

## Supporting Continuous Changes to Business Intents

Johan Silvander

*Software Engineering Research Lab Sweden, Blekinge Institute of Technology, Karlskrona,  
Sweden  
johan.silvander@bth.se*

Magnus Wilson

*Software Engineering Research Lab Sweden, Blekinge Institute of Technology, Karlskrona,  
Sweden  
magnus.wilson@bth.se*

Krzysztof Wnuk

*Software Engineering Research Lab Sweden, Blekinge Institute of Technology, Karlskrona,  
Sweden  
krzysztof.wnuk@bth.se*

Michael Svahnberg

*Software Engineering Research Lab Sweden, Blekinge Institute of Technology, Karlskrona,  
Sweden  
mikaelsvahnberg@bth.se*

**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 result will affect how Ericsson will build the business studio for their next generation business support systems.

**Keywords:** business intent; actor; viewpoint; business support system; intent-driven system; context frame; compositional system; knowledge creation; case study; continuous change.

## 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 [1]". 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 [2]". 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 [4]. Knowledge creation is obtained through interactions between actors [5]. 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 [6]. The construction of such a system has to support changes to a system's policies and rules in ways that are effective and efficient [7]. 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 [8] as a model to describe intent-driven systems (see Section 2.1). Based 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 [9]. 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 busi-

ness 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 2 the background is explained. In Section 3, background information about the case context, the study design, the execution, and the validity threats are presented. The result is presented in Section 4, the analysis is presented in Section 5 and a discussion is provided in Section 6. Finally, in Section 7, a summary and the conclusions are provided.

## 2. Background

This study is conducted as part of a design science [10] 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 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.

### 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 [8] 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 [11] and is used in different fields [12, 13, 14].

The idea of an intent-driven system as chains of interacting loops with one or several cycles is shown in Figure 1 (The original contributor of a figure is noted in the figure's caption text.). Figure 1 shows how two human actors (A and B) interact 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 about 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.

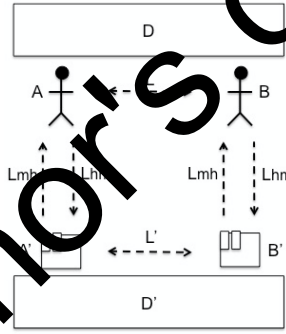


Fig. 1. An explanation model of intent-driven systems based on Pask's conversation theory (Contribution from the authors).

The language 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 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." [15].

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." [15].

Pask [8] 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 [16]. 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 [17] 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 [18] in eTOM. ITIL started as best practice processes and do not cover the business design parts of an enterprise. The TMG's Business Motivation Model (BMM) [4] 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 [19].

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 certain extent, desired outcome. Combinations of interactions between more than one real world actor exists in the literature [20, 21, 22], 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 [5]. The interactions between groups are supported in Pask's conversation theory [8].

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 [9], and a formal way of expressing the policies and rules, e.g. using Semantics of Business Vocabulary and Rules (SBVR) [23]. 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 [24].

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. [9], Beigi et al. [25], and Lewis et al. [26].

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 [17] is a process map and TMForum's TAM [27] is an application map the different parts of these maps could be used as part of a context frame. The contexts (Why, How, What, Who, Where and When) in the Zachman Framework [28] could be seen as an integrated part of the context frame.

## 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 2. In Figure 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  $Lhm_A$ . 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  $Lmh_A$ . This initiates a cycle where Stakeholder A interacts with Component A' until a common understanding of the business intents are accurately described in  $Lmh_A$ . 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.

[29] 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 [8] 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 [16]. We define a *context frame* as the total domain information for a specific domain an actor has obtained.

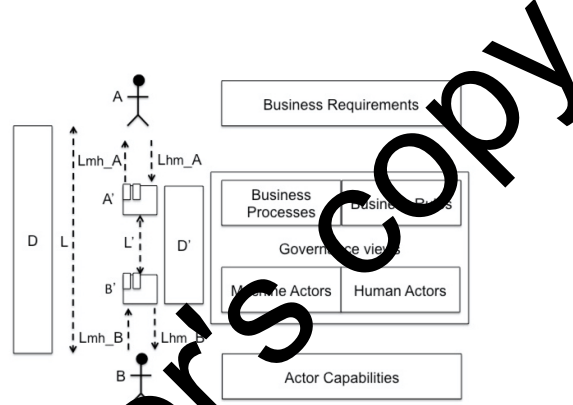


Fig. 2. An example of initial steps for intent-driven systems (Contribution from the authors).

### 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. [30] using a focus group approach [31, 32] in order to mimic the workflow used in the studied company. Information was captured in documents, drawings, and photos. We applied open coding technique [33] to analyze the collected qualitative data.

The case study was influenced by the spiral case study process [30] 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, adaptations 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.



Table 1. Description of the subjects' roles (F=Focus Group member, R=Reviewer), subject id and experiences.

Role	Id	Experience
F	1	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.
F	2	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.
F	3	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.
R	4	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.
R	5	More than 15 years of experience in the telecommunication industry, mainly engaged in product development work and driver of research activities at Ericsson.
R	6	More than 10 years of experience in the information technology industry, mainly engaged in research and consultancy.

### 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 [34] and was part of a design science [10] 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 model. 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.

### 3.2. 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 [30]. The unit of analysis is Ericsson's business studio.

#### 3.2.1. 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 [35]. 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 1. The first three subjects were part of the focus group and the rest of the subjects participated in reviews and review meetings only.

### 3.2.2. Data collection

The data was collected during focus group [31] 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 [32] with the interview questions described in Appendix A.

The views of the subjects became available during the interviews and captured as written text and pictures. The validation and correction of the captured 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, [4] it was selected as a foundation to explain how an enterprise's business intents could be expressed and realized.

### 3.2.3. 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 [33] 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 4.1 - Section 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.

## 4. Results

Below, the results obtained during the different phases (Appendix B) are presented. The results are divided into three sections, Section 4.1 - Section 4.3, each describing different aspects of the results.

### 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 which describes the design cycles of an enterprise. The text "Why", "What", "How" and "Characteristics of How"

was initially not part of Figure 3. The text was added during the study to be able to explain Figure 3 in a better way.

The “Why” row in Figure 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<sup>a</sup> represent the life cycle changes of the business intents on different levels in the enterprise.” Subject 3.

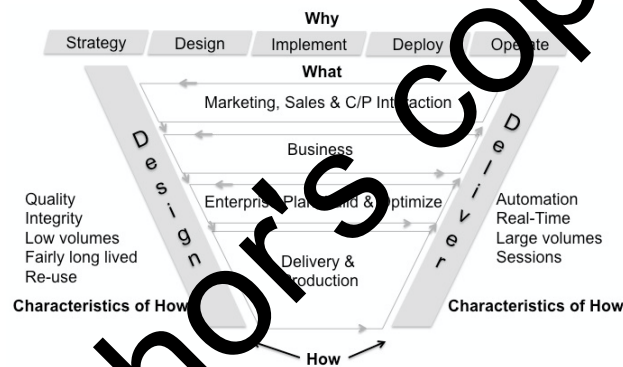


Fig. 3. The initial picture from Ericsson with explanations included (Contribution from the subjects).

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. Characteristics refers to “Showing the special qualities or traits of a person, thing, or group [36]”.

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

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

<sup>a</sup>The gray arrows are part of “What”, our note.

The overview explanation to the dynamics in Figure 3 was; the enterprise can be seen as an OODA-loop [37] (Figure 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. 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.

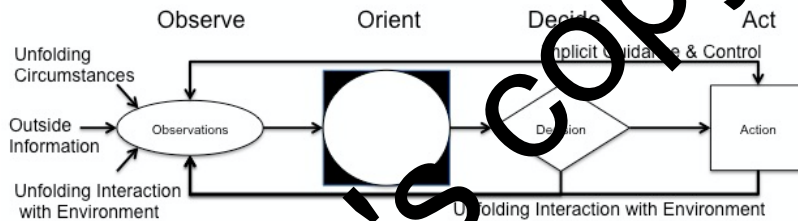


Fig. 4. A simplified version of the OODA loop [37] (Contribution from the subjects).

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 1.

The aim with Figure 3 is to create relevant business values for the enterprise. A value architecture [38] 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, which consists of the five main parts Strategy, Design, Implement, Deploy and Operate. Initially, the main parts were called phases.

“The value architecture should be structured in five main parts Strategy, Design, Implement, Deploy and Operate.” Subject 3.

The “Why” row in Figure 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 [4] as shown in Table 2. The mapping reflects the statement:

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 1. Governance View

Table 1. Mapping between Value Architecture and BMM.

Value Architecture Component	BMM
Strategy	End
Design	Means
Implement	The values guiding the construction of Organization Units, Business Processes and Directives.
Deploy	The values guiding the introduction of Organization Units, Business Processes and Directives.
Operate	The values guiding the use of Organization Units, Business Processes and Directives.

“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 [39]”. 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 [39]”.

“The business functions are where the value creation is taking place.” Subject 2.

In Figure 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 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.

How viewpoints can be supported is covered by the next section (Section 4.2).

#### 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 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 was transformed into Figure 5. The viewpoints are the intersections between value architecture components and business functions (The squares in Figure 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.

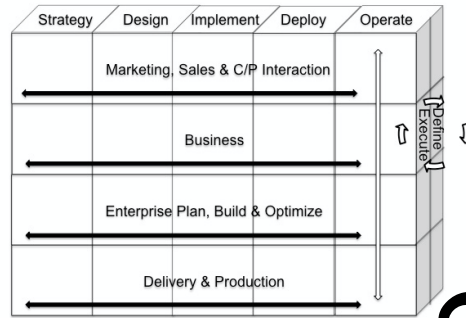


Fig. 5. The initial visualization of the viewpoints as the intersection between a value architecture component and a business function (Contribution from the authors).

With the help of Pask's conversation theory [8], 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 handled 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 [40]. RACI is an abbreviation for Responsible, Accountable, Consulted and Informed. These are the responsibilities and actors could be assigned for a specific task. This 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 [41] helped evolve Figure 5 into Figure 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 [42]. 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 TO-GAF?” Subject 2.

Figure 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 [43]. 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). 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 arrow) 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 1). The aspects of the governance views are part of Figure 6.

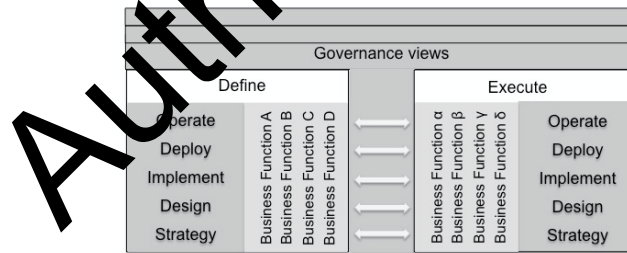


Fig. 6. A model providing continuous definition and execution of business intents in a governed way (Contribution from the subjects).

In Figure 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 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 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.

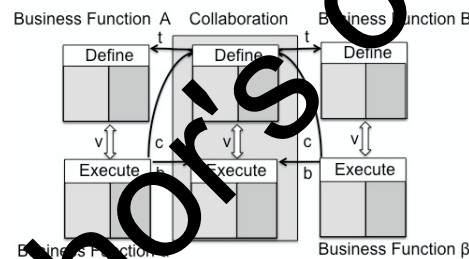


Fig. 7. The model in Figure 6 supporting business interactions and negotiations between business functions (Contribution from the authors).

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 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 4.3).

### 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 [8] and Erlang [44]. 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 [8]. 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 8. In Figure 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 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, knowl-

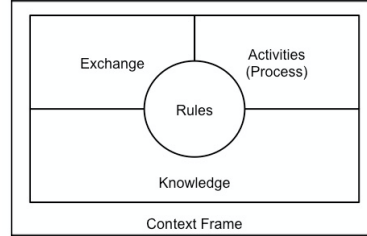


Fig. 8. The Context Frame meta-model (Contribution from the subjects).

edges, events, and activities. This brings us to the level of compositional context frames.

The composite context frame in Figure 9 is used for capturing the recursive nature of an actor, interactions between groups of actors as well as for meta-level structures. Figure 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 [5]. 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.

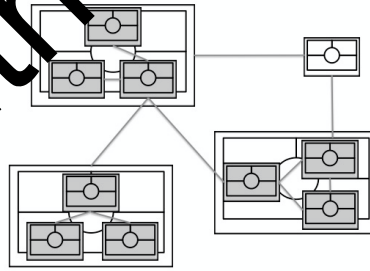


Fig. 9. The Composite Context Frame meta-model (Contribution from the subjects).

## 5. Analysis

During the analysis we created Figure 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 charg-

ing interval for a customer segment, several business processes and actors, both humans and machines, might be identified as affected by the requirement (Figure 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 6. A realization of the requirement requires collaborations between the affected business functions, as illustrated in Figure 7, with the help of a language  $L$  in Figure 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 7 and the horizontal arrows in Figure 6 will make use of the functionality described by Figure 2. This unfolds the recursive nature of Figure 2.

We build upon Pask's conversation theory [8] by introducing the context frame, Figure 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 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 2) could be realized.

## 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 [8] 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 2 can be ranked in descending order of expressiveness starting with the machine actors. Machine actors are often describe 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 [45]. 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 [46]. To obtain the possibility of a high level of automation of decision and action selection [29] 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 [24].

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 interaction is desirable.

### 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 [47]: 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 2) we have implemented a machine learning pipe-line based on the ID3 algorithm [48]. 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.

The Design in Figure 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 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 C.

## 6.2. *Validity threats*

The methodology suggested by Robson [32] to analyze threats to the validity and the corresponding counter measures is used.

### 6.2.1. *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 [32] was used continuously in the form of review meetings at different stages of the study. Moreover, applying open coding technique [33] 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.

### 6.2.2. *Interpretation*

To mitigate the risk of imposing 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 [32], and subjects with different roles and responsibilities. This, together with a distinct audit trail and the use of a open coding technique [33] 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.

### 6.2.3. *Biased theories*

The negative case analysis [32] was used to challenge the theories and counter researcher bias. The theories were presented to and reviewed by the subjects and the authors.

### 6.2.4. *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.

### 6.2.5. 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 [30] 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.

## 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 6). Figure 6 is anchored in the focus group through an iterative and incremental series of workshops. The model in Figure 6 is supported by Figure 2 which describes the initial steps of intent-driven systems. Figure 2 encapsulates the context frame meta-model (Figure 8 and Figure 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.

## References

- [1] Investopedia. *Investopedia*. <http://www.investopedia.com/terms/v/value-network.asp> Last checked 2015-12-01. 2015. (Visited on 06/20/2015).
- [2] S Geman. "Hierarchy in machine and natural vision". In: *Proceedings of the 11th Scandinavian Conference on Image Analysis* (1999), pp. 1–13.
- [3] William Aspray, Reinhrd Keil-Slawik, and David L. Parnas. "Position papers for Dagstuhl Seminar 9635 on History of Software Engineering". In: *History of Software Engineering* (1997), p. 61.
- [4] OMG. *Business Motivation Model*. <http://www.omg.org/spec/BMM/1.2/> Last checked 2015-12-01. 2014. (Visited on 04/20/2015).

## 24 REFERENCES

- [5] Ikujiro Nonaka, Ryoko Toyama, and Noboru Konno. "SECI, Ba and Leadership: a Unified Model of Dynamic Knowledge Creation". In: *Long Range Planning* 33.1 (2000), pp. 5–34. DOI: 10.1016/S0024-6301(99)00115-6.
- [6] M.J. DiMario, J.T. Boardman, and B.J. Sauser. "System of Systems Collaborative Formation". In: *IEEE Systems Journal* 3.3 (Sept. 2009), pp. 360–368. DOI: 10.1109/JSYST.2009.2029661.
- [7] Scott Ferguson et al. "Flexible and Reconfigurable Systems: Nomenclature and Review". In: *Volume 6: 33rd Design Automation Conference, Parts A and B*. ASME, 2007, pp. 249–263. ISBN: 0-7918-4807-8. DOI: 10.1115/DETC2007-35745.
- [8] Gordon Pask. *Conversation Theory*. Elsevier, 1976. ISBN: 0-444-41424-X.
- [9] Stefan Wiesner et al. *Advances in Production Management Systems. Innovative and Knowledge-Based Production Management in a Global-Local World*. IFIP Advances in Information and Communication Technology PART 1 p 281-288. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 281–288. ISBN: 978-3-662-44738-3. DOI: 10.1007/978-3-662-44739-0.
- [10] Alan R. Hevner et al. "Design Science in the Information Systems Research". In: *MSI Quarterly* 28.1 (2004), pp. 75–105.
- [11] Nigel Ford. "Modeling cognitive processes in information seeking: From Popper to Pask". In: *Journal of the American Society for Information Science and Technology* 55.9 (July 2004), pp. 769–782. DOI: 10.1002/asl.20021.
- [12] Usman Haque. "The Architectural Relevance of Gordon Pask". In: *Architectural Design* 77.4 (2007), pp. 54–61. DOI: 10.1002/ad.487.
- [13] Aleksey Heinze, Chris Procter, and Eduardo Scott. "Use of conversation theory to underpin blended learning". In: *International Journal of Teaching and Case Studies* 1.1/2 (2007), pp. 108–120. DOI: 10.1504/IJTCS.2007.014213.
- [14] G.M. Boyd. *Conversation Theory*. Lawrence Erlbaum Associates, 2004, pp. 179–197. ISBN: 1410609510.
- [15] Berner Lindstrom. *Learning Styles and Learning Strategies*. Tech. rep. 2. Goteborg Univ., Mondal (Sweden). Dept. of Education., 1983, p. 41.
- [16] Matthias Baldauf, Scharam Dustdar, and Florian Rosenberg. "A survey on context-aware systems". In: *International Journal of Ad Hoc and Ubiquitous Computing* 2.4 (2007), p. 283. DOI: 10.1504/IJAHUC.2007.014070.
- [17] TM Forum. *as401*. <https://www.tmforum.org/business-process-framework/> Last checked 2015-12-01. 2015. (Visited on 07/27/2015).
- [18] ITIL. *ITIL*. <http://www.itil-officialsite.com/> Last checked 2015-12-01. 2015. (Visited on 07/27/2015).
- [19] Luz Maria Priego-Roche, Agnès Front, and Dominique Rieu. "A framework for virtual organization requirements". In: *Requirements Engineering* (2015). DOI: 10.1007/s00766-015-0223-5.
- [20] J Colloc and C Sybord. "A multi-agent approach to involve multiple knowledge models and the case base reasoning approach in decision support systems". In: *Proceedings of the 35th Southeastern Symposium on System Theory, 2003*. Piscataway, NJ, USA: IEEE, 2003, pp. 247–251. ISBN: 0-7803-7697-8. DOI: 10.1109/SSST.2003.1194567.
- [21] W. Hu et al. "Corporate dashboards for integrated business and engineering decisions in oil refineries: An agent-based approach". In: *Decision Support Systems* 52.3 (Feb. 2012), pp. 729–741. DOI: 10.1016/j.dss.2011.11.019.
- [22] Rajiv Kishore, Hong Zhang, and R. Ramesh. "Enterprise integration using the agent paradigm: foundations of multi-agent-based integrative business information sys-



- tems". In: *Decision Support Systems* 42.1 (Oct. 2006), pp. 48–78. DOI: 10.1016/j.dss.2004.09.011.
- [23] OMG. *Semantics of Business Vocabulary and Rules*. <http://www.omg.org/spec/SBVR/1.2/> Last checked 2015-12-01. 2013. (Visited on 03/12/2015).
- [24] Dirk Heyne and Lars Mönch. "An agent-based planning approach within the framework of distributed hierarchical enterprise management". In: *Journal of Management Control* 22.2 (Nov. 2011), pp. 205–236. DOI: 10.1007/s00187-011-0140-9.
- [25] M.S. Beigi, S. Calo, and D. Verma. "Policy transformation techniques in policy-based systems management". In: *Proceedings. Fifth IEEE International Workshop on Policies for Distributed Systems and Networks, 2004. POLICY 2004*. IEEE, 2004, pp. 13–22. ISBN: 0-7695-2141-X. DOI: 10.1109/POLICY.2004.1309446.
- [26] David Lewis et al. "Semantic-Based Policy Engineering for Autonomic Systems". In: *Lecture Notes in Computer Science*. Vol. 3457. 2005, pp. 152–164. DOI: 10.1007/11520184\_12.
- [27] TMForum. *TAM*. <https://www.tmforum.org/application-framework/> Last checked 2015-12-01. 2015.
- [28] John Zachman. "Zachman Framework". In: *IBM Systems Journal* 26.3 (1987), pp. 276–292.
- [29] Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens. "A model for types and levels of human interaction with automation." In: *IEEE transactions on systems, man, and cybernetics. Part A. Systems and humans : a publication of the IEEE Systems, Man, and Cybernetics Society* 30.3 (2000), pp. 286–297. DOI: 10.1109/3468.844354.
- [30] P Runeson et al. *Case study research in software engineering: Guidelines and examples*. Hoboken, N.J.: Wiley, 2011. ISBN: 978-1-118-10435-4.
- [31] Jonathan Zrake et al. "Numerical Simulations of Driven Supersonic Relativistic MHD Turbulence". In: *Annual Review of Sociology*. Vol. 22. 2011, pp. 102–105. DOI: 10.1063/1.3623448.
- [32] C Robson. *Real world research: A resource for users of social research methods in applied settings*. 3rd. Chichester: Wiley, 2011. ISBN: 978-1-4051-8240-9.
- [33] K Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis*. London: SAGE, 2006. ISBN: 978-0761973539.
- [34] Gartner. *Customer Magic Quadrant*. <http://www.ericsson.com/ourportfolio/telecom-operators/operations-and-business-support-systems> Last checked 2015-12-01. 2014. (Visited on 05/04/2015).
- [35] Michael Quinn Patton. *Qualitative Research and Evaluation Methods*. 3rd ed. SAGE Publications Ltd, 2002. ISBN: 978-0761919711.
- [36] Merriam-Webster. *Merriam-Webster*. <http://www.merriam-webster.com/> Last checked 2015-12-01. 2015.
- [37] Chet Richards. "Boyd's OODA Loop". In: *Proceedings of Lean Software and Systems Conference 2011*. 2011, pp. 127–136.
- [38] Mutaz M. Al-Debei and Guy Fitzgerald. "The Design and Engineering of Mobile Data Services: Developing an Ontology Based on Business Model Thinking". 2010.
- [39] Open Group. *Open Group Definitions*. <http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html> Last checked 2015-12-01. 2013. (Visited on 04/28/2015).
- [40] Tom Costello. "RACI&#x2014;Getting Projects "Unstuck"". In: *IT Professional* 14.2 (Mar. 2012), pp. 64–63. DOI: 10.1109/MITP.2012.41.

## 26 REFERENCES

- [41] ITU-T. *X.200*. <http://www.itu.int/rec/T-REC-X.200-199407-I>. 1994. (Visited on 10/11/2016).
- [42] OMG. *TOGAF 9.1*. <http://pubs.opengroup.org/architecture/togaf9-doc/arch/index.html> Last checked 2015-12-01. 2011. (Visited on 04/20/2015).
- [43] Ikujiro Nonaka. “A Dynamic Theory of Organizational Knowledge Creation”. In: *Organization Science* 5.1 (Feb. 1994), pp. 14–37. DOI: 10.1287/orsc.5.1.14.
- [44] Joe Armstrong. “Making reliable distributed systems in the presence of software errors”. PhD thesis. Royal Institute of Technology, Stockholm, Sweden, 2003, p. 295.
- [45] OMG. *Business Process Model and Notation*. <http://www.bpmn.org/> Last checked 2015-12-01. 2014. (Visited on 04/28/2015).
- [46] Krisztian Balog et al. “Expertise Retrieval”. In: *Found. Trends Inf. Retr.* 6.2&#8211;3 (2012), pp. 127–256. DOI: 10.1561/15000000024.
- [47] Alexander Osterwalder and Yves Pigneur. *Business Model Generation*. 1st. Wiley, 2010. ISBN: 9780470876411.
- [48] J. R. Quinlan. “Induction of Decision Trees”. In: *Machine Learning* 1.1 (1986), pp. 81–106. DOI: 10.1023/A:1022643204877.
- [49] Business Rules Group. *The Business Rules Manifesto*. <http://businessrulesgroup.org/brmanifesto/BRManifesto.pdf> (Last accessed 20170201). 2003.
- [50] Scikit-learn.org. *Scikit Learn*. <http://scikit-learn.org/stable/index.html> (last checked 2016-11-18). 2016.
- [51] Peter Flach. *Machine Learning*. Cambridge University Press, 2012. ISBN: 9781107422223.
- [52] Leo Breiman et al. *Classification and Regression Trees*. Wadsworth & Brooks, 1984. ISBN: 9780412048418.
- [53] Xindong Wu et al. *Top 10 algorithms in data mining*. Vol. 14. 1. 2008, pp. 1–37. ISBN: 1011500701. DOI: 10.1007/978-1-0115-007-0114-2.

## 8. Appendix A: 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)

Author's copy

## 9. Appendix B: 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. Based on the feedback from the reviewers, corrections were made to the paper. A tenth phase was added to verify these corrections.

### 9.1. 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 and introduced their thinking of how the Observe, Orient, Decide & Act loop (OODA) [34] (Figure 4) could be seen in the figure (Figure 3). An initial draft of the subjects’ definition of governance views, Definition 1, was presented by Subject 1. Governance refers to “The discipline of monitoring, managing, and steering a business (or IS/IT landscape) to deliver the business outcome required [39]”. The outcome was the base for the information in The Initial Information from Ericsson (Section 4.1).

The next step was to define how Figure 3 captures the “Why”, “How” & “What” of a business intent.

### 9.2. 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 captures the “Why & What” of a business intent. The subjects used the Business Motivation Model [4] as a tool to relate the findings to a model available to the industry. An idea on how Pask’s conversation theory [8] could be used to capture the “How” was presented and a first sketch on Figure 8 was made. The outcome formed the base for the information about value architecture and value creation in Section 4.1 and the starting point for Viewpoint Information (Section 4.3).

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.

### 9.3. 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 4.1).

The main remark from the review was the lack of the viewpoints' visibility in Figure 3.

#### **9.4. 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. A brainstorming meeting on how to make the viewpoints visible was conducted. During this meeting, Figure 3 evolved into Figure 5. The outcome of the meeting formed the information in Supporting Viewpoints (Section 4.2).

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

#### **9.5. 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 [8] to capture the "How" was reused and became the base for the information in Viewpoint Information (Section 4.3).

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

#### **9.6. 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 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.

#### **9.7. 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 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 4.3).

The remark from the meeting with Ericsson research on the how to understand Figure 5, was not addressed until Phase 8.

**9.8. Phase 8: The model in Figure 6 (August 2015)**

The focus of the meeting with the focus group was to explain Figure 5. The use of the OSI model [41] helped to evolve Figure 5 into Figure 6. The outcome was Figure 6.

The change from Figure 5 to Figure 6 introduced concerns about the state of the information regarding business interactions in a value network.

**9.9. 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 7. The outcome of the meeting was Figure 7. The information regarding business interactions in a value network was included in Section 4.2. This became the final information for Supporting Viewpoints (Section 4.2).

**9.10. Phase 10: Verification of changes to the information due to review comments (October 2015)**

The goal with the meeting with the focus group was to verify that the changes to the information were in line with the expected study results. There were no major findings during the meeting. When Figure 2 was presented to the subjects, Subject 1 added the idea of giving weight to the changes of the business intents in terms of value and risk.

## 10. Appendix C: 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 [47]: customer type, customer relationship, channels, revenue streams, and a specific area of the value propositions.

### 10.1. Introduction

Business rules [49] 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 [48], 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

## 32 REFERENCES

above.

### 10.2. Methodology

We conducted a proof-of-concept to evaluate 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 [50].

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 [32] was used in the form of presentations to the involved practitioners. The negative case analysis [32] 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.

### 10.3. 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 [48]. In the implementation we use Shannon entropy as the impurity measure [51]. 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.

#### 10.4. 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<sup>TM</sup>.

#### 10.5. 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 require wild cards, explicit boundaries, and features with no gaps between the values. Weikit provides an implementation of decision trees based on the CART [52] 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 [53] 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 sets, and expose the result as trees [51] 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.

#### **10.6. 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.