coding to each of the first fifteen papers. Selective coding is done on all the read
papers. With the help of memos we create the first iteration of the extraction
table. This process continues until all papers are read. The papers from the
former sets are included in new iterations. A coding and recoding is applied on
papers when the understanding of the content is improved. The theoretical codes
are supported by using network views (a feature in Atlas.ti). The network views
based on codes are used to find the relation between the papers. The extracted
data are analyzed and synthesized based on the memo concept in grounded theory
\([20]\).

In Chapter 3 we use a case study methodology. The design of this case study
is based on Runeson et al. \([94]\) using a focus group approach \([114, 89]\) in order
to mimic the workflow used in the studied company. Information is captured
in documents, drawings, and photos. We apply open coding technique \([20]\) to
analyze the collected qualitative data.

The data are collected during focus group \([114]\) interviews with the appointed
persons. During the interviews the information is captured on whiteboards, in
PowerPoint documents, and directly in a document. The interviews are approx-
imately two hours in length and are based on semi-structured focus group inter-
views \([89]\). Data triangulation is achieved by using interviews, informal meetings,
continuous member checking \([89]\).

We apply open coding technique \([20]\) to analyze the collected qualitative data.
Open coding help us to find common terms and concepts as well as to find
synonyms and hyponyms which are used by the study subjects. The first and
second authors perform open coding and iteratively discuss its results. The aim
with this coding approach is to agree upon and to find new terms or concepts

---

\(^1\)Mendeley \(https://www.mendeley.com\)

\(^2\)Atlas.ti is a system for conducting grounded theory analysis of texts and voice recordings.
In this tool you can mark text and apply codes, and you can generate different views (e.g.
a network view, where different relationships between papers can be investigated.) of your
analysis. \(http://atlasti.com\)
Chapter 1. Overview

during each meeting. These terms or concepts are linked to existing ones and their meaning are revised, if needed.

For the quasi-experiment in Chapter 3 and the action research in Chapter 4, we are provided with data from the practitioners. In both cases we use machine learning pipe-lines as the test and validation instruments.

1.2.2 Limitations

For each chapter we discuss validity threats according to Robson [89] and perform the countermeasures we find appropriate. However, we would like to highlight the main validity threats for each chapter.

The semantics is a problem when interpreting data. In Chapter 2 we investigate eleven different subject areas. Depending on the subject area and the problem context the semantics sometimes is different. This might lead to an ineffective search string but we believe that the pilot study and the refinement of the search string are mitigating this problem. In Chapter 3 we have the possibility to talk to the participants and correct misinterpretations. However, there is always a risk that an agreement is done on a too high abstraction level, leaving the interpretation of the details as an assumed consensus.

The data we use for the quasi-experiment in Chapter 3 and the action research in Chapter 4 are test data, which the practitioners provide to us. The lack of real customer data might impose a limitation on the results. The implementations are not part of the practitioners current solution.

1.3 Research Framework and Research Execution

In this Section 1.3.1 we give an overview of the framework used for our research and in Section 1.3.2 the execution of the research is described.

1.3.1 Research framework

The research framework we have use in this thesis is the design science methodology [42, 109, 61]. Design science is the design and investigation of artifacts in context. The artifacts are designed to interact with a problem context in order to improve something in that context. The problem context can be extended with
social context and knowledge context. The social context consists of stakeholders who may affect or may be affected by the project. The knowledge context consists of knowledge from natural science, design science, design specifications, useful facts, practical knowledge, and common sense. Figure 1.5 shows a framework for design science.

In Figure 1.5 one can see the two different parts of design science, design and investigation. These two parts correspond to two kinds of research problems, design problems and knowledge questions. Knowledge questions are used to understand the existing knowledge which is useful for the artifact in context. We expect to find one answer to a knowledge question. The answer should be evaluated by truth, which is not dependent on the stakeholder goals. However, it is possible that the answer is wrong or incomplete, depending on our understanding of the problem context or the knowledge context. Design problems call for a change in the real world and require an analysis of actual or hypothetical stakeholder goals. A solution is a design, and there are usually many different solutions. The solutions are evaluated by their utility with respect to the stakeholder goals, and there is not one single best solution.

How the different chapters relate to Design and Investigation is indicated in Figure 1.5.

![Figure 1.5: Design Science framework (adopted from Wieringa [109]).](image-url)
1.3.2 Research execution

The problem context in this thesis is how Ericsson’s business studio can support continuous business-requirement engineering with the help of intent-driven systems. In order to design relevant artifacts, get answers to knowledge questions, and provide new answers to the knowledge context we perform three studies. These studies are presented in Chapter 2, Chapter 3, and Chapter 4. A brief overview of how we use the design science framework for the studies are given below.

In Chapter 2 we adopt the constructive worldview in order to generate a theory which can be used to create understanding about a problem context. Together with experts from Ericsson the ideas of what is needed to create an intent-driven system are assembled. These ideas form an understanding of what is needed and useful in order to enable flexible realizations of intent-driven systems. The core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents. Based on this we formulate two knowledge questions (RQ1 and RQ2).

In order to answer RQ1 and RQ2 we perform a systematic literature review. We use Kitchenham and Charters guidelines [53] for systematic literature reviews and our purpose with the systematic literature review is in-line with what Robson states [89]. The systematic literature review help us to understand the state-of-the-art with respect to intent-driven systems. During this systematic literature review, we study available methods and techniques that may be useful for supporting intent-driven systems, as well as the existence of aspects needed to enabling flexible realizations of intent-driven systems. The data extraction process is based on the data analysis used in grounded theory.

Answers to the RQ1 and RQ2 exist. The existence of methods/techniques which can be used as building blocks to construct intent-driven systems exist in the literature. How these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems is not evident in the existing literature. The synthesis shows a need for further research regarding the aspects; semantics, interchange of information, actor interaction in intent-driven systems, and governance of intent-driven systems. The existing answers to the knowledge questions (RQ1 and RQ2) give valuable knowledge to the problem context. However, in order to design artifacts we need more knowledge about the how to fill the gaps identified in Chapter 2.

In Chapter 3 we investigate the gaps that are identified in Chapter 2. In Chapter 2
we identify a gap in the literature regarding multi-actor interactions in intent-driven systems. This indicates the need for further research since business support systems need to support the correlation of activities between actors as well as influence the activities based on knowledge about the value networks in which the enterprise acts. These activities are used to fulfill or create business intents, which often are expresses as business rules. The life cycle of business intents needs to be governed and the a semantic interchange between different parts of the ecosystem must be supported. The aim of the study is to identify how a business support system can support continuous changes to business intents. The first step is to find a theoretical model which serves as a foundation for intent-driven systems. Based on this we formulate a new knowledge question (RQ3).

Once again we adopt the constructive worldview in order to generate a theory which can be used to create an artifact. The research methods we choose from are a case study or surveys and questionnaires. The problem with a semantic interchange, which we find in the systematic literature review, indicate a need for the possibility to clarify terms with the involved participants. We therefore conduct a study in which we can get knowledge from practitioners and contribute new answers to the knowledge context.

In order to answer the knowledge question we conduct a case study using a focus group approach with employees from Ericsson. We use the methodology defined by Runeson et al. [94]. The case study is influenced by the spiral case study process [94] with an iterative character in order to stepwise adjust the goal and scope to the iterative findings. In order to analyze the data we use the open-coding technique in grounded theory. Since we decide to validate the theoretical models, after the case study is performed, we use a type of mixed methods design called sequential exploratory design. The validation of the theoretical model was performed as a proof-of-concept, studied in a quasi-experiment.

The case study result in a model supporting continuous definition and execution of an enterprise. The model is divided into three layers; Define, Execute, and a common governance view layer. This makes it possible to support continuous definition and execution of business intents and to identify the actors needed to support the business intents’ life cycles. This model is supported by a meta-model for capturing information into viewpoints. We agree with Robson about the outcome [89], but instead of performing another case study in order to get deeper understanding of the models, we decide to validate the models with the help of a proof-of-concept.

In the proof-of-concept, we borrow ideas from the gaming industry where a specific context gives the character the possibilities to, for example find specific treasures and stipulates how these treasures can be handled. The proof-of-concept
shows that; the Design layer is supported by the possibility to, visually and logically, validate the correctness of the business rules before they are put in execution. The Execute layer is supported by the possibility to logically validate the correctness of the business rules before they are put in operation, and to deploy and operate the executable business rules as a context frame meta-model. The Governance views are supported by the fact that the executable business rules can be handled as immutable artifacts. During the proof-of-concept the need for visual and logical validation became evident. The automation of rules creation and distributed execution of the rules became vital aspects during the analyze of the proof-of-concept. The proof-of-concept was implemented with the help of a machine learning pipe-line.

In Chapter 4 we validate the theoretical models found in Chapter 3 which are supporting continuous changes to business intents. Supporting continuous changes to business intents can be seen as providing business flexibility. Business flexibility is the possibility to create new business models or change existing ones, in an effective and efficient way. This includes changes to business rules which supports tactics or strategies. We take an advocacy/participatory worldview and use action research as the research methodology.

We decided to stay technology-independent, in order to achieve a conceptual business model background. One way to achieve this is to investigate legal contracts. A business model is supported by a set of legal contracts, we start to derive the business rules from these type of contracts. Some of the information in a legal contract is not meaningful to translate into a business rule which should be executed in software, e.g. which laws should be used to solve a dispute. Many times the nature of the language used in legal contracts requires human interpretation. However, the majority of the terms and conditions in a legal contract can be translated into meaningful business rules which could be implemented in software. The legal contracts we have chosen to study are targeting support for value propositions, based on different business models. The business rules are based on following five parts of the Osterwalder canvas [74]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions.

In Chapter 4 we extend the design artifact in Chapter 3 in order to support legal contracts. There are small changes to the visual and logical verification of the business rules. We use another algorithm than used in the proof-of-concept (Chapter 5) in Chapter 3, since we would like to enforce legal rules instead of performing recommendations of product offerings. The analogy with the gaming industry is even more evident in this proof-of-concept since we change the (business) state in the enterprise when handling legal contracts.
Chapter 1. Overview

Since we are doing the implementation and it is not part of the product, we call it a proof-of-concept instead of action research. This proof-of-concept provides initial results regarding software architectural mechanisms which can support context descriptions and the context description’s support for business-driven software architecture, and the required level of speed and business flexibility demanded by the business ecosystem.

The knowledge-question-answering activity returns knowledge to the design-problem-solving activity. This can create new problems which generates an iteration over design problems and knowledge questions in design science. We start to answer knowledge questions which lead to a new design problem. We conduct a proof-of-concept which gives more insight to the problem at hand. The artifact which results from our design activities is returned to the question-answering activity which is used to create new design problems. The new design problems are described in the Section 1.4.3.

1.4 Conclusion and Further Research

In Section 1.4.1 we summarize the answers to the thesis research questions based on the conclusions from Chapter 2 - Chapter 4, we describes the produced design artifacts in Section 1.4.2, and in Section 1.4.3 we give suggestions on new design problems which can be included in our further research.

Before we continue we would like to remind the reader of the three phrases which were carved into the major temple in Delphi: “nothing in excess”, “know thyself”, and “make a pledge and mischief is nigh”. Currently we have studied the available state-of-the-art and there is indeed very little in excess available. But, our research is about getting new knowledge and to be able to share this knowledge in order to improve problem contexts. By knowing ourselves, and armed with the knowledge we have gained during our research, we are quite confident of being able to improving the problem context at hand. By only presenting a part of all possible design problems as our further research, we hope to have avoided the third phrase.

1.4.1 Conclusion

RQ1 is aimed at finding evidence in the literature for the support of intent-driven systems. RQ1 is divided into sub-questions, RQ1.1-RQ1.6, each covering one vital aspect of intent-driven systems. The results indicate existence of methods/techniques which can be used as building blocks to construct intent-driven...
systems. However, the majority of the papers found during the study use an instance centric view of the problem they solve, which introduce a tight coupling between the realization and the information in each method/technique. In the proof-of-concepts we took a small step in the direction of separating the realization from the information, by using generic algorithms to produce executable business rules.

The aim of RQ2 is to find evidence in the literature of how the result of RQ1’s sub-questions could be used to enable flexible realization of intention driven systems. The synthesis show no evidence of how these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems. For example, the governance of intent-driven systems as well as possibilities for semantic interchange of information is not evidently available in the literature. In the proof-of-concepts we took smalls step in the direction of improving the semantic interchange aspect and the governance aspect. The implementation makes it possible to, visually and logically, validate the correctness of the business rules before they are put in production. The possibility to generate executable code representing the model of the business rules, makes it possible to execute the same model in different components without the need of re-implementation.

A continuous definition and execution of a business intent’s life cycle in an enterprise and its value networks, could not be found in the existing literature. Nor did we find a meta-model supporting a context frame aware realization of a business intent’s life cycle in a compositional way. RQ3 is addressed by suggesting a solution supporting continuous re-definition and execution of an enterprise as a model of value architecture layers and business functions, divided into the Defined and the Execute layers. This model mimics the reality of an enterprise. To support the model we build upon Pask’s conversation theory [78] by introducing the context frame. This makes it possible to support continuous definition and execution of business intents supporting the enterprise and its value networks. To the best of our knowledge this ability is not presented in the literature. With the help of the proof-of-concepts we have realized the models in a limited way. The main contributions of the models is the support the models give for the understanding of how to separate business models from the implementation of a business support system.

1.4.2 Produced design artifacts

We have produced two design artifacts as a start to be able to answer mRQ. The proof-of-concepts are seen as new problem solving knowledge by the stakeholder.
The models in Chapter 3 have proved to be a valuable artifact for the stakeholder. The models in Chapter 3 are suggesting a solution supporting continuous re-definition and execution of an enterprise as a model of value architecture layers and business functions, describes the initial steps of intent-driven systems, and encapsulates the context frame meta-model.

In the next section we are suggesting new design problems to investigate, in order to get closer to the possibility of answering mRQ.

1.4.3 Suggested new design problems

The possibility of transforming business intents into executable business rules is vital in order to achieve an effective and efficient business. There is a need for a highly automative transformation process which will require effective and efficient algorithms for both for the translation of the legal contracts into business rules and the transformation of the business rules into executable business rules. To be able to find highly automative transformation processes we will continue to investigate how the business intents in legal contracts can be transformed into executable business rules.

Business rules can be evaluated and enforced as part of business processes. Many of the business processes in an enterprise can have a high level of automation. By encapsulate business processes in executable containers it can be possible to obtain an effective and efficient business by supporting dynamic, thin, and throwaway processes. In order to support encapsulation of business processes in executable containers, we will continue to investigate how context frames can be used as executable containers.

The transformation of business intents into executable business rules requires the possibilities to find needed capabilities, and to compose executable containers from these capabilities. Since there can exist a multitude of different stakeholder and their intents can be expressed in a none technical language, a fuzzy search mechanism might be needed. The methodology proposed in Chapter 6, and its suggested improvements, are small steps in the direction of an interactive fuzzy search mechanism. In order to be able to create compositions of capabilities, matching functionalities between the capabilities are needed. We will investigate how technical proposals to business intents can be exposed as compositions of context frames.

All the design problems described in this section share the need for governance. As mentioned before, the core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires
Chapter 1. Overview

a separation of concerns between governance of the parts of the system used to execute the stakeholder business, and governance of the parts which are used to design the business based on stakeholder intents. Since the execute and the design parts are related there are interactions between the different governance structures. From the described design problems we can conclude, business intent, business rules, container capabilities, and execution of containers need their own governance structures which are interacting with each other. The described structure forms a compositional system of governance structures. We need to extend our research to cover the governance of business rules for both the design and the execute parts, as well as their interactions. We will use this research as a leap-stone for further studies.
Chapter 2

A Systematic Literature Review on Intent-Driven Systems
Abstract

Context: The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of the intents. Only then are different computer-based agents able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents.

Objective: The aim is to find out which methods/techniques as well as enabling aspects, useful for an intent-driven system, that are covered by research literature.

Method: As a part of a design science study, a Systematic Literature Review is conducted.

Results: The existence of methods/techniques which can be used as building blocks to construct intent-driven systems exist in the literature. How these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems is not evident in the existing literature.

Conclusion: The synthesis shows a need for further research regarding semantic interchange of information, actor interaction in intent-driven systems, and the governance of intent-driven systems.

2.1 Introduction

The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of them. Only then are different computer-based agents able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement.

We define intent as a subject or type of possible behavior, i.e. something that can be interpreted to have significance. Any actor can have intents. When an actor publicly declare an intent in a certain context, it becomes a stated intent. A stated intent \(^1\) could be a declaration of capabilities an actor promise to provide, or a requirement an actor try to impose on another actor. A true cooperation between actors is based on the capabilities promised by the interacting actors.

\(^1\)In the remaining part of the paper we write intent instead of stated intent, if there is no ambiguity of the meaning.
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

A decision about whether an actor has kept its promise or not, can be done by the promising actor as well as any actor which is observing the behavior of the promising actor.

Today’s enterprises are part of value networks. A value network refers to “A set of connections between organizations and/or individuals interacting with each other to benefit the entire group [45]”. Enterprises in a value network can be seen as parts in a compositional system and are by themselves compositional systems, also known as a system of systems. Compositionality refers to “The evident ability of humans to represent entities as hierarchies of parts, with these parts themselves being meaningful entities, and being reusable in a near infinite assortment of meaningful combinations [39]”. The construction of a compositional system requires methods to achieve a holistic collective benefit through the individual systems’ participation and cooperation when each system adopts a solution that maximizes its own self-interest [29].

The actors in a compositional system may be humans or machines. By using software agents as machine actors enterprises can bring customers closer to suppliers of products and services, support their continued demand for change, inject further intelligence into enterprises and simplify the environment for both customers and employees [3]. Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the compositional systems the enterprise acts in.

We define an intent-driven system as a compositional system of computer-based agents where the agents declare, negotiate, and assess intents made by agents. The intents can be declared, negotiated, or assessed, on-behalf of an agent’s stakeholder or an agent’s self-interest.

Intent-driven systems capture stakeholders’ intents in the form of business requirements or capabilities, and transform these intents into a form that enables computer processing of them. Only then are different computer-based agents able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement. We model this according to Pask’s Conversation Theory [78], as illustrated in Figure 2.1. Two key components in Pask’s conversation theory are language and domain. A language L is defined as: A language L is defined as:

“L may be a natural, written or spoken, symbolic language, but it need not be. It may be a system of symbolic behaviors such as dance or actions such as key pressing. It may be formalized, as in, mathematics and higher level programing languages, but it need not be. It must however have many of the qualities of a natural language, with
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

possibilities to express and interpret commands, questions, answers, obediences, explanations, or descriptions.” [56].

A domain could be described as:

“A domain is a collection of topic’s, and a topic is essentially a relation. This may be a very concrete relation (a relation between alphabetic characters and the keyboard positions in type writing) or it may be an abstract relation (a relation between smugglers and the countries they operate in): To learn or solve a problem is to ‘bring about’ such a topic relation.” [56].

An example of the initial steps of intent-driven systems, supporting the business are shown in Figure 2.1. In Figure 2.1 we see how two stakeholders, A and B, representing a business side and a operational side, respectively, communicate with each other both directly and through computer agents, A’ and B’. Stakeholders A and B negotiate via a language L, regarding the domain D. Stakeholder A expresses its intents as business requirements, that are communicated to Component A’ via a language \( L_{hm,A} \). Component A’ interprets the intents as business processes and business rules that govern these processes. The results derived by Component A’ are presented to Stakeholder A in language \( L_{mh,A} \). This initiates a cycle where Stakeholder A interacts with Component A’ until a common understanding of the business intents are accurately described in \( L_{mh,A} \). Correspondingly, Stakeholder B expresses its intents in the form of the capabilities it aim to offer, and the job of Component B’ is to map these capabilities to machine and human actors that can together supply the offered capabilities. When a common understanding between the stakeholders and their related components are achieved the components are able to start negotiating with each other over language L’, regarding the domain D’. The results are presented back to their respective stakeholder in the form of effectiveness and efficiency measures. The stakeholders may then continue the negotiation via language L in an iterative cycle until an agreement can be made.

In order to obtain the effectiveness and efficiency measures, the components need to interact with their correlated components in the compositional system. These interactions may introduce changes to the system or involve new actors and components to be part of the decision process. The possibility to introduce changes to the system requires flexible realizations of intent-driven systems. The decision process’ level of automation, of decision and action selection, depends on the involved components capabilities and rights of taking decisions and performing actions. In order to control and manage these capabilities and rights, governance views are needed. Parasuraman et al. [76] presents a scale of the “levels of automation of decision and action selection” ranging from 1 (The computer offers...
Figure 2.1: An example of the initial steps of intent-driven systems, supporting the business.

no assistance; human must take all decisions and actions.) to 10 (The computer decides everything, acts autonomously, ignoring the human.).

Pask [78] stresses the fact that the different actors have obtained their specific domain information through several different interactions which makes the model recursive. This means that each actor obtains its specific domain information in different contexts. A context has boundaries defined by who, what, when, where and how [5]. We define a context frame as the total domain information for a specific domain an actor has obtained.

2.1.1 Background

The research project is done in collaboration with Ericsson and is based on the design science methodology [42].

At the time of the study, Ericsson was in an early pre-study phase of a business studio. The business studio is part of a business support system and will act in the area of planning and monitoring business intents. The idea with the business studio is to deliver support for a 360-degree view of an enterprise’s business. The view includes both the actual execution of an enterprise’s business and the intended changes to this execution. The business studio will support and govern the decisions and actions needed to maintain or change the way an enterprise does its business. Ericsson’s customers will be able to buy the business studio as a product or a service. With the help of the business studio, Ericsson’s customers will get support and knowledge about how to configure, monitor, and redesign their products and business. The software could be used in different business
areas, for example charging, billing, customer relationship management, partner relationship management, product management, order management, etc. The business studio supports the idea of continuous business-requirement engineering. Intent-driven systems are one of the cornerstones for the business studio product.

During the first phase of this study, the ideas of what is needed to create an intent-driven system were assembled. These ideas form an understanding of what is needed and useful in order to enable flexible realizations of intent-driven systems. The core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents. In order to understand the state-of-the-art with respect to intent-driven systems, it was decided to conduct a systematic literature review. During this systematic literature review, we study available methods and techniques that may be useful for supporting intent-driven systems, as well as the existence of aspects needed to enabling flexible realizations of intent-driven systems.

2.1.2 Research questions

The goal with the research questions is to find out which methods and techniques useful for supporting intent-driven systems are covered by literature, as well as the interactions between the methods and techniques. The first research question is to find evidence in the literature, of methods/techniques supporting intent-driven systems, i.e. supporting the construction, expressing and validation of business intents. The second research question is to find evidence in the literature of how the different methods/techniques could be used together to enable flexible realizations of intent-driven systems.

RQ1: Supporting Techniques and Methods for intent-driven systems

RQ1: What methods/techniques supporting intent-driven systems have been presented in literature?

According to Benaroch [8] the following three parts are needed when representing knowledge:

- Construct – how to construct the business intents,
- Express – how to express business intents in a match-able way, and
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

- Validate – how to ensure the business intents are actually met by a particular configuration.

We used Benaroch’s findings together with Figure 2.1 to divide RQ1 into sub-questions. The mapping of the sub-questions to the intent-driven systems model are shown in Figure 2.2.

Figure 2.2: Mapping of RQs to the intent-driven systems model.

To the left in Figure 2.2 the following sub-questions are found:

RQ1.1: What methods/techniques have been used to express and construct human actors’ intents?
RQ1.2: What methods/techniques have been used to validating the human actors’ intents?

To the right in Figure 2.2 the following sub-questions are found:

RQ1.3: What methods/techniques have been used to express and construct machine actors’ intents?
RQ1.4: What methods/techniques have been used to validating the machine actors’ intents?

In the middle of Figure 2.2 the following sub-questions are found:

RQ1.5: What methods/techniques have been used for matching intents expressed by human actors and intents expressed by machine actors?
RQ1.6: What methods/techniques have been used for validating the matching of intents expressed by human actors and intents expressed by machine actors?

RQ2: Realization of intent-driven systems

To find evidence in the literature for enabling flexible realizations of intent-driven systems we formulated the following research question:

RQ2: What evidences for enabling flexible realizations of intent-driven systems have been presented in literature?
Together with domain experts from Ericsson we used Figure 2.1 to find aspects needed to enabling flexible realizations of intent-driven systems. These aspects were used to evaluate if the methods/techniques found in RQ1.1-RQ1.6 can be used together, as described in the literature, for enabling flexible realizations of intent-driven systems.

*Semantics* is the aspect making it possible to obtain the Languages in Figure 2.1.

*Interchange of semantics* is the aspect making it possible for actors to collaborate on an intent with the help of their individual context frames.

*A governance aspect* is needed in order to provide knowledge about the correlation factors, support intents’ life cycles and the correlation between intents. Governance refers to “The discipline of monitoring, managing, and steering a business (or IS/IT landscape) to deliver the business outcome required [73]”. The governance supports the configuration, simulation and feedback of changes to existing business intents as well as for the introduction of new business intents.

*The aspects of actor interaction* have to be considered in order to enable flexible realizations of intent-driven systems. Pask’s conversation theory supports the interaction between an arbitrary number of actors. The actors could be human actors as well as machine actors. These aspects of actor interaction have to be considered.

### 2.1.3 Summary

The contribution of the paper is a review of existing methods/techniques which could be used as building blocks to construct intent-driven systems. The paper provides insight of what is needed to enable intent-driven systems with the help of these methods/techniques. The results may be used by practitioners who want to understand what exists and what is missing when constructing intent-driven systems. Researchers may find areas where gaps for realizing intent-driven systems have been identified.

The remainder of this article is organized as follows. The review methodology is presented in Section 2.2 and the execution results and an analysis thereof are presented in Section 2.3. The results are further discussed in Section 2.4 and conclusions and future work are presented in Section 2.5.
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

2.2 Review Methodology

In order to answer the research questions, we conducted a Systematic Literature Review. The used methodology is based on Kitchenham and Charters’ guidelines [53].

Since the ideas span a wider area than computer science we selected SCOPUS as the primary source, with subsequent additional searches in ISI/Web of science, ACM and IEEE Xplorer. The study selection criteria and data extraction properties were created based on our understanding of what is needed in order to enable flexible realizations of intent-driven systems, strengthened by discussions with domain experts at Ericsson. To further mitigate the risk of author bias, a researcher external to the project was invited to check the results. Before the study was started we performed a search for literature reviews. The result showed that no literature reviews, relevant for intent-driven systems, have been performed. Details on this pre-pilot can be found in A.1.

Below, we briefly present the specifics for this study with respect to the pilot study, search strings, the approach to study selection, search strategy, study quality assessment, data extraction process and the review protocol.

2.2.1 Pilot study

We piloted the systematic review procedure in order to establish a common view of the selection criteria (A.2). The selection criteria were applied on a trial search against SCOPUS, where inclusion criteria IC1-IC4 (Table 2.2) were applied. Forty papers were randomly selected. The main author and a researcher external to the project individually assessed titles and abstracts of the selected papers. The assessment showed a high consensus between the main author and the researcher (Cohen’s Kappa coefficient equal to 0.754). Discussions were performed on papers for which no consensus were found during the assessment. The discussions led to a higher consensus between the main author and the researcher.

The result of the pilot indicate that the search terms produce relevant hits, and that the initial choice of search database was reasonable. Thus, we decided to continue with the main search in additional sources and with amended search terms as described below.
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

2.2.2 Refining research question and/or search strings

During the pilot study, the papers selected as primary studies for full-text reading were used to find new keywords for the search string. The Author keywords were extracted from the selected papers. We selected the keywords to be part of the search string based on our understanding of intent-driven systems.

2.2.3 Search string

The aim with the search string is to find papers covering aspects needed to enable flexible realizations of intent-driven systems. The research questions are searching for methods/techniques supporting intent-driven systems. The search string is composed of terms grouped together as K, L, M, N and O and executed as K AND L AND M AND N AND O. The groups and their terms are found in Table 2.1. The new keywords found during the pilot study were added to group O. The other keywords were obtained from the research questions, and through brainstorming with domain experts from academia and the industry, as described in A.1 and A.2.

<table>
<thead>
<tr>
<th>Group</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>(&quot;intent&quot; OR &quot;intention&quot; OR &quot;vision&quot; OR &quot;strategy&quot; OR &quot;strategies&quot; OR &quot;goal&quot; OR &quot;tactic&quot; OR &quot;objective&quot;)</td>
</tr>
<tr>
<td>L</td>
<td>(&quot;driven&quot; OR &quot;motivated&quot; OR &quot;focused&quot;)</td>
</tr>
<tr>
<td>M</td>
<td>(&quot;system&quot; OR &quot;solution&quot; OR &quot;realization&quot; OR &quot;realisation&quot;)</td>
</tr>
<tr>
<td>N</td>
<td>(&quot;govern&quot; OR &quot;guide&quot; OR &quot;control&quot;)</td>
</tr>
<tr>
<td>O</td>
<td>(&quot;decision support&quot; AND &quot;DSS&quot;) OR &quot;behavior driven&quot; OR &quot;negotiation driven&quot; OR &quot;collaboration driven&quot; OR &quot;feedback observation&quot; OR &quot;adaptive policy&quot; OR &quot;adaptive policies&quot; OR &quot;adaptive rule&quot; OR &quot;adaptive decision&quot; OR &quot;adaptive visualization&quot; OR &quot;adaptive visualisation&quot; OR &quot;composition optimization&quot; OR &quot;composition optimisation&quot;)</td>
</tr>
</tbody>
</table>

Table 2.1: Search Strings for the main study.

2.2.4 Inclusion/exclusion criteria

The main criterion for inclusion as a primary study is the existence of explicitly described methods/techniques, which answer RQ1. During the primary study selection the main criterion should be observable in the title or abstract. During the full-text reading of a selected paper the main criterion should be described/explained in the text. The inclusion and exclusion criteria are described in Table 2.2. Additional information can be found in A.2.
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

<table>
<thead>
<tr>
<th>ID</th>
<th>Inclusion/exclusion parameter</th>
<th>Inclusion/exclusion criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC1</td>
<td>Year</td>
<td>2000-2014 (inclusive)</td>
</tr>
<tr>
<td>IC2</td>
<td>Language</td>
<td>English</td>
</tr>
<tr>
<td>IC3</td>
<td>Document type</td>
<td>Conference paper, Article, Review</td>
</tr>
<tr>
<td>IC4</td>
<td>Subject area</td>
<td>Any of the following subject areas: Multidisciplinary, Economics, Environmental Science, Social Sciences, Arts and Humanities, Psychology, Engineering, Mathematics, Business, Decision Sciences, or Computer Science.</td>
</tr>
<tr>
<td>IC5</td>
<td>Information structure</td>
<td>Full text available</td>
</tr>
<tr>
<td>IC6</td>
<td>Relevance of the contribution to intent-driven systems</td>
<td>Methods/techniques are described in theory and the theory is found useful for intent-driven systems.</td>
</tr>
<tr>
<td>EC1</td>
<td>Redundant information</td>
<td>A paper with the same information from different sources or an updated version(s) of the paper exists. The latest copy is kept.</td>
</tr>
</tbody>
</table>

Table 2.2: Inclusion/exclusion criteria

2.2.5 Search strategy

To cover a wide range of disciplines, we use Scopus, ISI/web of science, ACM digital library, and IEEE Xplorer. To get the initial set of papers we followed the following process. Each selected resource was searched using the search string with IC1-IC4 applied directly in the selected resource’s search engine. The retrieved papers were filtered from duplicates with the help of a reference management system\(^2\). Finally, the main selection criteria were applied and the remaining papers are part of the primary study for full-text reading. Reviews were analyzed based on their unique contribution, i.e. not based on their cited papers. The primary studies were obtained by applying IC5, IC6 and EC1.

2.2.6 Quality assessment criteria

Basic academic quality is addressed by the inclusion/exclusion criteria. We use the additional quality assessment to assess the level of contribution from each article towards intent-driven systems. A method/technique is regarded as useful for intent-driven systems; if the described theory could be used outside the paper’s specific solution or if the paper enables a flexible realization of an intent-driven system.

The relevance of the papers is judged according to the following criterion (where articles matching level 1 are deemed as being most useful, followed by articles matching level 2 and 3 (least useful). Articles not matching any criterion level are deemed as not useful):

\(^2\)Mendeley [https://www.mendeley.com](https://www.mendeley.com)
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

Level 1 The methods/techniques described are in the focus of the paper and articulated in the study. The described theory is found useful for intent-driven systems.

Level 2 The methods/techniques described are not in the focus of paper and not fully articulated in the study. The described theory is found useful for intent-driven systems.

Level 3 Methods/techniques are described in theory and the theory is found useful for intent-driven systems.

Since Level 3 and IC6 are identical we get a possibility to re-evaluate a paper’s relevance as a contribution to intent-driven systems.

2.2.7 Data extraction process

To find relevant papers we followed the process described in Section 2.2.5. A reference management system was used to import the papers and the first full text reading selection was done in the tool. The selected papers were then imported to a tool\(^3\) supporting the grounded theory approach [20] as described below. The papers were read iteratively resulting in a mature data extraction table (A.3). The extracted data was analyzed and synthesized based on the memo concept in grounded theory [20].

With the help of Atlas.ti, the coding was done as an iterative process. We applied open coding to each of the first fifteen papers. Selective coding was done on all the read papers. With the help of memos we created the first iteration of the extraction table (A.3). This process continued until all papers had been read. The papers from the former sets were included in new iterations. A coding and recoding was applied on papers when the understanding of the content has improved. The theoretical codes were supported by using network views (a feature in Atlas.ti). The network views based on codes where used to find the relation between the papers.

2.2.8 Review protocol

The aim of the review protocol is to reduce the potential researcher bias and to permit a replication of the review in the future. The protocol was evaluated by an

\(^3\) Atlas.ti is a system for conducting grounded theory analysis of texts and voice recordings. In his tool you can mark text and apply codes, and you can generate different views (e.g. a network view, where different relationships between papers can be investigated.) of your analysis. http://atlasti.com
Chapter 2. A Systematic Literature Review on Intent-Driven Systems

independent researcher, who is experienced in conducting systematic literature reviews. According to the researcher’s feedback and our own gathered knowledge during the process, we iteratively improved the design of the review protocol. Each step in the protocol was performed in iterations and a re-evaluation of the protocol was performed as part of the iteration. The re-evaluations were discussed and decided upon together with the researcher.

The review protocol is maintained in the reference management system. This includes the IC5, IC6 and EC1 (See Table 2.2). The IC1-IC4 were handled by the search engine and are not part of the protocol.

2.3 Results and Analysis

Since the ideas span a wider area than computer science we selected SCOPUS as the primary source, with subsequent additional searches in ISI/Web of science, ACM and IEEE Xplorer. The results of the search are shown in Table 2.3.

<table>
<thead>
<tr>
<th>Activity (Executed 20150301)</th>
<th>SCOPUS</th>
<th>ISI</th>
<th>ACM</th>
<th>IEEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keyword search</td>
<td>1314</td>
<td>14</td>
<td>130</td>
<td>3623</td>
</tr>
<tr>
<td>Excluded due to inclusion criteria 1 or 2 or 3</td>
<td>-239</td>
<td>-4</td>
<td>-21</td>
<td>-138</td>
</tr>
<tr>
<td>Retrieved papers</td>
<td>1075</td>
<td>10</td>
<td>109</td>
<td>51</td>
</tr>
<tr>
<td>Excluded due to duplicates or inclusion criterion 4</td>
<td>-260</td>
<td>-10</td>
<td>-7</td>
<td>-3</td>
</tr>
<tr>
<td>Primary Study Selection</td>
<td>815</td>
<td>0</td>
<td>102</td>
<td>48</td>
</tr>
<tr>
<td>Excluded due to inclusion criterion 6</td>
<td>-703</td>
<td>0</td>
<td>-87</td>
<td>-45</td>
</tr>
<tr>
<td>Primary study for full-text reading</td>
<td>112</td>
<td>0</td>
<td>15</td>
<td>3</td>
</tr>
<tr>
<td>Excluded due to inclusion criteria 5 or 6 or exclusion criterion 1</td>
<td>-80</td>
<td>0</td>
<td>-13</td>
<td>-2</td>
</tr>
<tr>
<td>Remaining Primary Studies</td>
<td>32</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2.3: Included and excluded studies

The individual research questions are evaluated one by one and at the end an intent-driven system perspective is discussed. Individual comments on the primary studies are available in A.4.

The information in the papers are categorized according to the quality criterion in section 2.2.6, as presented in Table 2.4. The table lists the articles fulfilling the three quality criterion levels. For papers passing criterion level 1 (i.e. deemed most useful for the understanding and construction of intent-driven systems), the theories/methods/techniques are listed. Table 2.4 is used to answer RQ1.

---

4Since IEEE Xplorer do not provide good search capabilities for large search strings a new method was developed. The search string was divided into two sets and these sets where, after duplicate removal, joined to find the result set. The resulting set contained 51 papers as indicated in the table. Before the join was performed 4485 papers were obtained. IC4 could not be obtained within IEEE Xplorer.
found articles that pass quality assessment criteria level 1, for all sub-questions of RQ1 (see Table 2.4). Together, the listed theories/methods/techniques from the articles passing level 1 of the quality criterion constitute the answer to RQ1.

<table>
<thead>
<tr>
<th>RQ Level 1 Level 2 Level 3</th>
<th>Level 1 Theories / Methods / Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 [1], [17], [19], [24], [37], [43], [57], [95]</td>
<td>Multi Criteria Decision Analysis, Multi Agent Systems, Negotiation protocols, Structured argument capturing, Bayesian theories, Decision Map and Discrete Wavelet Transformation.</td>
</tr>
<tr>
<td>[11], [23], [32], [44], [63], [75], [86], [106]</td>
<td></td>
</tr>
<tr>
<td>1.2 [19], [75]</td>
<td>Bayesian theories and Multi Criteria Decision Analysis.</td>
</tr>
<tr>
<td>1.3 [66] [7], [11], [59], [65], [75], [99], [102], [106]</td>
<td>Semantic technology based on WSDL.</td>
</tr>
<tr>
<td>1.4 [59], [75] [99]</td>
<td>Bayesian theories and Genetic algorithms.</td>
</tr>
<tr>
<td>1.5 [96] [7], [60], [80] [25], [58], [103], [110]</td>
<td>Algorithm to reason about most preferred outcome for a compositional system.</td>
</tr>
<tr>
<td>1.6 [7], [60]</td>
<td>Principal Component Analysis with or without neural networks.</td>
</tr>
</tbody>
</table>

Table 2.4: Mapping of papers to RQ1’s sub-questions.

To evaluate RQ2 the analysis is divided into a number of aspects needed to enable flexible realizations of intent-driven systems. We used all the papers in Table 2.4 to evaluate if the aspects are present in the literature. The first four aspects below are described in Section 2.1. The fifth aspect is used to understand if a theory supports more than one research question.

- Semantics
- Interchange of semantics
- Governance
- Actor interaction
- Commonalities (Theories supporting more than one research question.)

Below, we discuss each of these aspects in further detail.

**Semantics:** Business intents and their life cycles are developed and maintained through interactions. With the help of the various methods found in the papers (as listed in Table 2.4), these interactions are performed as collaborations between different actors. The outcome of an interaction between human actors results in conclusions stated in natural language. From a software engineering perspective the conclusions can be made executable in the form of policies and rules but this would require Natural Language Processing, suggested in [110], and a formal way
of expressing the policies and rules, e.g. using Semantics of Business Vocabulary and Rules [71]. Since Semantics of Business Vocabulary and Rules is business agnostic, semantics and ontologies are needed to give meaning to the policies and rules:

“We can summarize that currently there is no approach that deals with the distributed nature and the existing information asymmetry of the enterprise management domain in an appropriate way and allows for some support by means of application systems.” [43].

**Interchange of semantics:** Most papers take a common understanding between different actors for granted, and a few of them indicate the lack of semantics and ontologies but select a proprietary model to be able to do an experiment. Methods found in the papers (as listed in Table 2.5) provide possibilities for semantic interchange of information, or the abilities to match actors’ business intents. However, the papers do not consider the fact that the same data could have different meaning to different actors due to the actors’ individual context frames.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Discussed</th>
<th>Investigated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Governance framework</td>
<td>[7], [55], [110]</td>
<td></td>
</tr>
<tr>
<td>Interchange of Semantics and Ontologies</td>
<td>[4], [17], [23], [43], [52], [57], [65], [66], [80], [82], [110]</td>
<td>[11], [55]</td>
</tr>
</tbody>
</table>

Table 2.5: Mapping of papers related to governance frameworks or methods to interchange semantics and ontologies between actors.

“Interoperability of the CPS\(^5\) elements has to be guaranteed by specific requirements. Dynamically changing and emergent behavior must be included in the CPS specification. Natural language could be used as an informal requirements specification for exchange between the system user and stakeholders from various disciplines, but is often unclear and ambiguous. Furthermore [sic], it can barely be handled automatically.” [110].

**Governance:** Compositional systems have a need for governance since new or modified business intents introduce changes in the policies and rules. Methods which could be considered in the validating part of a governance framework are only discussed in some of the papers (as listed in Table 2.5). In the remaining literature no evidence of the governance aspects in the presented methods/techniques could be found.

\(^5\)Cyber Physical Systems, our note.
**Actor interaction:** In the literature, different combinations of interactions between human actors, machine actors, and the environment to fulfill a business intent could be found. The presence of the human actors, machine actors and the environment is either existing in the real world, outside a solution, or are embedded in the solution in the form of, for example survey data, specific models of humans or the simulated environment for a specific instance of a problem. In literature on criterion level 1 only one real world actor is supported to interact with a solution. Combinations with more than one real world actor exist in the literature [23, 44, 52], but it is not evident that the combinations presented in the literature can be used in other domains due to the tight coupling between the realization and the information in each solution.

**Commonalities:** Some theories found in the literature are common to more than one research question. This indicated a possibility to enable flexible realizations of intent-driven systems. Multi Agent Systems are used in papers supporting RQ1.1 and RQ1.3. Multi Criteria Decision Analysis supports this for RQ1.1 and Policy based systems for RQ1.3. In papers supporting RQ1.5 Natural Language Processing is suggested as a bridge between RQ1.1 and RQ1.3 but in practice this is hard to achieve due to different abstraction levels [110]. Principal Component Analysis is used as a common theory for RQ1.6. No common theories for RQ1.2 or RQ1.4 are found. It seems difficult to combine the different interaction patterns to enable flexible realizations of intent-driven systems due to the literature’s limited evidence of governance frameworks combined with the lack of methods to interchange semantics and ontologies between actors (Table 2.5).

“there is a huge semantic gap between the high-level specifications collected in PIT\(^6\) models and the particularities of a given transformation language collected in a PST\(^7\) model.” [11].

“to the best of our knowledge there is no agent-based approach proposed for the enterprise management domain so far. The interaction of agents is based on communication. Therefore, appropriate semantics and ontologies are required to allow for modeling collaborations between different enterprises (cf., for example, Vaishnavi and Kuechler 2005\(^8\)). However, it seems that a unified semantics and ontology is missing, but the enterprise ontology (cf. Uschold et al. 1997\(^9\)) is an appropriate starting point.” [43].

---

\(^6\)Platform Independent Transformation, our note.

\(^7\)Platform Specific Transformation, our note.


“We thus emphasise the point that as autonomic management is essentially human governance resulting in the constraint of adaptive behaviour using policies, we must address the semantics of both adaptive networks and adaptive application software in relating such policies to the expected human experience. This approach however, leaves many open questions relating to the limits of semantic based reasoning in the context of adaptive, networked systems.” [55].

We found few studies on large-scale open systems and noticed a low industry involvement in the existing literature. RQ1.1 and RQ1.6 are the only research question on criterion level 1 having industry involvement.

In summary, while there exists methods and techniques, listed in Table 2.4, addressing each of RQ1’s sub-questions, the synthesis show no evidence of how these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems. The existences of methods/techniques that can be used to express business intents and transfer or translate business intents between different actors’ domains are not evidently available in the existing literature. No evidence of the governance aspects needed to enable flexible realizations of intent-driven systems could be found in the literature. It is not evident that the actor interaction combinations presented in the literature can be used in other domains due to the tight coupling between the realization and the information in each solution.

2.4 Discussion

We expected to find evidence for the research questions, since the pilot study showed that the research questions and their subquestions were addressed by the papers selected for full text reading, and since we used of additional sources during the study. Many valuable contributions to intent-driven systems are presented but there is a lack of evidence for enabling flexible realizations of intent-driven systems in the existing literature. Only a minority of the studies are done on large-scale open systems, or with industry involvement, which might be one explanation for the lack of evidence for enabling flexible realizations of intent-driven systems. Another explanation might be the core idea of intent-driven systems; we are investigating the existence of methods/techniques and aspects which could serve as a generic foundation of systems which could change behavior of the system itself, based on stakeholder intents. The majority of the papers found during the study use an instance centric view of the problem they solve, which introduce a tight coupling between the realization and the information in
each method/technique. This was discovered during the full text reading of the papers.

The construction of a compositional system requires methods to achieve a holistic collective benefit through the individual systems’ participation and cooperation when each system adopts a solution that maximizes its own self-interest [29]. For example, when constructing a compositional system, it is necessary to understand the information in a system’s context frames as well as having temporal separation [30], and for systems that are part of a compositional system to coordinate and cooperate in uncertain environments [25]. The context frame is a key concept in achieving context-aware business architectures. It is vital to understand which context applies in a specific situation and how one should react upon this. In a compositional system multi-context capabilities are needed [77] since an actor’s context frame might have to act in several roles. Most contributions in this study are found in the area of expressing and validating human actor’s intents (RQ1.1 and RQ1.2, to some extent RQ1.5 and RQ1.6 were covered). The focus is on one context frame (with real world actors) only. This may be sufficient as a start but the transformation between different context frames in a later stage in not evidently available in the literature. Wiesner et al. [110] propose Natural Language Processing as an alternative to produce formal descriptions of the conclusions from the text-based results, but no solutions are given.

The information management for an enterprise being part of a compositional system becomes a critical aspect [10]. It is not evident that the validating methods/techniques of the transformations presented in the literature can be used in other context frames due to the tight coupling between the realization and the information in each method/technique (RQ1.2, RQ1.4 and RQ1.6). The contributions covering reusable capabilities and characteristics of software components do not take different context frames into consideration (RQ1.3 and RQ1.5). How to re-use the methods/techniques in different domains is not evidently available in the literature. For example, when changing between domains new ontologies might be needed. Frameworks supporting ontology negotiation exists, for example the FIPA Communicative Act Library Specification [34] used in JADE (JADE is proposed in [106]), but the translation capabilities between different context frames are not part of the found frameworks.

There is a need for governance since new business intents or a change of existing business intents introduces changes in the configuration of a compositional system. This problem is brought up by Wiesner et al. [110] but no solutions are given. In each part of a solution there exists problems of changing configurations in a way that are effective and efficient [33]. Before the changes are introduced the effectiveness and efficiency of the business intents should be evaluated, e.g. with capability levels [100]. Each actor’s level of automation of decision and
action selection has to be governed. The governance aspects of the presented methods/techniques are not evidently available in the literature.

2.4.1 Validity threats

In this section we discuss validity threats according to Robson [89].

Author bias

An extensive history as an industry practitioner may have influenced the aims of the study with a stronger bias towards ready solutions. A rigorous review methodology with stringent research questions help avoiding the risk that articles are dismissed without due consideration. Moreover, using a grounded theory approach [20] fosters a focus on the merit of each paper.

Finding the relevant papers for the study

In order to improve precision and recall, we obtained the terms in the search string as described below.

To avoid missing important search terms in the search string, the initial keywords were obtained from the research questions and through brainstorming with domain experts from academia and the industry. During the pilot study, the papers selected as primary studies for full-text reading were used to find new keywords for the search string. The Authors’ keywords were extracted from the selected papers. We selected the keywords to be part of the search string based on our understanding of intent-driven systems.

Multiple subject areas

Since we are searching in multiple subject areas some of the words and concepts have different meaning in the context of the different subject areas. In order to better understand the meaning we analyzed one subject area at the time, and read the relevant parts of the papers several times, in order to incrementally revise the authors’ understanding of the different subject areas.
Selection of information sources

Since the selection of information sources should cover a wide range of disciplines with its base in computer science, we used Scopus, ISI/web of science, ACM digital library, and IEEE Xplorer as information sources. To mitigate the risk of missing vital information sources we were recommended by a colleague to compare our search result with Inspec. We assumed the coverage to be fulfilled since ACM digital library and IEEE Xplorer are indexed by Inspec. To validate our assumption we performed a search in Inspec after the data extraction process. The reference management system showed that only duplicates were found.

Inclusion/exclusion criteria

The research question covers a broad topic. There is a need for inclusion and exclusions criteria that limit the amount of papers to read but catch the vital papers. During the pilot study the inclusion and exclusion criteria were evaluated with a researcher external to the project. Discussions between the authors were used to judge if a paper should pass with respect to IC6, if we were in doubt. If no agreement could be made between the authors, the researcher was used to judge if the paper should pass with respect to IC6.

Quality assessment criteria

We saw a need to use an additional quality assessment to assess the level of contribution from each article towards intent-driven systems. During the pilot study the quality assessment criteria were evaluated with a researcher external to the project. Discussions between the authors were used to judge how a paper should be ranked, if we were in doubt.

The use of Grounded Theory methodology

The use of grounded theory opens up for faulty interpretations if the process is used in a wrong way. A researcher external to the project evaluated the process used when applying grounded theory.

The rigor and relevance of the selected papers

The goal of this study is primarily to extract ideas that may be of use for intent-driven systems. Therefore, we choose to not perform a thorough analysis of rigor.
as per e.g. Ivarsson & Gorschek [48]. Instead we focus on relevance, using the quality assessment criteria listed in Section 2.2.6. This classifies the papers into three levels, with level 1 containing the most relevant ideas for constructing an intent-driven system.

## 2.5 Conclusion

Currently, there is a lack of overview of available methods/techniques available for supporting business intents, and in particular, methods/techniques that cover the life cycle perspective of business intents. To address this, we conduct, in this article, a systematic literature review on methods and techniques that support intent-driven systems, and the aspects needed to enabling flexible realizations of intent-driven systems.

RQ1 is aimed at finding evidence in the literature for the support of intent-driven systems. RQ1 is divided into sub-questions, RQ1.1-RQ1.6, each covering one vital aspect of intent-driven systems. The aim of RQ2 is to find evidence in the literature of how the result of RQ1’s sub-questions could be used to enable flexible realization of intention driven systems. The results indicate that while there are methods and techniques, listed in Table 2.4, addressing each of RQ1’s sub-questions, the synthesis show no evidence of how these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems. For example, the governance of intent-driven systems as well as possibilities for semantic interchange of information is not evidently available in the literature.

The existence of methods/techniques which can be used as building blocks to construct intent-driven systems exist in the literature. How these methods/techniques can interact with the aspects needed to enabling flexible realizations of intent-driven systems is not evident in the existing literature.

This indicates a need for further research regarding semantic interchange of information, actor interaction in intent-driven systems, and the governance of intent-driven systems.

Our next step is to investigating how multi-actor governance for intent-driven systems may be realized.
Chapter 3

Supporting Continuous Changes to Business Intents
Chapter 3. Supporting Continuous Changes to Business Intents

Abstract

Context: Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the value networks in which the enterprise acts. This requires the use of policies and rules to guide or enforce the execution of strategies or tactics within an enterprise as well as in collaborations between enterprises. With the help of policies and rules, an enterprise is able to capture an actor’s intent in its business support system, and act according to this intent on behalf of the actor. Since the value networks an enterprise is part of will change over time the business intents’ life cycle states might change. Achieving the changes in an effective and efficient way requires knowledge about the affected intents and the correlation between intents.

Objective: The aim of the study is to identify how a business support system can support continuous changes to business intents. The first step is to find a theoretical model which serves as a foundation for intent-driven systems.

Method: We conducted a case study using a focus group approach with employees from Ericsson. This case study was influenced by the spiral case study process.

Results: The study resulted in a model supporting continuous definition and execution of an enterprise. The model is divided into three layers; Define, Execute, and a common governance view layer. This makes it possible to support continuous definition and execution of business intents and to identify the actors needed to support the business intents’ life cycles. This model is supported by a meta-model for capturing information into viewpoints.

Conclusion: The research question is addressed by suggesting a solution supporting continuous definition and execution of an enterprise as a model of value architecture components and business functions. The results will affect how Ericsson will build the business studio for their next generation business support systems.

3.1 Introduction

The enterprises of today are part of value networks. A value network is defined as “A set of connections between organizations and/or individuals interacting with each other to benefit the entire group [45]”. Enterprises in a value network can be seen as parts in a compositional system and are by themselves compositional
systems, also known as a system of systems. Compositionality is defined as “The evident ability of humans to represent entities as hierarchies of parts, with these parts themselves being meaningful entities, and being reusable in a near infinite assortment of meaningful combinations [39]”. The actors in a compositional system may be humans or machines.

By using software agents as machine actors enterprises can bring customers closer to suppliers of products and services, support the customers continued demand for change, inject further intelligence into enterprises and simplify the environment for both customers and employees [3]. Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the compositional systems the enterprise acts in. This requires the use of policies and rules to guide or enforce the execution of strategies or tactics in an enterprise as well as in collaborations between enterprises.

With the help of policies and rules an enterprise is able to capture an actor’s intent in its business support system, and act according to this intent on behalf of the actor. Alignment between the actor’s actual intent and, by the system, interpreted intent determines the success of the overall business transaction.

Policies and rules are originating from business intents on vision-, strategy- or tactic-level [69]. Knowledge creation is obtained through interactions between actors [68]. Intents are the aim and purpose resulting from knowledge creation regarding internal or external influences.

The construction of a compositional system requires methods to achieve a holistic collective benefit through the individual systems’ participation and cooperation when each system adopts a solution that maximizes its own self-interest [29]. The construction of such a system has to support changes to a system’s policies and rules in ways that are effective and efficient [33]. Achieving the changes in an effective and efficient way requires knowledge about the affected intents and the correlation between intents. The correlations between intents are affected by factors such as value network structure, decision process, and the actors’ responsibilities. It is essential for an enterprise in a value network to understand the correlations between intents, have knowledge about the correlation factors, and how to influence the correlation factors.

An intent-driven system is a supporting system of a business support system. The aim of intent-driven systems is to provide knowledge about the correlation factors, support intents’ life cycles and the correlation between intents. The life cycle support includes translation of intents into policies and rules as well as changes to and verification of the policies and rules. We use Pask’s conversation theory [78] as a model to describe intent-driven systems (see Section 3.2.1).
Chapter 3. Supporting Continuous Changes to Business Intents

On Pask’s conversation theory we define a context frame as the total domain information for the specific domain an actor has obtained. We consider an actor’s viewpoint to be based on one or several of the actor’s context frames.

Since the value networks an enterprise is part of will change over time, the business intents’ life cycle states might change. This means actors must be adaptable to the intents’ life cycle both from a business design and business execution perspective. The lack of mechanisms that can handle a business intent life cycle for enterprises seems to be the main challenge. The major problems are the lack of methods that could be used to express business intents and transform the business intents between different actors’ domains [110]. Most contributions are expressing and validating human actor intents. The focus is only directed towards one context frame with real world actors. This may be sufficient as a start but the transformation between different context frames is greatly unexplored.

To find guidance about how to realize support for continuous changes to business intents we have formulated the following research question:

*RQ: How can support for continuous changes to business intents be realized in business support systems?*

The contribution of the paper is a theoretical model which serves as a foundation for intent-driven systems.

This paper is organized as follows. In Section 3.2 the background is explained. In Section 3.3, background information about the case context, the study design, and the execution are presented. The result is presented in Section 3.4, the analysis is presented in Section 3.5 and a discussion is provided in Section 3.6. Finally, in Section 3.7, a summary and the conclusions are provided.

### 3.2 Background

This study is conducted as part of a design science [42] project together with Ericsson AB, in the area of business support systems. At the time of the study Ericsson was in an early pre-study phase of a business studio. The business studio is part of a business support system and will be used in the area of planning and monitoring business intents. The idea with the business studio is to deliver support for a 360-degree perspective of an enterprise’s business. The perspective includes both the actual execution of an enterprise’s business and the intended changes to this execution. The business studio will support and govern the decisions and actions needed to maintain or change the way an enterprise does business. Ericsson’s customers will be able to buy the business studio as a product or a service. With the help of the business studio, Ericsson’s customers
Chapter 3. Supporting Continuous Changes to Business Intents

will get support and knowledge about how to configure, monitor, and redesign their products and businesses. The software could be used in different business areas, for example charging, billing, customer relationship management, partner relationship management, order management, etc. The business studio supports the idea of continuous business-requirement engineering. Intent-driven systems are one of the cornerstones for the business studio product. The aim was to find architectural theories needed to realize a robust, but still flexible, software architecture for the business studio. The study is focused on the experiences gained during the time the subjects were investigating and elaborating the requirements needed to support the planning and monitoring of business intents. The first step is to find a theoretical model which should be practically evaluated at a later stage.

3.2.1 Definition of intent-driven systems and related work

An actor’s intent has to be communicated to other actors. During the interaction about the intent the actors have to prove their understanding of the intent in order to gain a common understanding and knowledge about the intent. This interaction can evolve over several steps and might re-shape the original intent. Together with Ericsson we are using Pask’s conversation theory [78] as a model to describe intent-driven systems.

Pask’s conversation theory is a cybernetic and dialectic framework that offers a scientific theory to explain how interactions lead to “knowledge”. It is part of the scientific disciplines cybernetics and psychology. The theory has evolved with the help of theoretical and related empirical work [36] and is used in different fields [40, 41, 13].

The idea of an intent-driven system as chains of interacting loops with one or several cycles is shown in Figure 3.1 (The original contributor of a figure is noted in the figure’s caption text.). Figure 3.1 shows how two human actors (A and B) interacts about an intent, using language $L$ regarding domain $D$. The interaction results in some sort of common understanding. Each human actor interacts with its respective machine actor to translate the intent from domain $D$ to domain $D'$, using language $L_{hm}$. The two machine actors ($A'$ and $B'$) interact on the intent, using Language $L'$ regarding domain $D'$, and the outcome is fed back to their respective human actor, using language $L_{mh}$. One or both of the human actors may not be satisfied with the outcome. This will render the start of a new cycle.

The language is defined as:

45
“L may be a natural, written or spoken, symbolic language, but it need not be. It may be a system of symbolic behaviors such as dance or actions such as key pressing. It may be formalized, as in, mathematics and higher level programming languages, but it need not be. It must however have many of the qualities of a natural language, with possibilities to express and interpret commands, questions, answers, obediences, explanations, or descriptions.” [56].

A domain could be described as:

“A domain is a collection of topic’s, and a topic is essentially a relation. This may be a very concrete relation (a relation between alphabetic characters and the keyboard positions in typewriting) or it may be an abstract relation (a relation between smugglers and the countries they operate in): To learn or solve a problem is to ‘bring about’ such a topic relation.” [56].

Pask [78] stresses the fact that the different actors have obtained their specific domain information through several different interactions which makes the model recursive. This means that each actor obtained its specific domain information in different contexts. A context has boundaries defined by who, what, when, where and how [5]. We define a context frame as the total domain information for a specific domain an actor has obtained.

Since intents seldom exist in isolation, their correlations to other intents need to be addressed. The correlations of intents are affected by factors such as value network structure, decision process, and actor responsibilities. The correlations and the factors affecting the correlation need to be governed.
Initiatives from different organizations are covering parts of what is needed in a business studio. For example, the TMForum’s eTOM [104] is one process map directed towards the telecommunication industry. The eTOM is focused on the business execution part of an enterprise. TMForum has included parts of ITIL [46] in eTOM. ITIL started as best practice processes and do not cover the business design parts of an enterprise. The OMG’s Business Motivation Model (BMM) [69] covers parts of the knowledge needed for the business design. The BMM’s Assessment could be seen as part of an enterprise’s business design processes. Some of the concepts which are used during enterprise interactions, are shared with a requirement engineering framework for virtual organizations [83].

The business intent realization builds upon collaborations in the form of interactions between the different actors. The interactions between the actors are negotiated since each actor has its own view of a business intent. The negotiation may result in a, to some extent, desired outcome. Combinations of interactions between more than one real world actor exists in the literature [23, 44, 52], but it is not evident that the combinations presented in the literature can be used in other domains due to the tight coupling between the realization and the information in each solution. Interactions are not only taking place between individual actors. The interactions between groups are an integral part of the SECI-model which is used for knowledge creation [68]. The interactions between groups are supported in Pask’s conversation theory [78].

The outcome of an interaction between human actors results in conclusions stated in natural language. From a software engineering perspective the conclusions can be made executable in the form of policies and rules but this would require Natural Language Processing, suggested in [110], and a formal way of expressing the policies and rules, e.g. using Semantics of Business Vocabulary and Rules (SBVR) [71]. Since SBVR is business agnostic semantics and ontologies are needed to give meaning to the policies and rules. The appropriate semantics and ontologies are not available in today’s enterprises [43].

Intent-driven systems have a need for governance since new or modified business intents introduce changes in the correlations between business intents. In order to govern all the business intents, each involved actor is required to share a common understanding of the used governance model. To achieve that, collaboration between different governance models could be used. Governance frameworks are discussed in, e.g. Wiesner et al. [110], Beigi et al. [7], and Lewis et al. [55].

From an enterprise perspective the context frame could be considered as a business process or an activity within a business process with a responsible actor. The business process is supported by business rules. Since TMForum’s eTOM
is a process map and TMForum’s TAM is an application map the different parts of the these maps could be used as part of a context frame. The columns (Why, How, What, Who, Where and When) in the Zachman Framework could be seen as an integrated part of the context frame.

### 3.2.2 An example of the initial steps for intent-driven systems

An example of the initial steps of intent-driven systems, supporting the business are shown in Figure 3.2. In Figure 3.2 we see how two stakeholders, A and B, representing a business side and a operational side, respectively, communicate with each other both directly and through computer agents, A’ and B’. Stakeholders A and B negotiate via a language L, regarding the domain D. Stakeholder A’s intents are expressed as business requirements, that are communicated to Component A’ via a language $L_{mhA}$. Component A’ interprets the intents as business processes and business rules that govern these processes. The results derived by Component A’ are presented to Stakeholder A in language $L_{mhA}$. This initiates a cycle where Stakeholder A interacts with Component A’ until a common understanding of the business intents are accurately described in $L_{mhA}$. Correspondingly, Stakeholder B’s intents are expressed in the form of the capabilities it aim to offer, and the job of Component B’ is to map these capabilities to machine and human actors that can together supply the offered capabilities. When a common understanding between the stakeholders and their related components are achieved the components are able to start negotiating with each other over language L’, regarding the domain D’. The results are presented back to their respective stakeholder in the form of effectiveness and efficiency measures. The stakeholders may then continue the negotiation via language L in an iterative cycle until an agreement can be made.

In order to obtain the effectiveness and efficiency measures, the components need to interact with their correlated components in the compositional system. These interactions may introduce changes to the system or involve new actors and components to be part of the decision process. The decision process’ level of automation of decision and action selection, depends on the involved components capabilities and rights of taking decisions and performing actions. In order to control and manage these capabilities and rights, governance views are needed. Parasuraman et al. [76] presents a scale of the “levels of automation of decision and action selection” ranging from 1 (The computer offers no assistance; human must take all decisions and actions.) to 10 (The computer decides everything, acts autonomously, ignoring the human.).
Pask [78] stresses the fact that the different actors have obtained their specific domain information through several different interactions which makes the model recursive. This means that each actor obtains its specific domain information in different contexts. A context has boundaries defined by who, what, when, where and how [5]. We define a context frame as the total domain information for a specific domain an actor has obtained.

![Diagram](image)

Figure 3.2: An example of initial steps for intent-driven systems (Contribution from the authors).

### 3.3 Methodology

To answer the research question, we have conducted a case study. The design of this case study was based on Runeson et al. [94] using a focus group approach [114, 89] in order to mimic the workflow used in the studied company. Information was captured in documents, drawings, and photos. We applied open coding technique [20] to analyze the collected qualitative data.

The case study was influenced by the spiral case study process [94] with an iterative character in order to stepwise adjust the goal and scope to the iterative findings. Reviews were performed by the members of the focus group or external reviewers. The members of the focus group were responsible for presenting the goals for the sessions. Since Ericsson was in an early pre-study phase of a business studio, adaptions to the spiral case study process in the area of company responsibility were required. After each iteration, the company became responsible for validating the presented material via reasoning. The outcome of the validation process was used as input to the goals and scope of the next iteration.
3.3.1 Case context

Ericsson was chosen since the company is regarded as the leader in various Gartner Magic Quadrants in the Operation Support System/Business Support System domain [38] and was part of a design science [42] project. The design science project was supported by Ericsson’s upper level management in business unit Support Systems and governed by a steering group from Ericsson. The steering group includes experts and specialist in the domains of business support system, operation support system, and business modeling. The particular section of Ericsson that was studied is part of business unit Support System, which is responsible for the development of the business support system offerings.

3.3.2 Study design

The study investigates what is needed in order to realize support for continuous changes to business intents in business support systems. The study is a single holistic case study [94]. The unit of analysis is Ericsson’s business studio.

Subjects

The subject sampling strategy has been to select a sample of roles involved in the construction of the business studio. A combination of maximum variation sampling and convenience sampling was used to select the subjects [79]. In total six subjects were involved in the study. All subjects attended the study voluntarily.

The study subjects and their roles and experiences are described in Table 3.1. The first three subjects were part of the focus group and the rest of the subjects participated in reviews and review meetings only.

Data collection

The data was collected during focus group [114] interviews with the appointed persons. During the interviews the information was captured on whiteboards, in PowerPoint documents, and directly in a document. The interviews were approximately two hours in length and were based on semi-structured focus group interviews [89] with the interview questions described in Appendix B.1.

The views of the subjects became available during the interviews and captured as written text and pictures. The validation and correction of the captured
Table 3.1: Description of the subjects’ roles (F=Focus Group member, R=Reviewer), subject id and experiences.

<table>
<thead>
<tr>
<th>Role</th>
<th>Id</th>
<th>Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>1</td>
<td>More than 25 years of experience in the telecommunication industry. The subject's engagement consists of various roles as an Ericsson employee, teacher, academic researcher and entrepreneur in the machine-to-machine area.</td>
</tr>
<tr>
<td>F</td>
<td>2</td>
<td>More than 15 years of experience in the telecommunication industry. The subject’s main engagement was to act as sales support and responsible for the implementation of business support systems sold to Ericsson’s customers in the EMEA (Europe, Middle East, Africa) region.</td>
</tr>
<tr>
<td>F</td>
<td>3</td>
<td>More than 25 years of experience in the telecommunication industry. The subject’s engagement consists of various roles as an Ericsson employee; manager, system designer, process and tools development responsible and business consultant.</td>
</tr>
<tr>
<td>R</td>
<td>4</td>
<td>More than 25 years of experience in the telecommunication industry mainly engaged in business strategy and business innovation both at Ericsson and at a telecom operator.</td>
</tr>
<tr>
<td>R</td>
<td>5</td>
<td>More than 15 years of experience in the telecommunication industry mainly engaged in product development work and driver of research activities at Ericsson.</td>
</tr>
<tr>
<td>R</td>
<td>6</td>
<td>More than 10 years of experience in the information technology industry mainly engaged in research and consultancy.</td>
</tr>
</tbody>
</table>

material was conducted with the members of the focus group on the same day as the interview was held.

Since the subjects have good knowledge about the Business Motivation Model, BMM, [69] it was selected as a foundation to explain how an enterprise’s business intents could be expressed and realized.

Analysis

The data were captured in a version-controlled document. During the analysis new insights about the data arose. In order to align the understanding of the material, additional meetings with the concerned subjects were held. To get a second opinion, reviews of the preliminary result were conducted with subjects outside the focus group. We applied open coding technique [20] to analyze the collected qualitative data. Open coding helped to find common terms and concepts as well as to find synonyms and hyponyms used by the study subjects. The first and second authors performed open coding and iteratively discussed its results. The aim with this coding approach was to agree upon and to find new terms or concepts during each meeting. These terms or concepts were linked to existing ones and their meaning were revised, if needed. From this information we could form the main concepts which became Section 3.4.1 - Section 3.4.3.

After each meeting, the results were updated and distributed for immediate feedback. This was done in order to validate our common understanding of terms and
concepts in an iterative way and to avoid ambiguity and misunderstanding.

3.4 Results

Below, the results obtained during the different phases (Appendix B.2) are presented. The results are divided into three sections, Section 3.4.1 - Section 3.4.3, each describing different aspects of the results.

3.4.1 The initial information from Ericsson

During Phase 1 of the study, the research question was explained and the response by the subjects was their introduction of Figure 3.3 which describes the design cycles of an enterprise. The text “Why”, “What”, “How” and “Characteristics of How” was initially not part of Figure 3.3. The text was added during the study to be able to explain Figure 3.3 in a better way.

The “Why” row in Figure 3.3 outlines the five different phases of the design cycles. A business intent might not go through all the phases if it is not fulfilling the requirements of one or several of the phases. In each phase there is a feedback, which makes it possible to adapt a business intent to the possibilities foreseen.

“Every business intent is going through the strategy to operate phases.” Subject 1.

“The horizontal gray arrows\(^1\) represent the life cycle changes of the business intents on different levels in the enterprise.” Subject 3.

The statement:

‘The design and the delivery of an enterprise’s business intents have different characteristics regarding volume, speed, quality, etc.” Subject 1.

is illustrated by the characteristics of the “How” columns in Figure 3.3. Characteristics refers to “Showing the special qualities or traits of a person, thing, or group [64].”

The layers in the “What” indicates a top-down approach with fast feedback loops.

\(^{1}\)The gray arrows are part of ”What”, our note.
Chapter 3. Supporting Continuous Changes to Business Intents

Figure 3.3: The initial picture from Ericsson with explanations included (Contribution from the subjects).

“The vertical gray arrows show the effect of changes between different levels of the enterprise.” Subject 1.

The overview explanation to the dynamics in Figure 3.3 was; the enterprise can be seen as an OODA-loop [87] (Figure 3.4) on several levels where each action starts a new loop on another level.

“One can regard the gray arrows as interacting OODA-loops.” Subject 3.

The feedback from each loop will be an influence to another loop. The design and delivery environment are shown as the vertical boxes, “How”, in Figure 3.3. By separating between the design and the deliver environment, the enterprise can be handled in a multi-dimensional structure where each part of the structure needs governance and software support for itself.

Figure 3.4: A simplified version of the OODA loop [87] (Contribution from the subjects).
Chapter 3. Supporting Continuous Changes to Business Intents

The governance views are the main parts of a business studio where the different aspects of the business could be viewed, simulated and managed. This includes history, the present as well as the future of the business. The governance views should support guided decision support in order to sustain informed decisions with good effectiveness and efficiency.

The three main phases of governance are configuration, simulation and feedback:

Configuration needs to get binding rules, which could be precise or partly specified, and in that case it is left up to the configuration engine to compose those rules. This makes it possible to leave some configuration open until a decision is needed/desired. This phase has to cater for the version control as well as for the management of the updates. The points of measurement are setup here as well.

Simulation has to provide possibilities to test the intentions and take feedback into consideration when applicable. Another important aspect of the simulation phase is to gain knowledge about the different parts of the system. There are many reasons for this but one very important aspect is to identify unexpected behavior in the system and give guidance to detect the reason for this.

Feedback makes sure that the control of how the intentions are fulfilled by the system could be analyzed including monitoring of the different components’ and the systems’ state. This information will be part of some of the simulation, when feasible.

Definition 3.1: Governance view

Governance as a concept was explained by defining different views. The text was reviewed and updates to the text were done jointly. The final text is shown in Definition 3.1.

The aim with Figure 3.3 is to create relevant business values for the enterprise. A value architecture [28] was selected as a concept to represent the business intents.

“The business values relevant for an enterprise should be captured within a value architecture.” Subject 3.

The structure of the value architecture was found by examining the “Why” row in Figure 3.3, which consists of the five main parts Strategy, Design, Implement, Deploy and Operate. Initially, the main parts where called phases.

“The value architecture should be structured in five main parts Strategy, Design, Implement, Deploy and Operate.” Subject 3.
Chapter 3. Supporting Continuous Changes to Business Intents

Table 3.2: Mapping between Value Architecture and BMM.

<table>
<thead>
<tr>
<th>Value Architecture Component</th>
<th>BMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strategy</td>
<td>End</td>
</tr>
<tr>
<td>Design</td>
<td>Means</td>
</tr>
<tr>
<td>Implement</td>
<td>The values guiding the construction of Organization Units, Business Processes and Directives.</td>
</tr>
<tr>
<td>Deploy</td>
<td>The values guiding the introduction of Organization Units, Business Processes and Directives.</td>
</tr>
<tr>
<td>Operate</td>
<td>The values guiding the use of Organization Units, Business Processes and Directives.</td>
</tr>
</tbody>
</table>

The “Why” row in Figure 3.3 was named value architecture layer and the parts were renamed to value architecture components. These components were mapped by the subjects, during the following discussion, to the BMM [69] as shown in Table 3.2. The mapping reflects the statement:

“Every business intent is going through the strategy to operate phases.”

Subject 3.

The value creation is enabled by the business functions. A business function is a function that “Delivers business capabilities closely aligned to an organization, but not necessarily explicitly governed by the organization [73]”. A capability is defined as “An ability that an organization, person, or system possesses. Capabilities are typically expressed in general and high-level terms and typically require a combination of organization, people, processes, and technology to achieve [73].”

“The business functions are where the value creation is taking place.”

Subject 2.

In Figure 3.3’s “What” part, the value creation occurs in four high level business functions: “Marketing, Sales & Customer/Partner Interaction”, Business, “Enterprise Plan, Build & Optimize” and “Delivery & Production”. The Marketing, Sales & Customer/Partner Interaction are a set of business functions responsible for the interaction with parties outside the enterprise. The business function named Business is responsible for developing, maintaining, and terminating the way the enterprise do business. The Enterprise Plan, Build & Optimize is responsible for the enterprise’s capability to realize the intended business. The business function Delivery & Production are responsible for producing and delivering the products (goods and/or services) to the enterprise’s customers.

The business functions used in Figure 3.3 are only one possible example of an enterprise’s possible business functions. The number of business functions and the number of levels are specific to each enterprise.
3.4.2 Supporting viewpoints

The conclusion that there is a need for a layered architecture arose from the statement:

“It is a misunderstanding that the strategy is not executed on a daily basis. Everything is executed continuously, even the improvement work.” Subject 1.

The design and the delivery parts in Figure 3.3 were seen as two different layers. To avoid confusion and introduce a more precise definition, these layers were renamed to Define and Execute.

How the business intents should be defined and executed, and how the life cycles are managed, are supported by a number of business functions.

“The parts of the business intents are built by business functions.” Subject 2.

The viewpoints became visible when Figure 3.3 was transformed into Figure 3.5. The viewpoints are the intersections between value architecture components and business functions (The squares in Figure 3.5).

“Each intersection could describe different viewpoints of the enterprise.” Subject 3.

These viewpoints could be extended to fit the structure of the enterprise and the required number of business intents.

With the help of Pask’s conversation theory [78], the discussion led to a developed understanding of viewpoints as a grouping of a number of actors. The subjects proposed that different actors are responsible for managing a business intent’s birth and death.

“Each business intent’s life cycle needs to be handle by the responsible actors and as a part of the whole enterprise. Each business intent needs at least two actors responsible for performing the needed tasks and for taking care of the life and death of the business intents. The life and death are handled by different actors.” Subject 1.

To cover the responsibility of the actors, the subjects proposed RACI [26]. RACI is an abbreviation for Responsible, Accountable, Consulted and Informed. These are the responsibilities and actors could be assigned for a specific task. This
supporting continuous changes to business intents

Figure 3.5: The initial visualization of the viewpoints as the intersections between a value architecture component and a business function (Contribution from the authors).

...could be structured in the form of a matrix, supporting the mapping of viewpoints.

"The responsibilities of an actor could be viewed with RACI.” Subject 3.

It was decided to put more emphasis on the interaction between Define and Execute. The use of the OSI model [47] helped evolve Figure 3.5 into Figure 3.6. The transitions between the Define layer and the Execute layer were inspired by the OSI model and will create life cycle status change for each impacted viewpoint. Effective and efficient transitions require supporting methods. A bigger framework is necessary for larger transformations that form a transformation project, e.g. TOGAF [72]. TOGAF is a framework used for improving enterprise architectures. The framework supports the tasks needed in a transition project.

"When moving from Define to Execute there need to be some sort of change management support. Could we use something like TOGAF?” Subject 2.

Figure 3.6 describes the multifaceted problem of developing a business intent into a working solution. This is a collaborative knowledge creation process which could be described as a “spiral” of organizational knowledge creation [67]. The two key life cycles called Define and Execute, and governance views are the focal points here. A business intent is developed in each key life cycle by iterations through the components in the value architecture layers (Strategy, ..., Operate which were discovered in Figure 3.3). During the iterations, the business intent’s life cycle
is affected in terms of participating business functions (actors). One business function could be involved in one or several value architecture components, often represented by a specialized actor. These business functions need to interact and negotiate to achieve a definition and a corresponding execution of the business intent. How to perform the interaction and negotiation between the key life cycles on value architecture components (indicated by the horizontal arrows) was inspired by an OSI model approach of layer interactions. By using the OSI model approach, the interaction and negotiation between the affected business functions, on each component in the value architecture layers, is based on the agreements in the preceding value architecture component. Impacts and dependencies to other business intents have to be considered as well which will introduce the need for governance views (Definition 3.1). The aspects of the governance views are part of Figure 3.6.

Figure 3.6: A model providing continuous definition and execution of business intents in a governed way (Contribution from the subjects).

In Figure 3.7 we can see how different business functions (actors) collaborate regarding business intents. The business intents resulting from the collaboration can be found in the gray shaded middle part in Figure 3.7. We propose to view the realization of the business intents as virtual business functions of its own since their business intents, to some extent, deviates from the business intents of the involved business functions. This might result in different interpretations of the business intents. The curved arrows (c) in Figure 3.7, show how the collaboration’s definition is constrained by the business functions’ execution. How much the collaboration’s define layer can influence the business functions’ define layers depends on the power structure (v) between the involved parties. The power structure decides how well the collaboration’s business intents could be fulfilled, here indicated by the horizontal arrows (t) at the top of the figure. The success of the business intents’ definitions is dependent on the interactions between the different business functions’ Define and Execute as well, indicated by the vertical arrows (v). The execution of the collaboration, indicated by the horizontal arrows (b) at the bottom of the figure, is always dependent on the
business functions since the collaboration’s execution is constrained by the sum of the business functions’ execution. This is a recursive pattern which could be used for collaborations between business functions on any level as well as between enterprises. The recursive pattern is valid for enterprise collaborations since enterprises and business functions could be considered as different views on an organization, for example an outsourced business function could be an enterprise in its own rights.

![Diagram](image)

Figure 3.7: The model in Figure 3.6 supporting business interactions and negotiations between business functions (Contribution from the authors).

The Execute layer is responsible for continuous execution of business intents. The Define layer supports changes in the environment and changes to business intents by providing a continuous re-definition of the business intents. The execution speed differs between the different layers. The Define layer might require simulation capabilities while the Execute layer might need near real-time responses.

A business support system must cover all the affected viewpoints in order to manage the life cycle of business intents. A generic meta-structure is needed since the actual list of viewpoints is dependent of the instantiated business support system. Ericsson’s business studio is part of the Define layer and responsible for the governance (see Definition 3.1) of the business intents. There was a common concern about how the different viewpoints and their information could be handled, since a large amount of viewpoints were anticipated.

“There is a need for managing the information connected to each viewpoint. Even if there is an appointed actor some kind of software support is needed.” Subject 2.

The discussion around viewpoint information is covered by the next section (Section 3.4.3).
3.4.3 Viewpoint information

The idea of looking at the problem as a real world interaction between humans was inspired by Pask’s conversation theory [78] and Erlang [2]. The communication between actors could be described as events. How these events are treated depend on the role an actor has. An actor has to use its knowledge to gain understanding of the events’ meaning and which activities that should be triggered by an event. This is supported by a set of rules attached to the actor’s role. The interaction between actors could be described as:

- One Actor sends an Event to another Actor. The receiving Actor observes the Event and applies its Knowledge on the Information in the Event and the context in which the event was sent. When the Event is understood, Rules are applied in order to decide how to act on the Event. Activities selected by the Rules are performed and, maybe, a response Event is sent back and, maybe, other events (including a response event) are triggered.

- The Event can use different types of channels, e.g. face-to-face, phone, post, e-mail, chat etc. The different types of channels will make it possible for the Actors to apply different and/or limited sets of their knowledge in order to respond.

The context frame was formed with the help of Pask’s conversation theory [78]. We defined a context frame as the total domain information for a specific domain an actor has obtained. The context frame’s meta-model is shown in Figure 3.8. In Figure 3.8 the four areas in a context frame is shown. The exchange is the format and the information obtained or delivered during an interaction. The activity is the ability to decide and act on events. The knowledge is the understanding of the information obtained during the interactions, in the form of events, and how this information is related. The rules put constraints on the other three areas as well as on the interactions between these areas. The rules could be constraints put on the context frame in a strategic or tactic purpose. An actor’s viewpoint could be based on one or several of the actor’s context frames.

We suggest to use a compositional system of context frames to represent an actor’s knowledge in a certain domain. An example of a generic context frame for a compositional system is depicted in Figure 3.9. In the context frame meta-model, context is a part of knowledge. The context frame describes and encapsulates what, how, and why the context should be used, and to some extent, also where, when, and with whom.

It is essential to notice that one context frame could be associated with several other context frames. These context frames could have overlapping rules, knowledges, events, and activities. This brings us to the level of compositional context...
Chapter 3. Supporting Continuous Changes to Business Intents

Figure 3.8: The Context Frame meta-model (Contribution from the subjects).

frames.

The composite context frame in Figure 3.9 is used for capturing the recursive nature of an actor, interactions between groups of actors as well as for meta-level structures. Figure 3.9 shows the different perspectives of a composite context frame. These perspectives are the actor internal perspective, the actor external perspective, and the meta-level perspective. In the actor internal perspective, the composite context frame could be seen as one of an actor’s domains with several subdomains, or as an actor with several domains. In the actor external perspective, the composite context frame could be seen as a group of interacting actors which forms a Ba [68]. In the meta-level perspective the composite context frame could be seen as the structure of meta-levels, group of actors, or actors governed by a meta-level.

Figure 3.9: The Composite Context Frame meta-model (Contribution from the subjects).
3.5 Analysis

During the analysis we created Figure 3.2, which describes the initial steps of intent-driven systems. This figure helped us to combine the different artifacts into an architectural model which is possible to implement. How the different artifacts are combined is explained by the example below.

When changing or introducing a business requirement, e.g. changing the charging interval for a customer segment, several business processes and actors, both humans and machines, might be identified as affected by the requirement (Figure 3.2). These processes and actors might affect business functions responsible for legal aspects (Does the existing contracts allow changes to the prices?), marketing & sales (How can this be presented to new and existing customers?), invoicing (How can new pricing rules be calculated and presented?), customer care (What compensations could be given to complaining customer and are there a special treatment of the segment?) as illustrated in Figure 3.6. A realization of the requirement requires collaborations between the affected business functions, as illustrated in Figure 3.7, with the help of a language L in Figure 3.2. The outcome of the collaboration might require changes to a process’ activities or the process itself. Since processes and activities can be implemented with the help of humans or machines, the arrows marked with “v” in Figure 3.7 and the horizontal arrows in Figure 3.6 will make use of the functionality described by Figure 3.2. This unfolds the recursive nature of Figure 3.2.

We build upon Pask’s conversation theory [78] by introducing the context frame, Figure 3.8. We define a context frame as the total domain information for a specific domain an actor has obtained. The context frame has a compositional character, see Figure 3.9, to be able to represent how an actor has obtained knowledge in a certain domain. The context frame gives the possibility to support continuous changes to business intents by evaluating the risk and the value of the proposed changes.

The results of the analysis were discussed with the subjects in the focus group. It was decided to continue the design science study in order to investigate how the initial steps of intent-driven systems (Figure 3.2) could be realized.

3.6 Discussion

This study introduces a model of value architecture components and business functions, divided into the Defined and the Execute layers. This model mimics the reality of an enterprise. To support the model we build upon Pask’s conversation
theory [78] by introducing the context frame. This makes it possible to support continuous definition and execution of business intents supporting the enterprise and its value networks. To the best of our knowledge this ability is not presented in the literature.

The process of introducing business intents to an existing environment is many times slow and error prone. The number of formal and informal business intents in a large scale software intensive enterprise is often very large. The most common form of distributing business intents to various stakeholders is through natural language requirements. The artifacts in Figure 3.2 can be ranked in descending order of expressiveness starting with the machine actors. Machine actors are often described by interface specifications but these specifications sometimes missing useful meta data, e.g. pre-, post-conditions and invariants, the meaning of the attributes, and do not adhere to standardized semantics. Business requirements are expressed in natural language. Business processes are expressed in natural language, sometimes supported by a graphical notation to express the details of the process, e.g. Business Process Model and Notation [70]. Business rules often derived from requirements which are expressed in natural languages. A description of the human actors’ capabilities which is useful for fulfilling business intents are seldom in place but this could be improved with the help of Expertise Retrieval [6]. To obtain the possibility of a high level of automation of decision and action selection [76] the information acquisition has to be effective and efficient. This requires the possibility to automatically apply rules and constraints during the information acquisition to improve the sender’s and the receiver’s understanding of the information. These rules and constraints have to be governed with the help of governance models.

To govern all the business intents requires each involved party to share a common understanding of the used governance model. Since several actors have to collaborate to fulfill a business intent it is not realistic to assume the use of one homogeneous governance model. To achieve a common understanding of heterogeneous governance models, collaboration between different governance models could be used. This collaboration requires appropriate semantics and ontologies which are not available today [43].

To be able to understand the possible impact of a business intent, textual descriptions are not enough in a large scale software intensive enterprise. The support from visualizations which can make a multi dimensional business intent understandable for humans is needed. The possibility to investigate changes to business intents with the help of visual interactions is desirable.
3.6.1 Validation of the theoretical models

We have investigated if the theoretical models are possible to use in practice. In order to achieve this we have chosen to investigate a limited part of an enterprise's business rules. The business rules we have chosen to study are targeting recommendation support for value propositions, based on different business models. The recommendations are based on following five parts of the Osterwalder canvas [74]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions.

In order to verify a simplified version of the initial steps of intent-driven systems (Figure 3.2) we have implemented a machine learning pipeline based on the ID3 algorithm [84]. The pipeline is considered as a proof of concept, and as such is regarded as successful by the involved practitioners. The pipeline makes it possible to, visually and logically, validate the correctness of the business rules before they are put in production. The possibility to generate executable code representing the model of the business rules makes it possible to execute the same model in different components without the need of re-implementation. This might improve the coherence of the business rules in a business support system.

The Design in Figure 3.6 is supported by the possibility to, visually and logically, validate the correctness of the business rules before they are put in execution. The Execute is supported by the possibility to logically validate the correctness of the business rules before they are put in operation, and to deploy and operate the executable business rules as a context frame meta-model (Figure 3.8. The Governance views are supported by the fact that the executable business rules can be handle as immutable artifacts.

The proof-of-concept is described in Appendix B.3.

3.6.2 Validity threats

The methodology suggested by Robson [89] to analyze threats to the validity and the corresponding counter measures is used.

Author bias

An extensive experience as an industry practitioner may have influenced the aims of the study with a stronger bias towards solutions. To avoid the risk of imposing solutions from the authors, member checking [89] was used continuously in the form of review meetings at different stages of the study. Moreover, applying
open coding technique [20] to analyze the collected qualitative data, fostered a focus on the merit of each interview session before a solution perspective could be evaluated. A colleague external to the project evaluated the process.

**Interpretation**

To mitigate the risk of imposing a meaning on what is happening an objective interpretation is needed. Triangulation was used to prevent that a single viewpoint of the information was gathered and presented during the study. Data triangulation was achieved by using interviews, informal meetings, continuous member checking [89], and subjects with different roles and responsibilities. This, together with a distinct audit trail, and the use of an open coding technique [20] ensured an objective interpretation. The need for observer triangulation was regarded as small since review meetings were held as soon as possible after a meeting.

**Biased theories**

The negative case analysis [89] was used to challenge the theories and counter researcher bias. The theories were presented to and reviewed by the subjects and the authors.

**The use of focus groups**

When using focus groups the facilitating of the group process might be a problem. Biased views and conflicts due to dominant subjects may influence the result. The size of the group made it easy to guide the sessions. The subjects are senior in their professional roles and are willing to share and listen to each other.

**Adaptions to the spiral case study process**

In order to stepwise adjust the goal and scope to the iterative findings, adaptions to the spiral case study process [94] were made. These adaptions were in the area of company responsibility. A senior researcher external to the project and knowledgeable in the case study methodology evaluated the adaptions made to the spiral case study process.
3.7 Summary and Conclusions

This paper presents a case study conducted at Ericsson with the aim to support business intents in a business support system. At the time of the study the intended solution was in its initial stage. In order to support business intents, the possibility to express viewpoints and information capturing mechanisms, became the focus of the study.

A continuous definition and execution of a business intent’s life cycle in an enterprise and its value networks, could not be found in the existing literature. Nor did we find a meta-model supporting a context frame aware realization of a business intent’s life cycle in a compositional way. The research question is addressed by suggesting a solution supporting continuous re-definition and execution of an enterprise as a model of value architecture layers and business functions (Figure 3.6). Figure 3.6 is anchored in the focus group through an iterative and incremental series of workshops. The model in Figure 3.6 is supported by Figure 3.2 which describes the initial steps of intent-driven systems. Figure 3.2 encapsulates the context frame meta-model (Figure 3.8 and Figure 3.9) which captures the information in the viewpoints. The results will affect how Ericsson will build the business studio for their next generation business support systems.

Together with Ericsson we will, as a next step, continue to investigate how intent-driven systems could be realized.

Acknowledgment

We thank the reviewers for their valuable comments.
Chapter 4

Encouraging Business Flexibility by Improved Context Descriptions
Chapter 4. Encouraging Business Flexibility by Improved Context Descriptions

Abstract

Business-driven software architectures are emerging and gaining importance for many industries. As software-intensive solutions continue to be more complex and operate in rapidly changing environments, there is a pressure for increased business flexibility realized by more efficient software architecture mechanisms to keep up with the necessary speed of change. We investigate how improved context descriptions could be implemented in software components, and support important software development practices like business modeling and requirement engineering. This paper proposes context descriptions as an architectural support for improving the connection between business flexibility and software components. We provide initial results regarding software architectural mechanisms which can support context descriptions as well as the context description’s support for business-driven software architecture, and the business flexibility demanded by the business ecosystems.

4.1 Introduction

Business requirements are shaped by collaboration and continuous knowledge creation [67] between stakeholders, who are driven by intents while acting in business ecosystems. The business requirements and the speed of implementing them become the dominant concern for software intensive product development (SIPD) [9] companies and forces these companies to reach new levels of agility and orchestration of digital resources.

For SIPD, this challenge translates into creating efficient software architectures which support business flexibility in order to adapt existing business models or support new business models as a response to changes in the business ecosystems. Software components are often expensive to re-use and maintain in new or multiple business models due to a lot of business logic connecting various components while functions are hard-wired to certain business environments. Components cannot be re-coded every time a business model changes. Therefore, new software architectures need to support the complete lifecycle of connecting business models to software components with an efficient support for changeability.

This paper focuses on supporting business flexibility by using specific context descriptions. The aim is to transform these context descriptions into executable containers which could be used to support the needed business flexibility. The remaining part of the paper is structured as follows: Section 4.2 presents background and related work, Section 4.3 and 4.4 provide information about how the
idead have been used by practitioners, and Section 4.5 concludes the paper.

4.2 Background and Related Work

Business architecture flexibility focuses on business trade-offs that need to be quickly resolved and how they impact both function layers and realization layers. Depending on the estimated future value for relevant stakeholders [51], the business architecture flexibility allows for agile changes to the realization layer. Availability and flexibility are recognized as important aspects in high uncertain business environments [88].

The transition to service driven economy has given the birth to Industrial Product-Service Systems [62] with the focus on lifecycle-integration of products and services. New possibilities for capturing value as well as for “on-demand lock/unlock” of business value options are possible with the digital delivery of software and value. This requires the software components to support new levels of flexibility for option-locking support, including governance.

Several significant contributions have been made in decomposing value for software products [50] or describing industrial context in software engineering [81]. Castro et al. focused on bridging the gap between the software systems and their operational environment using i* modeling framework [18] leveraging on goal based modeling. Goal based modeling of requirements and agent-based software engineering are common approaches to capture the requirements on software components, e.g. KAOS, MAS, and TAO [101]. However, in practice these frameworks still lacks usability [27] in which industry can effectively and efficiently industrialize these practices and develop efficient software architecture.

Creating software components that can be orchestrated and bring value to the relevant stakeholders in business ecosystems and timely respond to frequent changes remains the main challenge. This is partly addressed by Software product lines [12] and industrial Product-Service systems [62] which focused on changeability [88], as ways to create flexible, adaptable and efficient component-based software architectures.

Supporting business flexibility requires support for agile business policies and business rules [16] which are used to govern how an enterprise does its business [69]. It is desired to have a common governance structure and a standardized way of handling the business rules. Rosca et al. have contributed valuable knowledge in the area of common governance of business rules [90, 92]. However, we use to the more declarative nature of the business rules [16].
To fully support business flexibility, we need to better understand and define the business context. Modeling context is also critical for developing context-aware software systems [15]. Baldauf et al. [5] summarized context-aware systems including methods to achieve context-awareness, e.g. resource discovery, sensing, context model, context processing, hierarchical context data. Despite several similarities, context-aware software systems focus on dynamically discoverable services rather than dynamically changing business opportunities. This paper builds upon the definition provided by Baldauf et al. [5] and introduces context description and context frame as concepts for achieving context-aware business architectures.

By composition, context descriptions can be used to create efficient re-usable descriptions that can be used not only in business requirements but also in business rules. The context description is what gives a context frame a scope (boundary) and defines a meaning (semantics). In this paper, we propose the context frame as a fundamental building block in new software business architectures to create self-adaptive software components that can understand, negotiate and adapt to a business context.

4.3 Case Context

In today’s implementations of business support systems, business rules are configured in different places of the system, and in different formats. This makes it hard to have a common view of what is defined, and to execute the same logic in different parts of systems, without re-implementing the rules.

Since humans are defining the business rules, these rules are usually ambiguous. Visual and logical support to verify the correctness of the defined business rules are desired. Parts of the business rules could be made executable in order to make the operation of the business more efficient and effective. The process of translating business rules to executable business rules is error-prone due to human interpretation. Sharma et al. [98] have proposed a method to find business rules in requirements documents. This is a good start but most business rules are not about the information system itself but rather about the business the information system shall support.

It is desired to use a software algorithm to translate business rules into executable business rules. A way to execute business rules is to use a common rule engine for all the components in a business support system. This approach might not be desired or possible. Instead, the possibility to express executable business rules in a different software language, which could be distributed to the different
components in a business support system, might be an option. Contrary to many proposed solutions, we believe that a business rule could be triggered by several different events. This makes the use of a simple event-condition-action architecture not suitable for the problem at hand.

Together with Ericsson we have performed a proof-of-concept to investigate if it is possible to support visual and logical verification of business rules, and to generate executable business rules. We have chosen to investigate a limited part of an enterprise’s business rules. The business rules we have chosen to study are targeting support for value propositions, based on different business models. The business rules are based on following five parts of the Osterwalder canvas [74]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions.

4.4 Proof of Concept

Since a business model is supported by a set of legal contracts, we started to derive the business rules from these type of contracts. Some of the information in a legal contract is not meaningful to translate into a business rule which should be executed in software, e.g. which country laws should be used to solve a dispute. Many times the nature of the language used in legal contracts requires human interpretation. However, the majority of the terms and conditions in a legal contract can be translated into meaningful business rules which could be implemented in software.

We have implemented machine learning pipe-lines which make it possible to conduct visual and logical verification of business rules, and generate executable business rules. This process can be regarded as the creation of a context frame. We have added different types of functionality which is regarded as needed when handling business rules. Missing data is handled as a wild card, i.e. all values are true. Continuous values have a defined boundary and there are no value gaps in the data. Since a human is defining the business rules, entering all possible combinations by hand is not an option. A meta-data file is used to describe the nature of the features.

In order for the solution to exist in an event-driven environment, the extracted business rule was extended with the events it is intended for. We have added the possibility to use two classification columns. These classification columns respectively represents eligible objects and the allowed actions on the business rules. The idea is borrowed from the gaming industry where a specific context gives the character the possibilities to, for example find specific treasures and stipulates
how these treasures can be handled. The combination of event, eligibility and action makes it possible to mimic a business process.

There is a strong demand on the possibility to separate the design from the execution and the need for governance of the business rules throughout their lifecycle. This demand is in-line with TMForum’s eTOM [104]. The design is supported by the possibility to, visually and logically, validate the correctness of the business rules before they are put in execution. The execution is supported by the possibility to logically validate the correctness of the business rules before they are put in operation, and to deploy and operate the executable business rules as a context frame. The governance views are supported by the fact that the executable business rules can be handle as immutable artifacts.

The pipe-lines are considered as a proof of concept, and as such is regarded as successful by the four practitioners involved in its evaluation. Three of them are system architects and one is a business support system expert. The pipe-lines make it possible to, visually and logically, validate the correctness of the business rules before they are put in production. Generating executable code representing the model of the business rules, makes it possible to execute the same model in different components without the need of re-implementation. This might improve the coherence of the business rules in a business support system. It was concluded that this way of supporting business processes can support the business models of the business support system itselfs, as well as the business models of the enterprises which are running their business with the help of the business support system.

There are several improvements to the pipe-lines which should be considered. The precision of feature value has to be configurable feature by feature, and with different values for the maximum limit and minimum limit. The executable code representing the model of the business rules should support additional languages, for example Java™. The ability to extract information from the legal contracts must be improved. We will investigate how we can leverage on the research made in the area of common governance of business rules [90, 92]. During the proof-of-concept we elaborated with the logical visualization of the deployed business rules. We plan to investigate if a graph database can support the needed deployment capabilities regarding visualization and semantics.

There are no real-time requirements on the transformation from business rules to executable business rules. Since the data set will vary in the number of used feature columns and since each feature has its own characteristic, the use of a typed-language is not ideal. Based on this, Python fulfills the requirements as a suitable implementation language for the problem at hand.
4.5 Conclusions and Further Work

This vision paper illustrates the potential benefits with introducing context descriptions and context frames to support business flexibility. The implementation makes it possible to, visually and logically, validate the correctness of the business rules before they are put in production. The possibility to generate executable code representing the model of the business rules, makes it possible to execute the same model in different components without the need of re-implementation. However, the PoC does not include support for business requirements and the software architecture mechanisms supporting the context frame are very basic.

Together with Ericsson we plan to improve the solution in order to make it useful in a business support system offering. This includes the improvements discussed in this section and in Section 4.4.

Acknowledgment

This work was partially supported by Ericsson AB. We thank the members in our former units for their support and the time they spent with us during this work.
Chapter 5

Towards Executable Business Rules
Chapter 5. Towards Executable Business Rules

Abstract

Context: In today’s implementations of business support systems, business rules are configured in different places of the system, and in different formats. This makes it hard to have a common view of what is defined, and to execute the same logic in different parts of systems. It is desired to have a common governance structure and a standardized way of handling the business rules.

Objective: To investigate if it is possible to support visual and logical verification of business rules and to generate executable business rules.

Method: Together with practitioners we conducted an experiment.

Results: We have implemented a machine learning pipe-line which supports visual and logical verification of business rules, and the generation of executable business rules. From a machine learning perspective, we have added the possibility for the ID3 algorithm to use continuous features.

Conclusion: The experiment shows that it is possible to support visual and logical verification of business rules, and to generate executable business rules with the help of a machine learning pipe-line.

5.1 Introduction

Business rules [16] are used to govern how an enterprise does its business. Since humans are defining the business rules, these rules might be ambiguous. Visual and logical support to verify the correctness of the defined business rules are desired. Parts of the business rules could be made executable in order to make the operation of the business more efficient and effective. The process of translating business rules to executable business rules is error-prone due to human interpretation. It is desired to use a software algorithm to translate business rules into executable business rules. A way to execute business rules is to use a common rule engine for all the components in a business support system. This approach might not be desired or possible. Instead, the possibility to express executable business rules in different software language, which could be distributed to the different components in a business support system, might be an option.

We have evaluated if a machine learning pipe-line could be used to mitigate the problems stated above. The pipe-line consists of three parts. A pre-processing part, the use of a decision tree algorithm called ID3 [84], and a post-processing part.
Chapter 5. Towards Executable Business Rules

The pre-processing is responsible to prepare the business rule data for the processing done by the ID3 algorithm. Since the ID3 algorithm cannot handle continuous data the pre-processing makes it possible to mimic continuous data in a way that the ID3 algorithm can handle.

The ID3 algorithm is implemented according to the base algorithm. We have added a warning in the ID3 algorithm in order to notify the user if there might be ambiguities in the business rules. Normally the ID3 algorithm will use majority voting and continue with out any further notice.

The post-processing consists of three steps. The first step is responsible for visualizing the result from the ID3 algorithm in a tabular format as well as in a tree format. The second step will create an executable model of the business rules. The third step will use the executable model for the logical validation of the business rules. Since the model is stored in an executable format, the model could be distributed to components in a business support system.

The pipe-line has proved to be valuable in order to mitigate the problems stated above. Another contribution from this study is the possibility to use the ID3 algorithm with continuous values.

5.1.1 Background

Knowledge creation in enterprises is obtained through interactions between actors [68]. Intents are the aim and purpose, resulting from knowledge creation regarding internal or external influences. Business intents are often described in the form of activities and events, which are governed by business rules.

In today’s implementations of business support systems, business rules are defined in different places of the system, and in different formats. Sharma et al. [98] have proposed a method to find business rules in requirements documents. This is a good start but most business rules are not about the information system itselfs but rather about the business the information system shall support. The formats of the business rules are ranging from text in documents to software language code. Since some business rules are valid for several components, the same logic is implemented several times but in different ways. This makes it hard to have a common view of what is defined. Changes or additions to the business rules might involve re-implementation of code which is time consuming and error-prone. Instead, it is desired to have a common governance structure and a standardized way of handling the business rules. Rosca et al. have contributed valuable knowledge in the area of common governance of business rules [90, 91, 92].
Chapter 5. Towards Executable Business Rules

The nature of the business rule data can range from, for example integers, product identifiers, date and time intervals, etc. This requires handling of numerical-, categorical-, and nominal-data. By introducing meta-data it is possible for rule executers to understand the syntax and semantics of the data. A base for the syntax and semantics could be an extension to the W3C’s XML Schema Part 2 [107]. Explicit boundaries for the numerical data is needed.

The execution of business rules should be distributed to different rule executers. The rules executers are used to guide or create business processes which are responsible to adhere to these rules.

The research project is done in collaboration with Ericsson and is based on the design science methodology [42]. Ericsson is a major player in the area of business support systems. It was decided to investigate if the field of machine learning could be used to simplify the governance of business rules.

5.1.2 Research Question

We would like to investigate if it is possible to support visual and logical verification of business rules and to generate executable business rules. We have chosen to investigate a limited part of an enterprise’s business rules. The business rules we have chosen to study are targeting recommendation support for value propositions, based on different business models. The recommendations are based on following five parts of the Osterwalder canvas [74]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions. In order to validate this we stated the following research question:

*RQ: For a limited part of an enterprise’s business rules, can a machine learning pipe-line support visual and logical verification of business rules, and generate executable business rules?*

We stated the following null-hypothesis: It is not possible to use a machine learning pipe-line to support visual and logical verification of business rules. Neither is it possible to use the pipe-line to generate executable business rules.

5.1.3 Introduction Summary

The remainder of this article is organized as follows. The methodology is presented in Section 5.2 and the execution results and an analysis thereof are presented in Section 5.3 and Section 5.4. The results are further discussed in Section 5.5 and conclusions and future work are presented in Section 5.6.
5.2 Methodology

We conducted an experiment to evaluate if it is possible to use a machine learning pipeline to support visual and logical verification of business rules, and generate executable business rules.

The nature of the business rule data can range from, for example integers, product identifiers, date and time intervals, etc. This requires handling of numerical-, categorical-, and nominal-data. In the context we are investigating, the number of features is estimated to be between three and six (in normal cases) and have an upper limit of ten features. The majority of the features will be categorical or nominal. Each categorical- or nominal-feature is estimated to have between two and ten unique values. Since humans are writing the business rules the number of rows are estimated to be less than 30. For each row there shall be two different classification columns. These classification columns respectively represent eligible objects and the allowed actions on these objects. The idea is borrowed from the gaming industry where a specific context gives the character the possibilities to, for example find specific treasures and stipulates how these treasures can be handled.

There are no real-time requirements on the generation of the executable business rules. The solution should be implemented in a language which is used by Ericsson. We decided to use Python since it has many machine learning contributions, for example Scikit [97]. The implementation is done with the help of Jupyter notebook, from the Anaconda distribution for Mac (x86 64-bits), which supports Python 3.5.2.

Examples of business rules which could be valid for recommending value propositions, based on different business models, were provided to us. The examples were delivered as csv-files. These examples were used to verify the solution. The verification of the solution was done together with practitioners from Ericsson.

To avoid the risk of imposing solutions from the authors, member checking [89] was used in the form of presentations to the involved practitioners. The negative case analysis [89] was used to challenge the solution and counter researcher bias. The solutions were presented to and feedback given by, the involved practitioners at different stages of the study. The selection of test data was performed by the involved practitioners.
Chapter 5. Towards Executable Business Rules

5.3 Results

We have implemented a machine learning pipe-line which makes it possible to conduct visual and logical verification of business rules, and generate executable business rules. We have added different types of functionality which is regarded as needed when handling business rules. Missing data is handled as a wildcard, i.e. all values are true. Continuous values have a defined boundary and there are no value gaps in the data. Since a human is defining the business rules, entering all possible combinations by hand is not an option. A meta-data file is used to describe the nature of the features. The ability to generate executable code, which can be distributed in a system, is another requirement. The pipe-line consists of three parts. A pre-processing part, the use of a decision tree algorithm called ID3, and a post-processing part.

The first step of the pre-processing is responsible for reading in the data and the meta-data. The number of feature columns is not restricted by the software. The next step is to transform missing data into a wildcard character. The result is shown in a tabular format. The third step uses the meta-data to determine the feature columns with continuous characteristics. The values in these feature columns are converted into a format which mimic continuous data. The fourth step splits the data set into two sets, each responsible for one of the classification columns. These classification columns respectively represents eligible objects and the allowed actions on these objects. A decision tree for each of these data sets are created with the help of the ID3 algorithm.

The ID3 algorithm is implemented without any deviations from its specification [84]. In the implementation we use Shannon entropy as the impurity measure [35]. However, we have added a warning message if majority voting is used since this might indicate ambiguity in the business rule set. The result is a Python dictionary which represents the resulting model of the business rules.

The first step of the post-processing is responsible for creating a dot-notation based on the model. Graphviz is used to present the model as a tree graph. The tree graph makes it easier for the responsible actors to understand, and analyze the results of the defined business rules. A presentation of the business rules in a tabular format is done as well. The second step uses the model to create executable Python code which represents the model of the business rules. This code is displayed to the user. The last step gives the possibility to make predictions against the model. The predictions are made against the executable Python code. The executable model of the business rules is persisted as a Python module. This makes it possible to distribute the executable model to different components in a business support system.
Chapter 5. Towards Executable Business Rules

The aim is to use the executable business rules to guide or create business processes which are responsible to adhere to these rules. Since these business rules can be distributed, the use of a monolithic rule engine can be avoided. The business rules can be defined in a tool which can export its content as a csv-file, for example Excel.

5.4 Analysis

The pipe-line is considered as a proof of concept, and as such is regarded as successful by the involved practitioners. The pipe-line makes it possible to, visually and logically, validate the correctness of the business rules before they are put in production. The possibility to generate executable code representing the model of the business rules makes it possible to execute the same model in different components without the need of re-implementation. This might improve the coherence of the business rules in a business support system.

From a machine learning perspective, we have added the possibility for the ID3 algorithm to use continuous features.

Based on this analysis we can reject the null-hypothesis.

There are several improvements to the pipe-line which should be considered. The precision of the limits have to be based on the precision of the affected feature value. The precision has to be configurable feature by feature, and with different values for the maximum limit and minimum limit. There is a request for a visualization of the difference between the features used in the business rules and those used in the model. This would give the responsible actor a better indication of features which have no impact on the model. The stability of the dot-notation generation has to be improved. The executable code representing the model of the business rules should support additional languages, for example Java™.

When we shall cover more of the business rules aimed for recommendations, it must be possible to distinguish between numerical values which are pure positive or pure negative. The possibility to use logical expressions as input will be required. The handling of date- and time-intervals, and durations are regarded as needed.
5.5 Discussion

The decision trees’ over-fitting problem is turned into an advantage when we want to evaluate or execute defined business rules.

An advantage of the ID3 algorithm is the use of multi-split instead of the binary split. The multi-split gives a better overview of the model, from a human perspective. Another advantage with the multi-split is the rather trivial procedure to produce executable code based on the model. Jearanaitanakij [49] has extended the ID3 algorithm to handle continuous data. We have chosen to use the standard ID3 algorithm because when date- and time-intervals, and durations are introduced as feature characteristics, we can extend the ID3 algorithm with the same type of mechanisms as we use for continuous data.

Business rules requires wild cards, explicit boundaries, and features with no gaps between the values. Scikit provides an implementation of decision trees based on the CART [14] algorithm. We have tried to use this implementation but it is not suitable for the mixture of the data and the additional requirements we are putting on the data. To our understanding, the C4.5 algorithm [111] and the extended version of Hoeffding trees [31] shares the same limitations as the CART algorithm, regarding our requirements on the data. Instead of changing an existing algorithm, we decided to use a base implementation of a simple algorithm. In this case we do not need to verify the algorithm, but rather concentrate on the desired functionality of the pipe-line.

Is a machine learning pipe-line for business rules a correct approach, and is it worth the effort? The response from the practitioners indicates a usefulness of machine learning algorithms in the business rule domain. The decision tree algorithm cannot be used for all types of business rules. Instead we have to investigate which machine learning algorithms can be useful for each specific business rule type. We believe that most of the pre- and post-processing functionality is useful if we decide to implement other types of business rules with the help of rule lists, and expose the result as rule trees [35].

The estimated size of the data set is small which will not require special memory handling. There are no real-time requirements on the transformation from business rules to executable business rules. Since the data set will vary in the number of used feature columns and since each feature has its own characteristic, the use of a typed-language is not ideal. Based on this, Python fulfills the requirements as a suitable implementation language for the problem at hand.

In order to for the solution to exist in an event-driven environment, the metadata has to be extended with the events a business rule is intended for. Contrary
to many proposed solutions, we believe that a business rule could be triggered by several different events. This makes the use of a simple event-condition-action architecture not suitable for the problem at hand.

The literature do not provide many papers in the area we are investigating. The main contributors in the area are Rosca et al. Their contributions are on a less technical level than this study but the ideas for a governance frame work of business rules are useful. This might indicate a need for further studies in the area or an inability, from our side, to find the right papers.

5.6 Conclusion and Future Work

The experiment shows that it is possible to support visual and logical verification of business rules, and to generate executable business rules with the help of a machine learning pipe-line. We have extended the ID3 algorithm to be used with continuous values.

The suggested improvements, described in Section 5.4, will be implemented in order to validate the pipe-line for a real business support system environment. How to use the work of Rosca et al. in the area of common governance of business rules will be further investigated.

During the experiment, the involved practitioners have shown a great interest in using machine learning as a part of their business support system offering.
Chapter 6

Uncovering Implicit Rules in Medicine Diagnosis
Abstract

Context: Decisions taken by experts may be based on explicit and implicit rules. By uncovering the implicit rules the expert may have the possibility to explain its decisions in a better way, both for itself and the person which the decision is affecting. In the area of medicine, laws are enforcing the expert to be able to explain its decision when a patient is complaining about a decision. Another vital aspect is the ability of the expert to explain to the patient why a certain decision is taken, and the risks associated with the decision.

Objective: To investigate if it is possible for a machine learning pipe-line to find implicit rules used by experts, when they decide if a patient could be operated or not.

Method: We conduct an analysis of a data set, containing information about patients and the decision if an operation should be performed or not.

Results: We have implemented a machine learning pipe-line which supports detection of implicit rules in a data set. The detection of the implicit rules are supported by an algorithm which implements an agglomerative merging of feature values. We have improved the original algorithm by showing the borders of the feature values of a discretization bin.

Conclusion: The analysis of the data set shows it is possible to find implicit rules used by the experts with the help of an agglomerative merging of feature values.

6.1 Introduction

Decisions taken by experts may be based on explicit and implicit rules. By uncovering the implicit rules the expert may have the possibility to explain its decisions in a better way, both for itself and the person which the decision is affecting.

In the area of medicine, laws are enforcing the expert to be able to explain its decisions when a patient has complaints. Another vital aspect is the ability of the expert to explain to the patient why a certain decision is taken, and the risks associated with the decision. We are investigating a data set, containing patient features and decisions about if an operation should be performed or not.

We like to uncover implicit rules used by a human expert, and use the findings to enlighten the expert about how the decisions are taken. The algorithms we use
should give an explanation to the decision in a human understandable format. Tree algorithms and rule algorithms produce human understandable result. Other types of algorithms could be used to select the best features to explain the data set. The third type of algorithms we find useful are feature discretization algorithms. The last two types of algorithms could have supervised or un-supervised methods.

Our ideas of finding human defined rules are rooted in the area of Fuzzy sets [113]. Ross [93] describes different methods for finding the membership function by using different machine learning algorithms. Rakus-Andersson [85] has used fuzzy sets to improve decision making in medical diagnosis. The difference of finding the membership function in fuzzy set and what we want to achieve is the knowledge of the feature. In fuzzy sets, the different classifications of a feature is stated by experts and the membership function can be calculated based on this knowledge. We are investigating how the different classifications of a feature can be uncovered.

We state the following research question:

*RQ: Can a machine learning pipe-line detect implicit rules in a data set were the implicit rules are useful as a human judgement, and by using the rules found in the data set compared to the rules stated by the expert get a significant improvement when using a machine learning algorithm?*

We state the following null-hypothesis: It is not possible to use a machine learning pipe-line to detect implicit rules in a data set were the implicit rules are useful as a human judgement, and by using the rules found in the data set compared to the rules stated by the expert get a significant improvement when using a machine learning algorithm.

The remainder of this article is organized as follows. The methodology is presented in Section 6.2 and the execution results and an analysis thereof are presented in Section 6.3 and Section 6.4. The results are further discussed in Section 6.5 and conclusions and future work are presented in Section 6.6.

### 6.2 Methodology

We conduct an analysis of a data set containing information about patients and the decision if an operation should be performed or not. We try to evaluate if it is possible to find implicit rules, used by a doctor when deciding if a patient can have an operation or not.
Examples of real-world data from a hospital, together with the rules used for judgement, were provided to us. The examples were delivered as csv-files.

There are no real-time requirements on the generation of the implicit rules. We have decided to use Python since it have many machine learning contributions, for example Scikit [97]. The implementation is done with the help of Jupyter notebook, from the Anaconda distribution for Mac (x86 64-bits), which supports Python 3.5.2.

We have implemented the agglomerative merging algorithm described by Flach [35], as well as Wilcoxon’s signed-rank test. The other algorithms we are using are implemented by Python libraries.

### 6.3 Results

We have implemented a machine learning pipe-line which makes it possible to find implicit rules, used by a doctor when deciding if a patient can have an operation or not. Since a human is defining the rules, the implicit rules must make sense to a human being. The pipe-line consists of three parts. A pre-processing part, the use of a decision tree algorithm called CART [14], and a post-processing part.

In the first step of the pre-processing we analyze the data set. In Table 6.1 we can see that there are two features with missing data and less than 30% of the class variables values are indicating a positive result, i.e. No operation should be performed.

<table>
<thead>
<tr>
<th>Year</th>
<th>CRP</th>
<th>Age</th>
<th>Sex</th>
<th>Weight</th>
<th>No Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>count</td>
<td>55</td>
<td>55</td>
<td>55</td>
<td>43</td>
<td>55</td>
</tr>
<tr>
<td>mean</td>
<td>2001.15</td>
<td>28.88</td>
<td>71.22</td>
<td>0.44</td>
<td>68.25</td>
</tr>
<tr>
<td>min</td>
<td>2000</td>
<td>4.0</td>
<td>29.0</td>
<td>0</td>
<td>42.2</td>
</tr>
<tr>
<td>max</td>
<td>2002</td>
<td>137.0</td>
<td>94.0</td>
<td>1</td>
<td>96.0</td>
</tr>
</tbody>
</table>

Table 6.1: Statistics for the data set.

In the second step of the pre-processing we are investigating the correlation between the features, and between the features and the class variable. In Table 6.2 we can see that the correlation between the features and the class variable are not strong, less than 0.5. The strongest correlation between the features are between the weight and the sex (-0.40). The correlation between the other features are not strong.

The Shapiro-Wilk test on the CRP feature and the Age feature indicates that a non-parametric correlation test is needed (Table 6.3). We use Kendall’s tau
since it is robust for small data sets. The Kendall’s tau test show no evidence of correlation between the investigated features and the class variable. There is no evidence of strong correlation between the Weight and the Sex features when we perform a Kendall’s tau test.

### Table 6.3: Shapiro-Wilk and Kendall measures.

<table>
<thead>
<tr>
<th></th>
<th>Shapiro</th>
<th>Kendall’s tau</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>0.76, 3.27e-08</td>
<td>0.38, 3.67e-05</td>
</tr>
<tr>
<td>CRP</td>
<td>0.76, 1.30e-07</td>
<td>0.36, 0.0002</td>
</tr>
<tr>
<td>Age</td>
<td>0.92, 0.001</td>
<td>0.29, 0.001</td>
</tr>
</tbody>
</table>

During the last step of the pre-processing we perform an agglomerative merging of the feature values, using Chi square. The result is shown in Table 6.4. Table 6.4 shows the number of negative class variable values, the number of positive class variable values, and the value range for the feature variable. We have improved the original algorithm by showing the boarders of the feature values of a discretization bin.

### Table 6.4: Agglomerative merging of feature values with Chi 2.

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>range</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRP</td>
<td>16.0</td>
<td>2.0</td>
<td>[4.0, 7.0]</td>
</tr>
<tr>
<td></td>
<td>19.0</td>
<td>20.0</td>
<td>[8.0, 72.0]</td>
</tr>
<tr>
<td></td>
<td>0.0</td>
<td>8.0</td>
<td>[86.0, 137.0]</td>
</tr>
<tr>
<td>Age</td>
<td>5</td>
<td>0</td>
<td>[29, 50]</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>[51, 51]</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0</td>
<td>[54, 62]</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>[63, 66]</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>0</td>
<td>[67, 73]</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>16</td>
<td>[74, 87]</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>10</td>
<td>[89, 94]</td>
</tr>
<tr>
<td>Weight</td>
<td>0</td>
<td>6</td>
<td>[42.2, 47.0]</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>4</td>
<td>[49.0, 52.8]</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>0</td>
<td>[55.0, 72.0]</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>2</td>
<td>[72.5, 72.5]</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0</td>
<td>[74.0, 85.0]</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>6</td>
<td>[85.7, 88.0]</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0</td>
<td>[95.0, 96.0]</td>
</tr>
</tbody>
</table>

During the processing step we use four different data sets. The first data set is the original data set after adding -1 for missing data and duplication of all the
rows with a positive class variable value. This data set is used as a base for the other three data sets. The second data set is binned according the rules which were given by the experts. The third data set is binned according to the results in Table 6.4. The fourth data set uses the experts rules and introduce additional binning information from Table 6.4. The new values which are used for binning are natural to humans when they do an estimation, e.g. a “young” person has the age of 50 or less, instead of 62 or less.

When perform a stratified 10-fold cross validation with the CART algorithm. The result from the precision measures can be seen in Table 6.5. In Table 6.5 we can see that the precision and the standard deviation have improved when we use the fourth data set compared to the second data set.

<table>
<thead>
<tr>
<th>Fold</th>
<th>No bins</th>
<th>Expert</th>
<th>Binning</th>
<th>Adapted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.833333</td>
<td>0.750000</td>
<td>0.900000</td>
<td>0.900000</td>
</tr>
<tr>
<td>2</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.900000</td>
</tr>
<tr>
<td>3</td>
<td>0.708333</td>
<td>0.875000</td>
<td>0.875000</td>
<td>0.875000</td>
</tr>
<tr>
<td>4</td>
<td>0.583333</td>
<td>0.583333</td>
<td>0.583333</td>
<td>0.800000</td>
</tr>
<tr>
<td>5</td>
<td>1.000000</td>
<td>0.900000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>6</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.875000</td>
</tr>
<tr>
<td>7</td>
<td>0.800000</td>
<td>0.800000</td>
<td>0.800000</td>
<td>0.583333</td>
</tr>
<tr>
<td>8</td>
<td>0.900000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>9</td>
<td>0.708333</td>
<td>0.875000</td>
<td>0.875000</td>
<td>1.000000</td>
</tr>
<tr>
<td>10</td>
<td>1.000000</td>
<td>0.666667</td>
<td>0.875000</td>
<td>0.875000</td>
</tr>
<tr>
<td>avg</td>
<td>0.853333</td>
<td>0.845000</td>
<td>0.890833</td>
<td>0.880833</td>
</tr>
<tr>
<td>stdev</td>
<td>0.151444</td>
<td>0.144615</td>
<td>0.129663</td>
<td>0.124043</td>
</tr>
</tbody>
</table>

Table 6.5: Precision measures.

We perform a F1-measure in order to emphasis the search for true-positives instead of true-negatives. The result from the precision measures can be seen in Table 6.6. In Table 6.6 we can see that the precision and the standard deviation have improved when we use the fourth data set compared to the second data set.

<table>
<thead>
<tr>
<th>Fold</th>
<th>No bins</th>
<th>Expert</th>
<th>Binning</th>
<th>Adapted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.733333</td>
<td>0.750000</td>
<td>0.873016</td>
<td>0.873016</td>
</tr>
<tr>
<td>2</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.873016</td>
</tr>
<tr>
<td>3</td>
<td>0.708333</td>
<td>0.857143</td>
<td>0.857143</td>
<td>0.857143</td>
</tr>
<tr>
<td>4</td>
<td>0.571429</td>
<td>0.571429</td>
<td>0.571429</td>
<td>0.708333</td>
</tr>
<tr>
<td>5</td>
<td>1.000000</td>
<td>0.844444</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>6</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
<td>0.857143</td>
</tr>
<tr>
<td>7</td>
<td>0.708333</td>
<td>0.708333</td>
<td>0.708333</td>
<td>0.571429</td>
</tr>
<tr>
<td>8</td>
<td>0.844444</td>
<td>1.000000</td>
<td>1.000000</td>
<td>1.000000</td>
</tr>
<tr>
<td>9</td>
<td>0.708333</td>
<td>0.857143</td>
<td>0.857143</td>
<td>1.000000</td>
</tr>
<tr>
<td>10</td>
<td>1.000000</td>
<td>0.666667</td>
<td>0.828571</td>
<td>0.828571</td>
</tr>
<tr>
<td>avg</td>
<td>0.827421</td>
<td>0.825516</td>
<td>0.869563</td>
<td>0.856865</td>
</tr>
<tr>
<td>stdev</td>
<td>0.162054</td>
<td>0.150156</td>
<td>0.143347</td>
<td>0.135942</td>
</tr>
</tbody>
</table>

Table 6.6: F1-measures.

In the first step of the post-processing we are comparing all the precision measures

88
with the help of a Friedman test and, if needed, a Nemenyi test. In the second step, we do the same for all the F1-measures.

In the third step of the post-processing we are comparing the precision measures for data set two and data set four with the help of a Wilcoxon's signed-rank test. In the fourth step, we do the same for the F1-measures for data set two and data set four.

6.4 Analysis

We decide to classify the missing data with the value -1 and let the tree algorithm handle this as a decision point. Since the data is skewed towards a representation of a negative classification, we decide to duplicate the rows which contains a positive classification. The duplication of the rows gives a data set which contains 45% of positive classifications. The weak correlation between the features, and between the features and the class variable, indicates that we will not introduce any correlation errors when we duplicate the rows with a positive class variable value.

With the help of agglomerative merging of the feature values we can find implicit rules used by the experts. The bold text in Table 6.7 shows the main differences between the original rules and the extensions we introduce with the help of agglomerative merging of the feature values.

In Table 6.7 we can see a new CRP level, named “very high”. From our perspective it seems intuitive with a new upper level since the values can be above 200.

The Age feature has a new level called “young”, and the upper boarder for “old” has been raised by nine years. The additional level “young” seems to be natural limit. However, the raise of “old” from 80 to 89 cannot be regarded as intuitive or natural.

Since there was no distinction between the males and the females in the rules used by the experts, the changes of the boarders of the Weight feature seems natural. The strongest correlation between the features are between the sex and the weight. We have introduced a normal weight for females and a normal weight for males. The normal weight for females are regarded as under weight for males. On the other hand, the normal weight for males is regarded as over weight for females.

The results from the measurements (Table 6.5 and Table 6.6) indicates that data set four performs better than data set two. However, the post-processing steps
shows no evidence of any difference between the different data sets regarding the performance measures or the F1-measure. Based on this, we can not reject the null-hypothesis.

### 6.5 Discussion

We considered three different tree algorithms to uncover implicit rules, ID3 [84], CART [14], and C4.5 algorithm [111]. Since ID3 do not use binary splits we find this algorithm most suitable for our purpose. However, it is still hard to find the human usable rules. We have chosen to use the CART algorithm to evaluate if the addition of the found implicit rules will improve the model. We use CART since it has the capability of handling continuous data, which ID3 is lacking.

We use an implementation of the CN2 rule induction algorithm [22], but the result has the same draw backs as the tree algorithms in the form of detecting human usable rules.

The use of feature selection algorithms do not seem to be useful for the data set at hand. The number of features are very small and the correlation shows no strong indication of a specific feature which dominates the classification (see Table 6.2).

We use the agglomerative discretization instead of the divisive discretization [35] since we get the minimized discretization as an initial step. The initial step could be used to find the base for fuzzy set member functions, and is a way to use fuzzy classification. We have improved the original algorithm by showing the boarders of the feature values of a discretization bin. This improvement makes it possible to detect ranges of feature values which are not used in the classification.
Chapter 6. Uncovering Implicit Rules in Medicine Diagnosis

We need to do some visual improvements to make it easier to capture the implicit rules. We would further like to investigate if there is a possibility to use a mathematical approach in order to find fuzzy sets which can be used for fuzzy classification.

6.6 Conclusion and Future Work

The analysis shows that it is possible to find implicit rules used by experts with the help of an agglomerative merging of feature values.

We believe that the extensions we have done to the agglomerative discretization algorithm are a valuable contribution for finding implicit rules in a data set. The suggested improvements, described in Section 6.5, are needed in order make it easier to capture the implicit rules used by the experts.
A.1 Investigating the Need for a Systematic Literature Review

We constructed a search string in order to determine if similar work had already been performed. The words in the search string were obtained from the research questions and through brainstorming with domain experts from academia and the industry. We grouped the words according to Table A.1.1.

As a first step we performed a search in the Scopus library within title, keyword and abstract. The search API from Scopus includes the stemming functionality. We used a search string composed of the groups in Table A.1.1 with AND between them. The search showed that no single paper covered a intent-driven systems.

The second step was to find out if any literature reviews have been done in this area. After a discussion with domain experts from Ericsson we formed a search string as K AND L AND M AND N AND V AND (O OR P OR Q OR R OR S OR T OR U) AND “review”. Once again we performed a search in the Scopus library within title, keyword and abstract. The search found no single paper with a literature review on intent-driven systems.

The last step was to find out if any literature reviews have been done in this area by only keeping some of the groups. We formed a new search string as K AND L AND M AND N AND V AND “review”. Once again we performed a search in the Scopus library within title, keyword and abstract. The search showed one paper [54]. This paper is about eye movement systems in the area of neurophysiology, and thus we conclude that no literature review has been conducted on intent-driven systems.

<table>
<thead>
<tr>
<th>Group</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>(&quot;intent&quot; OR &quot;intention&quot; OR &quot;vision&quot; OR &quot;strategy&quot; OR &quot;goal&quot; OR &quot;tactic&quot; OR &quot;objective&quot;)</td>
</tr>
<tr>
<td>L</td>
<td>(&quot;driven&quot; OR &quot;motivated&quot; OR &quot;focused&quot;)</td>
</tr>
<tr>
<td>M</td>
<td>(&quot;system&quot; OR &quot;solution&quot; OR &quot;realization&quot;)</td>
</tr>
<tr>
<td>N</td>
<td>(&quot;govern&quot; OR &quot;guide&quot; OR &quot;control&quot;)</td>
</tr>
<tr>
<td>O</td>
<td>(&quot;actor&quot; OR &quot;stakeholder&quot; OR &quot;agent&quot;)</td>
</tr>
<tr>
<td>P</td>
<td>(&quot;express&quot; OR &quot;expose&quot; OR &quot;formulate&quot;)</td>
</tr>
<tr>
<td>Q</td>
<td>(&quot;view-point&quot; OR &quot;context&quot; OR &quot;value&quot; OR &quot;characteristic&quot; OR &quot;capability&quot;)</td>
</tr>
<tr>
<td>R</td>
<td>(&quot;component&quot; OR &quot;part&quot; OR &quot;building block&quot; OR &quot;role&quot;)</td>
</tr>
<tr>
<td>S</td>
<td>(&quot;translate&quot; OR &quot;transform&quot; OR &quot;bridge&quot; OR &quot;map&quot; OR &quot;match&quot;)</td>
</tr>
<tr>
<td>T</td>
<td>(&quot;simulate&quot; OR &quot;visualize&quot; OR &quot;validate&quot;)</td>
</tr>
<tr>
<td>U</td>
<td>(&quot;effectiveness&quot; OR &quot;efficiency&quot;)</td>
</tr>
<tr>
<td>V</td>
<td>(&quot;decision support&quot; AND &quot;DSS&quot;) OR &quot;behavior driven&quot; OR &quot;feedback observation&quot;)</td>
</tr>
</tbody>
</table>

Table A.1.1: The initial keywords


Appendix A. Appendix to Chapter 2

A.2 Pilot Study

The systematic review procedure was piloted in order to establish a common view of the selection criteria.

We decided to use IC1-IC3 in order to find relevant papers from an academic point of view. Since intent-driven systems include more subject areas than computer science we decided to use IC4 to broaden the search to include relevant subject areas. We based IC6 on RQ1.1-RQ1.6 which are derived from our understanding of intent-driven systems. IC6 is the main criterion for including a paper as a primary study.

The first selection was done against SCOPUS (finished 2014-12-03) and 1227 papers where found. After applying IC1-IC3, 1052 papers were retrieved. IC4 was applied on the retrieved set which left 795 papers for primary study selection. The number of papers that were regarded as primary studies for full-text reading were 106. We used a search string which is composed of terms grouped together as K, L, M, N and O and executed as K AND L AND M AND N AND O. The groups and their terms are found in Table A.2.1.

<table>
<thead>
<tr>
<th>Group</th>
<th>Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>(&quot;intent&quot; OR &quot;intention&quot; OR &quot;vision&quot; OR &quot;strategy&quot; OR &quot;strategies&quot; OR &quot;goal&quot; OR &quot;tactic&quot; OR &quot;objective&quot;)</td>
</tr>
<tr>
<td>L</td>
<td>(&quot;driven&quot; OR &quot;motivated&quot; OR &quot;focused&quot;)</td>
</tr>
<tr>
<td>M</td>
<td>(&quot;system&quot; OR &quot;solution&quot; OR &quot;realization&quot; OR &quot;realisation&quot;)</td>
</tr>
<tr>
<td>N</td>
<td>(&quot;govern&quot; OR &quot;guide&quot; OR &quot;control&quot;)</td>
</tr>
<tr>
<td>O</td>
<td>(&quot;decision support&quot; AND &quot;DSS&quot;) OR &quot;behavior driven&quot; OR &quot;feedback observation&quot;)</td>
</tr>
</tbody>
</table>

Table A.2.1: The initial keywords for the pilot study

In order to find new keywords papers regarded as primary studies for full-text reading were selected. The Authors’ keywords were extracted from the selected papers. We selected the keywords to be part of the search string based on our understanding of intent-driven systems. The new keywords helped to form the final search strings (Table 2.1)

We piloted the data extraction (Appendix A.3) by selecting the first two papers in each subject area, if applicable, from the set of relevant papers found in the SCOPUS search. Thoughts from the Grounded Theory were applied on the selected set of papers to find better data extraction questions. Based on the outcome we changed some of the data extraction questions in terms of wording, added new questions and re-ordered the questions. This was done in order to get a more accurate and smoother way to find answers to the research questions.

During the pilot it was decided to add exclusion criterion 1.
A.3 Data Extraction Protocol

This appendix contains information about the data extraction.

An extraction form with the name of the properties and the purposes of the properties was created (Table A.3.1).

<table>
<thead>
<tr>
<th>ID</th>
<th>Property</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>Industry Involvement.</td>
<td>Overview of the studies.</td>
</tr>
<tr>
<td>P2</td>
<td>Additional information.</td>
<td>Overview of the studies.</td>
</tr>
</tbody>
</table>

Table A.3.1: Properties

A.3.1 Data extraction protocol for RQ1:1-RQ1:6

The data extraction protocol for RQ1:1-RQ1:6 are described in Table A.3.2 and Table A.3.3.

Reason: The questions are selected to be able to see the maturity of the method used and possible proprietary solutions. This is of great interest for the company involved in the research project.

Definitions: No Stated - if the study does not define the applied method and it cannot be derived or interpreted from reading the paper.

A.3.2 Data extraction for P1

The property 1 is described in Table A.3.4.

Reason: It is vital to see the involvement from industry since this can be used to understand where any competitors are making research.

Definitions: No Stated - if the study does not state involvement from industry and it cannot be derived or interpreted from reading the paper.

A.3.3 Data extraction for P2

Information deemed vital for the study but not captured by the other properties.
## Appendix A. Appendix to Chapter 2

<table>
<thead>
<tr>
<th>Code</th>
<th>ID</th>
<th>Question</th>
<th>RQ1:1</th>
<th>RQ1:2</th>
<th>RQ1:3</th>
<th>RQ1:4</th>
<th>RQ1:5</th>
<th>RQ1:6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Relevance(1)</strong></td>
<td>1</td>
<td><strong>What is the relevance of the contribution to IDS?</strong> Select one: 1. Describes a (validating) method in the intended context of IDS. 2. In-between 1 and 3. 3. Describes a (validating) method that could be used in IDS.</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Contribution(2)</strong></td>
<td>2</td>
<td><strong>How could the contribution be used in IDS?</strong> Select one: 1. As a contribution to a framework 2. As a contribution to a theory 3. As a contribution to background material The area of the contribution.</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Standards and Theories(3)</strong></td>
<td>3</td>
<td><strong>Is the contribution to the (validating) method based on a standard/academic theory?</strong> Yes (which), No, Not stated</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td><strong>How is the standard/academic theory presented?</strong> Used, defined</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Limitations(4)</strong></td>
<td>5</td>
<td><strong>Are there any limitations in the suggested contribution to the (validating) method?</strong> Size, Distribution, Concurrency, Sharing, Others, Not stated</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td><strong>Is the contribution to the (validating) method targeting a specific domain?</strong> Yes (which), No, Not stated</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td><strong>Role support(5)</strong></td>
<td>7</td>
<td><strong>What types of actors are supported?</strong> Parties (organizations, individuals), Services, Resources, Not stated</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td><strong>How are the actors represented?</strong> RRL, by software (agents, simulation, etc.), by data, Not stated</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

Table A.3.2: Data Extraction for RQ1:1-RQ1:6 part 1
### Appendix A. Appendix to Chapter 2

<table>
<thead>
<tr>
<th>Code</th>
<th>ID</th>
<th>Question</th>
<th>RQ1:1</th>
<th>RQ1:2</th>
<th>RQ1:3</th>
<th>RQ1:4</th>
<th>RQ1:5</th>
<th>RQ1:6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>How are the definitions of contexts supported?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>“Blank sheet”, specific source (which), generic sources, Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Partitions supported, overlapping partitions supported, Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>How is the governance of context definitions supported?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Fixed after definition, Merge of contexts (full or partial), Split of contexts, No support, Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>How are the combinations of actors, views and contexts supported?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>One “view” of one “context” (one of several actors), Several views of one context (one or several actors), Several contexts (one or several actors), Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>How are the validation of combinations of actors, views and contexts supported?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>One “view” of one “context” (one of several actors), Several views of one context (one or several actors), Several contexts (one or several actors), Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tools and Technology(7)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>What technologies are used for constructing the contribution to the (validating) method?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Known technologies (which), New technologies (used by others), Not stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>Are any specific tooling required for the construction of the contribution to the (validating) method?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Yes (Which tools), No, Not Stated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table A.3.3: Data Extraction for RQ1:1-RQ1:6 part 2

<table>
<thead>
<tr>
<th>Code</th>
<th>ID</th>
<th>Question</th>
<th>P1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>Is the industry involved according to the paper?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Yes (Which companies), No, Not Stated</td>
<td></td>
</tr>
</tbody>
</table>

Table A.3.4: Data Extraction for P1

97
A.4 Comments on Primary Studies

This appendix contains individual comments on the selected primary studies. Table A.4.1 contains individual comments on papers passing criterion level 1 (Section 2.2.6). Individual comments on the other papers are found in Table A.4.2.

Table A.4.1 lists papers passing criterion level 1, the research questions a paper supports and comments on the paper. If a paper does not support a research question on criterion level 1 this research question is put in brackets.

<table>
<thead>
<tr>
<th>Paper</th>
<th>RQ</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>1.1</td>
<td>Theory combined with experiment: A new fuzzy protocol for negotiations used by agents in a grid environment. The aim with the protocol is to achieve a flexibility to relax decision to complete a deal when faced with trading pressure. The main features of the protocol are: “A more accurate consideration of the trading pressure.”, “The possibility to handle non-reasonable behavior of negotiator agents.” and “The ability to handle multiple trading opportunities and market competition.”. The main features of the presented fuzzy protocol could be used to negotiate business intents between the involved actors. This supports RQ1.1.</td>
</tr>
<tr>
<td>[7]</td>
<td>1.6, (1.3, 1.5)</td>
<td>Theory combined with experiment: Methods to transform goals into policy configurations. These transformation methods are not discipline specific. This will make it possible to use the same method independently of the policy discipline as well as supporting a policy configuration with heterogeneous policy disciplines. A case based transformation approach was evaluated and two other approaches were discussed (static rules and policy table lookup). In order to be able to make use of the multi dimensional information the number of dimensions needed to be considered has to be reduced. Two dimension reduction methods where presented, principal component analysis (PCA) and feature selection. The use of PCA to reduce and find suitable mappings between actor goals and component behavior might be useful. This supports RQ1.6. Business level abstractions are done in order to map business goals to policies. As presented in the paper, the abstraction level seems to be on a too low level to qualify as business goals. The use of a case database as a knowledge base for the transformation model focus on policies according to IETF. This makes the paper not passing criterion level 1 as support for RQ1.3 and RQ1.5.</td>
</tr>
<tr>
<td>[17]</td>
<td>1.1</td>
<td>Theory combined with results from a real world case: Describes how a Decision Support System (DSS) for process improvement could be realized with the help of a multi agent system. The multi agent system combined with a knowledge representation model is used for simulations. This knowledge representation model is shown and discussed. The use of a multi agent system in order to do simulation in a DSS supports RQ1.1.</td>
</tr>
<tr>
<td>[19]</td>
<td>1.1, 1.2</td>
<td>Theory combined with experiment: Presents a solution to increase a decision support system’s efficiency when dealing with a great amount of alternatives and criteria. This is done by using Multi Criteria Decision Analysis combined with a functionality to explain the decisions taken. To acquire the information from a human actor fuzzy logic has been used. This makes it easier for the actor to express the information in a natural way. The Multi Criteria Decision Analysis is broken down into a structure close to the Context Frame concept. How ever, the solution supports only one actor. This supports RQ1.1. The functionality to explain the decisions taken supports RQ1.2.</td>
</tr>
</tbody>
</table>
### Appendix A. Appendix to Chapter 2

<table>
<thead>
<tr>
<th>Reference</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[24]</td>
<td>1.1</td>
<td>Theory combined with experiment: Presents a framework to support decision makers in reasoning under uncertainty. The paper describes how to use Decision Maps to assist in Multi-Criteria Decision Analysis. The framework’s strength is its possibility to handle heterogeneous information. The possibility to handle heterogeneous information supports RQ1.1.</td>
</tr>
<tr>
<td>[37]</td>
<td>1.1</td>
<td>Theory combined with experiment: Describes a Decision Support System (DSS) where one group of actors is using Fuzzy Multi Criteria Decision Analyze and the other actors are modeled as agents (based on survey data) or a simulation (environment model). The DSS will support a decision making process in a complex choice situation with multiple management objectives and multiple strategies that are under consideration. The possibility to support a decision making process in a complex choice situation with multiple management objectives and multiple strategies that are under consideration supports RQ1.1.</td>
</tr>
<tr>
<td>[43]</td>
<td>1.1</td>
<td>Theory combined with experiment: Presents a framework for distributed planning in hierarchical structures with the help of a Multi Agent System (MAS) and Multi Decision Criteria Analyzes based on a model approach. Requirements for a decision support system based on this concept are proposed. Distributed planning in hierarchical structures supports RQ1.1. According to the paper MAS approaches are appropriate for the enterprise management but there is a lack of appropriate semantics and ontologies.</td>
</tr>
<tr>
<td>[57]</td>
<td>1.1</td>
<td>Theory combined with experiment: Presents a methodology for structured argumentation. The methodology is stated to be non-intrusive on an enterprise’s existing analytic products and methods. The structured arguments methodology consists of an argument template used to govern the analytic process structure and arguments which is the instantiation of an argument template. Defining templates for a structure. The structure itself could be used to define different context frame. This supports RQ1.1.</td>
</tr>
<tr>
<td>[59]</td>
<td>1.4, (1.3)</td>
<td>Theory combined with experiment: Presents a way to improve the decision strategies for agents with the use of genetic algorithms. The method used to build the models are recursive in its nature. Measure the fitness of an Agent in a Multi Agent System in order to improve the performance of new agents. This is done with the help of evolutionary mechanisms. This supports RQ1.4 but the paper do not pass criterion level 1 as support for RQ1.3.</td>
</tr>
<tr>
<td>[60]</td>
<td>1.6, (1.5)</td>
<td>Theory combined with experiment: Presents a method to define which policies to use to present information, based on earlier achievements, to solve complex reinforcement learning problems. The comparing and validation of the policies are done offline with the help of importance sampling method. Principal Component Analysis and Neural networks are combined in order to solve the problems with finding relevant solutions. This supports RQ1.6 but the paper do not pass criterion level 1 as support for RQ1.5.</td>
</tr>
<tr>
<td>[66]</td>
<td>1.3</td>
<td>Theory combined with experience: Using PROV-N and W7 to create a Service Provenance Ontology which could be used by service creators to supply all necessary provenance information of web services. Since the idea with provenance is to provide “information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness” this supports RQ1.3.</td>
</tr>
<tr>
<td>[75]</td>
<td>1.2, 1.4, (1.1, 1.3)</td>
<td>Theory combined with experiment: Presents a way to inferring relevant but not directly observed information from partial information in a situation specific context with the help of Bayesian theories as the underlying knowledge representation. This supports RQ1.2 and RQ1.4. How the needed context frame is described and added is not part of the paper. Neither is the constructions of the rules. This makes the paper not passing criterion level 1 as support for RQ1.1 and RQ1.3.</td>
</tr>
</tbody>
</table>
Appendix A. Appendix to Chapter 2

<table>
<thead>
<tr>
<th>Paper</th>
<th>RQ</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>[95]</td>
<td>1.1</td>
<td>Theory combined with experiment: Propose the use of Discrete Wavelet Transformation (DWT) in Decision Support Systems for event driven enterprises. According to the paper improvements of visualization is achieved due to the DWT’s ability to compress data without losing the details needed for analytical purposes. The possibility to visualize huge amount of information in order to support actors’ business intents supports RQ1.1.</td>
</tr>
<tr>
<td>[96]</td>
<td>1.5</td>
<td>Theory combined with experiment: Provides an algorithm that could be used to reason about the most preferred outcome for a compositional system both in terms of quantitative and qualitative preferences. The possibility to reason about preferred outcomes for a compositional system supports RQ1.5.</td>
</tr>
</tbody>
</table>

Table A.4.1: Comments on papers passing criterion level 1

Table A.4.2 lists papers not passing criterion level 1, the research questions a paper supports and comments on the paper.

<table>
<thead>
<tr>
<th>Paper</th>
<th>RQ</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>[4]</td>
<td>1.3, 1.4</td>
<td>Theory: The paper describes what is needed by a adaptive policy framework. There are some findings that might be of use for Intention Driven Systems.</td>
</tr>
<tr>
<td>[11]</td>
<td>1.1, 1.3</td>
<td>Theory combined with experiment: Constructing a tool “MeTAGeM” that supports Model Driven Development of model transformations.</td>
</tr>
<tr>
<td>[21]</td>
<td>1.3</td>
<td>Theory: Provides a theory of what is needed to be able to apply Digital Policy Management. This might be useful from an Intention Driven System’s point of view.</td>
</tr>
<tr>
<td>[23]</td>
<td>1.1</td>
<td>Theory combined with experiment: Using Multi Agent System to combine different context frames. The case based approach is used to find new outcomes. The agent structure is described. The different adaptations points when context frames are change are highlighted.</td>
</tr>
<tr>
<td>[25]</td>
<td>1.5</td>
<td>Theory: Introduce a cognitive model to agent coordination and cooperation with the help of Belief Desire Intent and Observe Orient Decide and Act.</td>
</tr>
<tr>
<td>[32]</td>
<td>1.1</td>
<td>Theory combined with experiment: Uses cognitive input to offload humans depending on their state. Using neural network and statistic analysis to determine the mental state of a human. No indications on how the context frame where deployed. This has some significance for a generic Intention Driven System.</td>
</tr>
<tr>
<td>[44]</td>
<td>1.1</td>
<td>Theory combined with experiment: Multi Criteria Decision Analysis based on an agent solution. The desire from one of the actors has to be correlated with the constraints to fulfill the desires by the second actor. There are no proposals on how to create the needed context frames and how to align the context frames.</td>
</tr>
<tr>
<td>[52]</td>
<td>1.1</td>
<td>Theory: Multi Agent System is used for integration of Decision Support Systems (DSS). The theory might be useful since integration of existing DSSs is part of a realistic Intention Driven System implementation. Different ontology constructs are discussed in the paper.</td>
</tr>
<tr>
<td>[55]</td>
<td>1.3</td>
<td>Theory: Policy based management based on OWL-S. Describing the input, output, preconditions and effects on a service and the resources used by the service. Using semantic query based network services to collect network status information.</td>
</tr>
<tr>
<td>[58]</td>
<td>1.5</td>
<td>Theory: How to narrow the gap between human (linguistic) fuzzy model and evolving fuzzy models. This is a position paper but the ideas might be useful.</td>
</tr>
</tbody>
</table>
Appendix A. Appendix to Chapter 2

<table>
<thead>
<tr>
<th>Reference</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[63]</td>
<td>1.1</td>
<td>Theory combined with experiment: The theory might be useful since it combines Natural Language Processing with knowledge and mental models. The focus of the paper is a component called “semantic de-biased associations model” which are using a semantic triple to combine concepts (expressed in Natural Language) with associations (a weighted meaning) and history (cases). The experiment is done with a proprietary tool.</td>
</tr>
<tr>
<td>[65]</td>
<td>1.3</td>
<td>Theory combined with experiment: Using Dynamic Bayesian Network together with risk estimation to present awareness to the operator in a way that makes it easier for the operator to take right decisions in a stressful situation. It is possible to simulate the effect of an action taken by the operator but the action will not be enforced from the system.</td>
</tr>
<tr>
<td>[80]</td>
<td>1.5</td>
<td>Theory combined with experiment: Uses Unified Modeling Language to model Multi Agent Systems and add a first class modeling element called multi-role interaction (mRI) in order to represent interactions abstractly. There are two static organization view models: Role model and Ontology model (not shown in the paper). Together with behavior organization view and traceability view. This is built in to a proprietary tool but the ideas are useful.</td>
</tr>
<tr>
<td>[82]</td>
<td>1.3</td>
<td>Theory: The paper reason about different ways (and schools) to represent knowledge. This fits with the problem of defining a context frame.</td>
</tr>
<tr>
<td>[86]</td>
<td>1.1</td>
<td>Theory combined with experiment: Extends Market Driven Agents with the capabilities to negotiate when the environment becomes open and dynamic.</td>
</tr>
<tr>
<td>[99]</td>
<td>1.3, 1.4</td>
<td>Theory combined with experiment: Using simulation to validate the effect of new parameter settings with the help of information from the real environment. A specific real time simulation protocol (ArenaRT) is used.</td>
</tr>
<tr>
<td>[102]</td>
<td>1.3</td>
<td>Theory combined with experiment: Using a holonic Multi Agent System to optimize the allocation of resources. This gives an insight into how an agent can play different roles depending on its interactions.</td>
</tr>
<tr>
<td>[103]</td>
<td>1.5</td>
<td>Theory: Describes a way to break down the information in different levels to make it more understandable and manageable. This is done in order to make sure that certain resources can provide information in order to meet a specific goal(s). This theory could be useful input to Intention Driven Systems.</td>
</tr>
<tr>
<td>[106]</td>
<td>1.1, 1.3</td>
<td>Theory combined with experiment: Describes how agents can change according the surroundings as well as to changes in goals. The solution is based on deliberative coherence theory. How the needed information for a context frame is entered and is not in the scope of this papers.</td>
</tr>
<tr>
<td>[108]</td>
<td>1.3</td>
<td>Theory: Focus on how agents can find the required skill from other agents when needed. A protocol for this issue is presented.</td>
</tr>
<tr>
<td>[110]</td>
<td>1.1, 1.3, 1.5</td>
<td>Theory: Challenges for Requirement Engineering in constructing Cyber Physical Systems. Definitions of terms and suggestions for using Natural Language Processing as a tool for the Requirement Engineering. The information is based on a Systematic Literature Review. This paper highlights the problems that have to be solved in an Intention Driven System.</td>
</tr>
</tbody>
</table>

Table A.4.2: Comments on papers not passing criterion level 1
Appendix B

Appendix to Chapter 3
Appendix B. Appendix to Chapter 3

B.1 Interview Instrument

Presentation of the subject (15 min)

Interview questions and discussion (90 min)

Initial open-ended questions:
- Could you describe the different viewpoints needed to support business intents?
- How would you like to divide the support for the business intents?

Intermediate questions:
- What characterize the ___?
- How could the ___ be grouped?
- Are there any specific actors attached to ___?
- Are ___ based on any standard?

Ending questions:
- Have you thought about how to structure the needed information?
- How do you intend to control/manage the information?
- Is there anything else I should know?
- Is there anything you would like to ask me?

Summary (15 min)
Appendix B. Appendix to Chapter 3

B.2 Study Execution

An initial interview with the subjects in the focus group and a follow-up meeting with the same subjects was conducted during a full day. During the analysis of the material several open questions arose. Six additional meetings had to be conducted with the focus group to be able to answer the open questions. Two external reviews were conducted in order to get a second opinion. Final conclusions were based on all the gathered data.

The nine iteration phases are described below and the “Why”, “How” & “What” are concepts outlined in Figure 3.3. Based on the feedback from the reviewers, corrections were made to the paper. A tenth phase was added to verify these corrections.

B.2.1 Phase 1: Initial meeting (February 2015)

The research question was presented to the subjects in the focus group. During the same day a follow-up meeting was held with the focus group subjects. The subjects presented and explained Figure 3.3 and introduced their thinking of how the Observe, Orient, Decide & Act loop (OODA) [87] (Figure 3.4) could be seen in the figure (Figure 3.3). An initial draft of the subjects’ definition of governance views, Definition 3.1, was presented by Subject 3. Governance refers to “The discipline of monitoring, managing, and steering a business (or IS/IT landscape) to deliver the business outcome required [73]”. The outcome was the base for the information in The Initial Information from Ericsson (Section 3.4.1).

The next step was to define how Figure 3.3 captures the “Why”, “How” & ‘What” of a business intent.

B.2.2 Phase 2: Defining Why & What (February 2015)

At the beginning of the meeting with the focus group subjects it was decided to focus on a deeper understanding of how Figure 3.3 captures the ”Why & What” of a business intent. The subjects used the Business Motivation Model [69] as a tool to relate the findings to a model available to the industry. An idea on how Pask’s conversation theory [78] could be used to capture the ”How” was presented and a first sketch on Figure 3.8 was made. The outcome formed the base for the information about value architecture and value creation in Section 3.4.1 and the starting point for Viewpoint Information (Section 3.4.3).
Appendix B. Appendix to Chapter 3

It was decided to verify if the results were understandable to subjects outside the focus group. Subject 4, who did not participate in the focus group meetings, accepted to review the results.

B.2.3 Phase 3: External review (March 2015)

Subject 4, who is not part of the focus group, reviewed the found results. The results were updated after a meeting with the reviewer and the members of the focus group. The outcome formed the final information for The Initial Information from Ericsson (Section 3.4.1).

The main remark from the review was the lack of the viewpoints’ visibility in Figure 3.3.

B.2.4 Phase 4: Defining How (April 2015)

The goal of the meeting with the subjects in the focus group was to make the viewpoints visible in Figure 3.3. A brainstorming meeting on how to make the viewpoints visible was conducted. During this meeting, Figure 3.3 evolved into Figure 3.5. The outcome of the meeting formed the information in Supporting Viewpoints (Section 3.4.2).

When the viewpoints were made visible, the concern about how to capture the information arose.

B.2.5 Phase 5: Context frame (April 2015)

The aim with the meeting with the subjects in the focus group was to gain an understanding of how to capture the viewpoint information. During the meeting the idea of using Pask’s conversation theory [78] to capture the ”How” was reused and became the base for the information in Viewpoint Information (Section 3.4.3).

The subjects decided to present the results to Ericsson research in order to find synergies and get a second opinion.
Appendix B. Appendix to Chapter 3

B.2.6 Phase 6: Presentation to Ericsson research (May 2015)

A presentation for members of Ericsson Research in business area Business Support Systems (Subject 5 and Subject 6) was performed. One remark was made on Figure 3.5, which was regarded hard to understand.

The need for scalability and modularization became evident during the presentation. These needs are valid both inside an enterprise and for the interactions between enterprises.

B.2.7 Phase 7: Composite context frame & value networks (June 2015)

The focus group decided to concentrate on scalability and modularization during this meeting. The compositional context frame (Figure 3.9) evolved from the context frame when scalability and modularization inside an enterprise was discussed. A model for business interactions in a value network was created in order to support scalability and modularization of business interactions between enterprises in a value network. The outcome of the meeting was the final information for Viewpoint Information (Section 3.4.3).

The remark from the meeting with Ericsson research on the how to understand Figure 3.5, was not addressed until Phase 8.

B.2.8 Phase 8: The model in Figure 3.6 (August 2015)

The focus of the meeting with the focus group was to explain Figure 3.5. The use of the OSI model [47] helped to evolve Figure 3.5 into Figure 3.6. The outcome was Figure 3.6.

The change from Figure 3.5 to Figure 3.6 introduced concerns about the state of the information regarding business interactions in a value network.

B.2.9 Phase 9: Business interactions (September 2015)

The purpose of the meeting with the focus group was to improve the model for business interactions in a value network. A new way to describe the business interactions in a value network was found. This gave the solution to how the business functions collaborate on business intents, Figure 3.7. The outcome of
the meeting was Figure 3.7. The information regarding business interactions in a value network was included in Section 3.4.2. This became the final information for Supporting Viewpoints (Section 3.4.2).

B.2.10 Phase 10: Verification of changes to the information due to review comments (October 2016)

The goal with the meeting with the focus group was to verify that the changes to the information were in line with their expected study results. There were no major findings during the meeting. When Figure 3.2 was presented to the subjects, Subject 1 added the idea of giving weights to the changes of the business intents in terms of value and risk.
B.3 Proof-of-concept

We would like to investigate if it is possible to support visual and logical verification of business rules and to generate executable business rules. We have chosen to investigate a limited part of an enterprise’s business rules. The business rules we have chosen to study are targeting recommendation support for value propositions, based on different business models. The recommendations are based on following five parts of the Osterwalder canvas [74]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions.

B.3.1 Introduction

Business rules [16] are used to govern how an enterprise does its business. Since humans are defining the business rules, these rules might be ambiguous. Visual and logical support to verify the correctness of the defined business rules are desired. Parts of the business rules could be made executable in order to make the operation of the business more efficient and effective. The process of translating business rules to executable business rules is error-prone due to human interpretation. It is desired to use a software algorithm to translate business rules into executable business rules. A way to execute business rules is to use a common rule engine for all the components in a business support system. This approach might not be desired or possible. Instead, the possibility to express executable business rules in different software language, which could be distributed to the different components in a business support system, might be an option.

We have evaluated if a machine learning pipe-line could be used to mitigate the problems stated above. The pipe-line consists of three parts. A pre-processing part, the use of a decision tree algorithm called ID3 [84], and a post-processing part. A pipe-line is a common construct in machine learning. It is used to chaining together different processing steps and machine learning models.

The pre-processing is responsible to prepare the business rule data for the processing done by the ID3 algorithm. Since the ID3 algorithm cannot handle continuous data the pre-processing makes it possible to mimic continuous data in a way that the ID3 algorithm can handle.

The ID3 algorithm is implemented according to the base algorithm. We have added a warning in the ID3 algorithm in order to notify the user if there might be ambiguities in the business rules. Normally the ID3 algorithm will use majority voting and continue with out any further notice.
The post-processing consists of three steps. The first step is responsible for visualizing the result from the ID3 algorithm in a tabular format as well as in a tree format. The second step will create an executable model of the business rules. The third step will use the executable model for the logical validation of the business rules. Since the model is stored in an executable format, the model could be distributed to components in a business support system.

The pipe-line has proved to be valuable in order to mitigate the problems stated above.

**B.3.2 Methodology**

We conducted a proof-of-concept to evaluated if it is possible to use a machine learning pipe-line to support visual and logical verification of business rules, and generate executable business rules.

The nature of the business rule data can range from, for example integers, product identifiers, date and time intervals, etc. This requires handling of features with numerical-, categorical-, and nominal- characteristics. For each row there shall be two different classification columns. These classification columns respectively represents eligible objects and the allowed actions on these objects. The idea is borrowed from the gaming industry where a specific context gives the character the possibilities to, for example find specific treasures and stipulates how these treasures can be handled. As an illustration we use two business rules: “A customer must be offered products which it is eligible for.” and “An agent’s manager can offer a customer 5 percent discount on the total price of an order item.”. These two business rules are supported by two business rules which supports definitions: “A customer’s eligibility for a product is based on the business relationship type.” and “The total price of an order item is always computed as the product unit price times its quantity.”.

There are no real-time requirements on the generation of the executable business rules. The solution should be implemented in a language which is used by Ericsson. We decided to use Python since it have many machine learning contributions, for example Scikit [97].

Examples of business rules which could be valid for recommending value propositions, based on different business models, were provided to us. The examples were delivered as csv-files. These examples were used to verify the solution. The verification of the solution was done together with practitioners from Ericsson.

To avoid the risk of imposing solutions from the authors, member checking [89] was used in the form of presentations to the involved practitioners. The negative
case analysis [89] was used to challenge the solution and counter researcher bias. The solutions were presented to and feedback given by, the involved practitioners at different stages of the study. The selection of test data was performed by the involved practitioners.

B.3.3 Results

We have implemented a machine learning pipe-line which makes it possible to conduct visual and logical verification of business rules, and generate executable business rules. We have added different types of functionality which is regarded as needed when handling business rules. Missing data is handled as a wild card, i.e. all values are true. Continuous values have a defined boundary and there are no value gaps in the data. Since a human is defining the business rules, entering all possible combinations by hand is not an option. A meta-data file is used to describe the nature of the features. The ability to generate executable code, which can be distributed in a system, is another requirement. The pipe-line consists of three parts. A pre-processing part, the use of a decision tree algorithm called ID3, and a post-processing part.

The first step of the pre-processing is responsible for reading in the data and the meta-data. The number of feature columns is not restricted by the software. The next step is to transform missing data into a wildcard character. The result is shown in a tabular format. The third step uses the meta-data to determine the feature columns with continuous characteristics. The values in these feature columns are converted into a format which mimic continuous data. The fourth step splits the data set into two sets, each responsible for one of the classification columns. These classification columns respectively represents eligible objects and the allowed actions on these objects. A decision tree for each of these data sets are created with the help of the ID3 algorithm.

The ID3 algorithm is implemented without any deviations from its specification [84]. In the implementation we use Shannon entropy as the impurity measure [35]. However, we have added a warning message if majority voting is used since this might indicate ambiguity in the business rule set. The result is a Python dictionary which represents the resulting model of the business rules.

The first step of the post-processing is responsible for creating a dot-notation based on the model. Graphviz is used to present the model as a tree graph. The tree graph makes it easier for the responsible actors to understand, and analyze the results of the defined business rules. A presentation of the business rules in a tabular format is done as well. The second step uses the model to create executable Python code which represents the model of the business rules.
This code is displayed to the user. The last step gives the possibility to make predictions against the model. The predictions are made against the executable Python code. The executable model of the business rules is persisted as a Python module. This makes it possible to distribute the executable model to different components in a business support system.

The aim is to use the executable business rules to guide or create business processes which are responsible to adhere to these rules. Since these business rules can be distributed, the use of a monolithic rule engine can be avoided. The business rules can be defined in a tool which can export its content as a csv-file, for example Excel. Business rules are often declarative in their nature. In order to for the solution to exist in an event-driven environment, the data was extended with the events a business rule is intended for. Contrary to many proposed solutions, we believe that a business rule could be triggered by several different events. As an illustration we use a business rule which states: A customer must be assigned to an agent if the customer has placed an order. This business rule will be evaluated at, at least, two different events; when a customer places an order and when an agent resigns.

B.3.4 Analysis

The pipe-line is considered as a proof of concept, and as such is regarded as successful by the involved practitioners. The pipe-line makes it possible to, visually and logically, validate the correctness of the business rules before they are put in production. The possibility to generate executable code representing the model of the business rules makes it possible to execute the same model in different components without the need of re-implementation. This might improve the coherence of the business rules in a business support system.

There are several improvements to the pipe-line which should be considered. The precision of the limits have to be based on the precision of the affected feature value. The precision has to be configurable feature by feature, and with different values for the maximum limit and minimum limit. The executable code representing the model of the business rules should support additional languages, for example Java™.

B.3.5 Discussion

The decision trees’ over-fitting problem is turned into an advantage when we want to evaluate or execute defined business rules.
An advantage of the ID3 algorithm is the use of multi-split instead of the binary split. The multi-split gives a better overview of the model, from a human perspective. Another advantage with the multi-split is the rather trivial procedure to produce executable code based on the model.

Business rules requires wild cards, explicit boundaries, and features with no gaps between the values. Scikit provides an implementation of decision trees based on the CART [14] algorithm. We have tried to use this implementation but it is not suitable for the mixture of the data and the additional requirements we are putting on the data. To our understanding, the C4.5 algorithm [111] shares the same limitations as the CART algorithm, regarding our requirements on the data. Instead of changing an existing algorithm, we decided to use a base implementation of a simple algorithm. In this case we do not need to verify the algorithm, but rather concentrate on the desired functionality of the pipeline.

Is a machine learning pipe-line for business rules a correct approach, and is it worth the effort? The response from the practitioners indicates a usefulness of machine learning algorithms in the business rule domain. The decision tree algorithm cannot be used for all types of business rules. Instead we have to investigate which machine learning algorithms can be useful for each specific business rule type. We believe that most of the pre- and post-processing functionality is useful if we decide to implement other types of business rules with the help of rule sets, and expose the result as trees [35] as well as generate executable business rules.

There are no real-time requirements on the transformation from business rules to executable business rules. Since the data set will vary in the number of used feature columns and since each feature has its own characteristic, the use of a typed-language is not ideal. Based on this, Python fulfills the requirements as a suitable implementation language for the problem at hand.

There are several improvements which have to be done before the concepts in the proof-of-concept could be part of a business support system implementation. The main improvements are described here. The Governance have to provide a repository where rules can be examined and compared to avoid or solve contradictions. The deployment of the business rules needs to be supported by business processes which can handle error conditions etc. The Design environment needs a process which could guide the designers through the processes of creating executable business rules. The context frame meta-model should be able to act as an execution container in order to support more advanced rules.
Appendix B. Appendix to Chapter 3

B.3.6 Conclusion

The proof-of-concept shows that it is possible to support visual and logical verification of business rules, and to generate executable business rules with the help of a machine learning pipe-line.
Bibliography


Advances in Production Management Systems. Innovative and
Stefan Wiesner et al.
Design Science Methodology for Information Systems and Software Design
Roel Wieringa.
Xindong Wu et al.
Top 10 algorithms in data mining
Alan N. Steinberg and Christopher L. Bowman. "Adaptive context discovery and exploita-
Neumann
TMForum.
Restrepo et al.
Japanese Journal of Management
Scikit-learn.org.

ABSTRACT

Context: Software supporting an enterprise’s business, also known as a business support system, needs to support the correlation of activities between actors as well as influence the activities based on knowledge about the value networks in which the enterprise acts. This can be supported with the help of intent-driven systems. The aim of intent-driven systems is to capture stakeholders’ intents and transform these into a form that enables computer processing of them. Only then are different machine actors able to negotiate with each other on behalf of their respective stakeholders and their intents, and suggest a mutually beneficial agreement.

Objective: When building a business support system it is critical to separate the business model of the business support system itself from the business models used by the enterprise which is using the business support system. The core idea of intent-driven systems is the possibility to change behavior of the system itself, based on stakeholder intents. This requires a separation of concerns between the parts of the system used to execute the stakeholder business, and the parts which are used to design the business based on stakeholder intents. The business studio is a software that supports the realization of business models used by the enterprise by configuring the capabilities provided by the business support system. The aim is to find out how we can support the design of a business studio which is based on intent-driven systems.

Method: We are using the design science framework as our research framework. During our design science study we have used the following research methods: systematic literature review, case study, quasi experiment, and action research.

Results: We have produced two design artifacts as a start to be able to support the design of a business studio. These artifacts are the models and quasi-experiment in Chapter 3, and the action research in Chapter 4. The models found during the case study have proved to be a valuable artifact for the stakeholder. The results from the quasi-experiment and the action research are seen as new problem solving knowledge by the stakeholder.

Conclusion: The synthesis shows a need for further research regarding semantic interchange of information, actor interaction in intent-driven systems, and the governance of intent-driven systems.