

Requirements Engineering for Capability Driven Development

Jelena Zdravkovic¹, Janis Stirna¹, Jan-Christian Kuhr², and Hasan Koc³

¹Department of Computer and Systems Sciences, Stockholm University, Forum 100,
SE-16440, Kista, Sweden

jelenaz@dsv.su.se, js@dsv.su.se

²SIV Software-Architektur & -Technologie GmbH, DE-18184 Roggentin, Germany

jan-christian.kuhr@siv.de

³University of Rostock, Institute of Computer Science

Albert-Einstein-Straße 22, 18059 Rostock, Germany

hasan.koc@uni-rostock.de

Abstract. Lately, the notion of capability has emerged in IS engineering as an instrument to context dependent design and delivery of business services. Representing core business functionalities of an organization, capabilities, and capability driven IS development can be seen as both – a shift beyond and complement to the widely established service-oriented engineering paradigm where needs of customers form the leading modeling and design perspective. To ensure the needs of business stakeholders for variety of business contexts that an organization faces, and thus facilitate successful systems delivery, capability-driven development needs a well-defined method for requirements engineering, as well as its confirmation in practices. In this paper a process for specifying requirements capabilities and their designs is proposed. An application of the proposed approach to the area of business process outsourcing (BPO) services is carried out for the German company SIV.

Keywords: Requirements Engineering, Requirements Engineering Process, Enterprise Modeling, Capability Modeling.

1 Introduction

Organizations are facing the need to adapt their business services according to various situations in which their applications need to be used. To this end an ongoing EU FP7 project “Capability as a Service in digital enterprises” (CaaS) has been conceived [1]. The ethos of the project is support of the capture and analysis of changing business context in design of information systems (IS) using the capability notion.

Capability as a concept originates from competence-based management and military frameworks, offering a complement to traditional Enterprise Modeling (EM) approaches by representing organizational knowledge from a result-based perspective.

In the specification and design of services using business planning as the baseline, capability is seen as a fundamental abstraction to describe what a core business does

[2] and, in particular, *as an ability and capacity for a company to deliver value, either to customers or shareholders, right beneath the business strategy* [3,4].

The key rationale behind the CaaS initiative of developing a capability driven approach to development is to make IS designs more accessible to business stakeholders by enabling them to use the capability notion to describe their business needs more efficiently. The prevailing Model Driven Development (MDD) paradigm for IS development mostly relies on the models defined on a relatively low abstraction level. In contrast, EM captures organizational knowledge and provides the necessary motivation and input for designing IS. Our intention is to enable a holistic approach to model-oriented IS development starting from EM supporting both – the business and technological perspective.

As we have envisioned it in [5] and [6], Capability Driven Development (CDD) requires a number of concepts to be specified, such as business goals, processes, resources, Key Performance Indicators (KPIs), as well the parameters describing business contexts for different capabilities. It is therefore important to define Requirements Engineering (RE) for capability, to ensure a proper specification of the models in consideration and thus facilitate successful development of IS that are able to run and switch between changing operating contexts.

The objective of this paper is to analyze relevant theories and best practices in RE Process and to propose an approach to elicit, analyze, specify and manage the requirements for capabilities and their patterns within the life-cycle of an entire development methodology. A requirement in this context refers to a documented business functional need that the subsequent capability design must be able to support through executable processes and services.

To ensure a cross-industry applicability of the CDD, we have followed action research methodology, by introducing requirements engineering process for capability modeling in company SIV.AG [7] – an independent software vendor in the utility industry.

The rest of the paper is organized as follows. Section 2 briefly describes the theory of the RE process, as well as the Capability Meta-model and the current version of Capability Development Life Cycle. Section 3 defines the RE process for capability and capability patterns, which is illustrated in section 4 with a case of the company SIV. Section 5 presents a brief discussion, conclusions and future work.

2 Theoretical Foundations and Related Work

In this section brief overviews of the topics and the results related to the research of this paper are presented.

2.1 Requirements and RE Process

In the business analysis community, a requirement is seen as a condition describing the current or a future state of any aspect of an enterprise. A basic objective is to ensure that requirements are visible to and understood by all stakeholders [8]. As IS have become the norm for supporting functionalities of enterprises, many of initial

business requirements once refined to lower level of details become system-related, and eventually end-up as software solutions.

From a development life-cycle perspective, RE is considered critical to avoid wrong, incomplete, or ambiguous requirements which will be as such delivered to a next development phase. Hence, it is a common practice to use a process to steer successful management of requirements. Although the activities of the RE process in literature differ from practice, the following are widely accepted as the major [9, 10]:

- *Requirements Elicitation*: in this activity where stakeholders and their requirements are identified using different elicitation techniques (e.g., interviews, focus groups, documentation, etc.).
- *Requirements Analysis*: the elicited requirements are analyzed individually for feasibility, conflicts, ambiguity, redundancy, priority, etc., and they are negotiated with stakeholders for acceptance.
- *Requirements Validation*: the activity concerns checking that the documented requirements specification is understandable, consistent, complete, and meets stakeholder needs.
- *Requirements Management*: this activity is performed to record and track changes to requirements at any time of development.
- *Requirements Documentation*: it is a supportive activity of elicitation, analysis, validation and management, where each requirement is modeled and represented in the way that is understood by relevant stakeholders.

In complex IS development the above activities of the RE process may interweave, and they are performed incrementally and iteratively, i.e. during each iteration more details are elaborated [10].

The traditional way of RE process, where each of the outlined activities have been often performed sequentially and documented in details, has been challenged by the needs of rapidly changing business environments. Consequently, agile methods for system development have emerged [11]. They have set a focus on working software over comprehensive documentation, interactions over processes, responsiveness to change, etc. The methods hence include practices such as short iterations, frequent releases, simple design, minimal documentation in the form of user stories or meeting minutes, implementing requirements as new evolve or existing change, and test cases [12]. In contrast to a few well-established methodologies for agile software development such as XP and Scrum, effective and agile ways for RE are less conclusive than traditional, in terms of theories and practices [13]

2.2 Capability Driven Development

The capability meta-model (CMM) presented in Figure 1 is developed on the basis of industrial requirements and related research on capabilities. It provides the theoretical and methodological foundation for the CDD [5, 6]. The meta-model has three main sections: a) *Enterprise modeling* representing the organizational designs with the *Goals, KPIs, Processes* (with concretizations as *Process Variants*) and *Resources*.

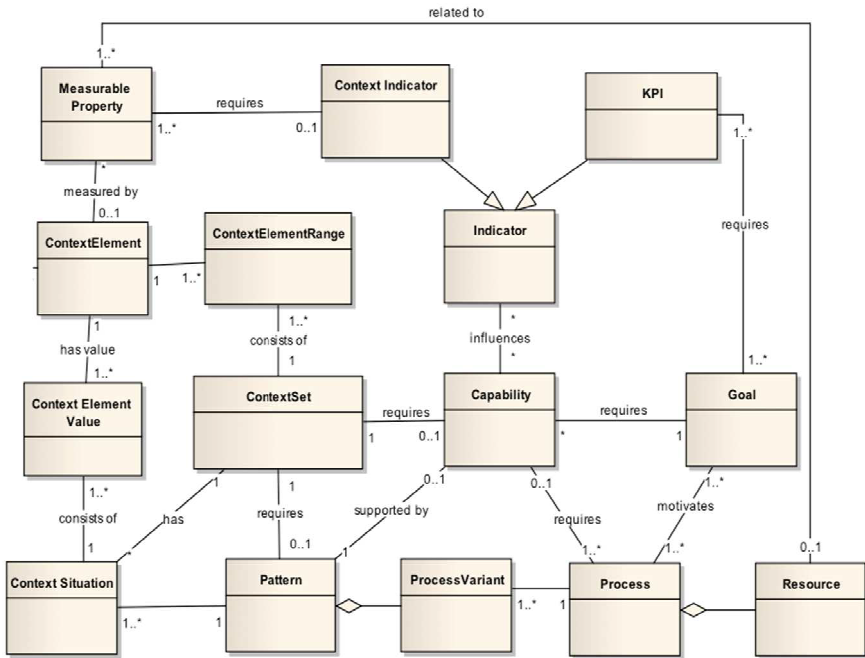


Fig. 1. A Meta-model supporting Capability Driven Development

b) *Context modeling* represented with *Situational Context* instantiating a set of elements under which the solutions should be applied including *Context Indicators* for measuring the context properties (*Measuring Property*); and c) *Patterns* for delivering *Capability* by reusable solutions for reaching business goals under different situational contexts. The context defined for the capability should match the context in which the pattern is applicable. The presented meta-model is a foundation for developing capability-oriented software applications driven by *capability development life cycle process* (Figure 2).

The process starts with EM for capturing business change and required capability, for instance, in order to re-configure existing or to create new goals and business processes to, combined with captured business contexts, elicit required capabilities and patterns. This is followed by development and deployment of the capability delivery application requiring composition and integration of the technologies supporting defined models. During the execution of the application the changes of context are monitored, and run-time adjustments algorithms are used to calculate if the changes are become such to require another capability pattern. Monitoring of defined KPIs facilitate capability refinement and pattern updating.

In the next section we will elaborate the beginning part of the CDD life cycle utilizing the principles of RE to elaborate how required capabilities and corresponding patterns can be elicited, represented, and managed with incoming business changes.

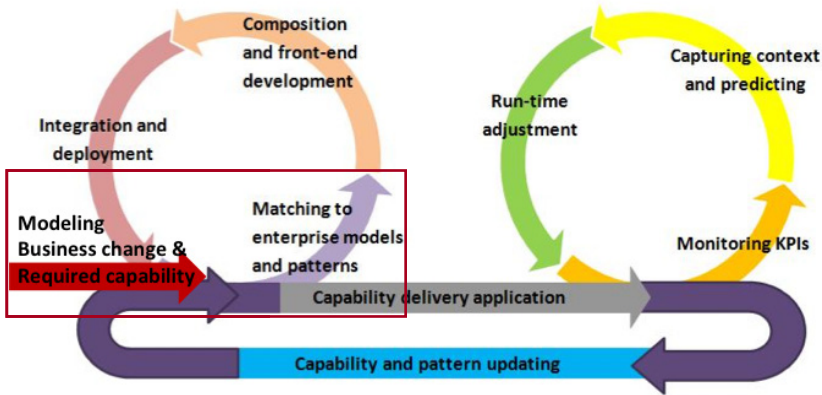


Fig. 2. Capability Driven Development, the life cycle process

3 A Capability-Oriented Requirements Engineering Process

In our view, capability-oriented RE should be facilitated through an efficient process for discovering and defining capabilities of an enterprise acting in varying business contexts. Following the traditional as well as recent views on the RE process and the presented aim for development of capability driven applications, we have elicited a number of objectives for the RE process and the means to achieve them (Table 1). The rest of this section will present in detail our views on the activities of an RE process for capability specification.

Table 1. The main objectives of the RE process for capability specification

Objective	Means
Support specification of concept-wide requirements inherited from EM, and of varying business situations; facilitate specification of patterns of capability delivery that are meant to be sufficiently general and reusable in long-term.	Apply the principles and the activities of the RE process to systematically collect and manage the requirements.
Integrate EM and MDD to overcome inconsistencies between the final software application and the requirements for capability.	Use a model-oriented approach and an integrated tool platform for documenting the requirements as well as for application development.
Facilitate rapid application development and efficient support for requirements change.	Apply incremental and iterative RE, as well as agile practices.

3.1 Elicitation

The objective of the elicitation phase is to iteratively draft requirements for new capabilities according to the CMM. The tasks are (1) identify relevant stakeholders and (2) to elicit capabilities by identifying the sub-models from the CMM.

Stakeholders' identification: the knowledge to be elicited according to the Capability Meta-Model involves several types of included sub-models and a number of concepts: Enterprise Models with a focus to Goal- and Business Process- Models with Process Variants and Resources; Context Models for specifying potential context situations of the capability; and Patterns for specifying reusable capability elements. Table 2 shows relevant stakeholder roles that have been identified.

Table 2. Stakeholder roles in RE for capability

Stakeholder	Responsibility
Business analyst	Identify new, or change/improve existing enterprise sub-models, i.e. goals, processes, resources, and KPIs.
Context analyst	Identify the context-sub model.
Requirements engineer	Has the knowledge of CDD
Customer	Has benefits of delivered capabilities
Capability user	Is directly involved in the interactions of a delivered capability

Any of the roles in Table 2 will be further refined in the beginning of the elicitation process to: a) “localize” the roles for a concrete business model – e.g., in a model where goods from a seller are delivered to a customer by an intermediary delivery company, when eliciting goals of the business, both the Business Analyst of the seller and the deliverer may be considered for this stakeholder role; b) “instantiate” the roles – the roles are to be specified for concrete people, and/or organizations.

Capability elicitation: three strategies are proposed to cover different starting perspectives in capability elicitation – *goal-first*, *service-first*, and *context-first*.

Goal-first elicitation strategy is used when the organization has decided to fulfill the goals following an overall business strategy, and/or where the organization’s enterprise model is elaborated, including actors, business concepts and rules, goals, processes, etc.. The starting focus is to consider existing business goals, and if needed, defining new goals. The iteration tasks are the following:

- A goal-pathway is elicited from a top goal and completed to leaf goals.
- KPIs are defined for the goals, and mandatory for the leaf goals.
- For each leaf goal, zero or more capabilities are identified (for example the goal “To optimize case throughput” identifies the capability “Dynamic business service provider support”, see section 4.1). If for a goal it is concluded that there is no a feasible capability, then it is left for a future consideration.

Each identified capability is further elicited in a separate iteration to:

- Identify the process variant models enacting the capability
- Identify all relevant context elements from the context sub-model of CMM, and relate them to the variation points of the processes.

After an elicitation iteration is completed, a next one is initiated with a focus to another possible goal pathway; the process is continued until the goal model is fully analyzed. The stakeholders from Table 2 are engaged according to their responsibilities in the form of focus groups, for each of the tasks. Business Analyst is the key stakeholder role in this strategy.

Service-first elicitation strategy is used when the customer needs are of the highest importance. Capabilities are therefore identified to support these needs, which are often articulated as top business services. The iteration tasks are the following:

- A desired customer need (i.e. business service), supported by a number of organization's business processes is described as a new capability. For example, the members of a municipality may request "marriage registration e-service", which in turn will lead to the elicitation of a candidate capability "marriage registration" further specified by the tasks below.
- The goals of the capability are elicited, where at least one goal must be elicited with a corresponding KPI.
- The process variant models enacting the capability are identified and improved.
- Identify all context elements from the context sub-model of CMM, and relate them to the variation points of the processes.

After an elicitation iteration as above is completed, a next one is initiated with a focus to another customer need/service; the process is continued until all possible services are exhausted. Customer is the key stakeholder role in this strategy.

Context-first elicitation strategy is chosen when the coverage of wide range of the business contexts of the organization is the most important. E.g. the first main outcome: one or more defined capabilities. The iteration tasks are the following:

Initially observed contexts are analyzed and refined to as many as possible context sets. These are then matched with goals to elicit a new capability, and needed processes are (re)designed accordingly. E.g. if a company offers its services in several countries the local legislations (how heavily regulated it is) are relevant contexts.

- A context set is identified and mapped to a new capability. E.g. in the area of business process outsourcing one might consider two capabilities – one for heavily regulated business environments, and one for more loosely regulated.
- The goals of the capability are elicited, where at least one goal must be elicited with a corresponding KPI.
- The process variant models enacting the capability are identified and improved, variation points are set to match the identified context sets.

After an elicitation iteration as above is completed, a next one is initiated with a focus to another context set; the process is continued until all possible business contexts are exhausted. Context Analyst is the key stakeholder role in this strategy.

The overall way of working according to these elicitation strategies is envisioned to be incremental and iterative. In this sense the capability modeling follows the principle of multi-perspective EM approaches where different modeling perspectives, such as goals, processes, and concepts are modeled in dedicated sub-models. The sub-models are not elicited sequentially, i.e. starting with one sub-model, completing it, moving on to the next one, and so on. Modeling in this way would lead to sub-models that are poorly integrated, inconsistent, and some their parts end up as not addressing the actual goal of the project because the focus of the modelers may shift unintentionally. At some point, to rectify this problem, the modelers would need to "stitch" the model together by identifying gaps and introducing inter-model links.

To avoid this from happening goal, context, and process sub-models of CMM are elicited iteratively and incrementally (and hence analyzed and documented similarly), by switching the focus of modeling among different sub-models. I.e. a CMM model is specified reasonably “in balance” – as proposed in the 3 strategies above, i.e. an iteration is driven to elicit a single capability with corresponding sub-models’ parts.

3.2 Analysis

The objective of the requirements analysis is to assess if the elicited capabilities are correct following the criteria outlined in section 2.1. Table 3 shows the criteria for the context in discussion.

Table 3. Main analysis criteria for capability specification

Parameter	Action
Necessity	Is the elicited capability needed in respect to the given goals? The CMM mandates at least one goal for a capability, ensuring a motivation to exist. Another assessments relate to the existence of at least one business context in which the capability will bring a value to the customer, or at least one requested business service that would be supported by the capability.
Feasibility	An elicited capability is through a feasibility study checked by stakeholders to ensure that it can be developed in the context of economic, time, technical and other constraints set for the development project.
Redundancy	When a new capability is about to be elicited, stakeholders need to get the knowledge on the existing capabilities; the supporting mechanism is the single development environment containing the models of existing capabilities, and the repository of all the capability patterns with descriptions.
Consistency	An elicited capability should not be contradictory, ambiguous or in a conflict with other existing capability. The first two criteria can be assessed by examining the consistency of the CMM sub-models of the capability, and their completeness. Possible conflicts need to be examined on different levels – e.g. if they exist in the goal model of the capability, or in the process resources, or if any of the context elements are in conflict.

The accepted capabilities from the analysis can also be negotiated among stakeholders in order to prioritize them, which is a common practice in RE [10]. From an agile perspective, both the analysis and prioritization of capabilities can be done during elicitation – the single and integrated development platform (Table 1) enables these activities to be done concurrently, and especially when the relevant stakeholders are able to continuously or frequently collaborate (Table 2).

For each capability that is successfully analyzed, a candidate capability pattern is assigned (see Figure 1) either by selecting it from a pattern repository or by developing a new pattern from the existing business solutions (e.g. process variants). Patterns require dedicated activities for elicitation, analysis, documentation, validation and management shown by the bottom loop in Figure 2, which we consider outside the scope of this paper.

3.3 Documentation

The intention of CDD is to facilitate a holistic and integrated approach to model-oriented IS development starting from enterprise modeling. Therefore,

- Documentation should be done in a tool environment using modeling languages,
- Natural-language annotations are used for descriptions of models, elements, etc.
- Development environment is open to the use of any modeling language for specifying requirements for goals, processes, context, and patterns.
- Intra- and inter-model links can be defined for traceability purposes.
- Documented requirements can be searched, or navigated.

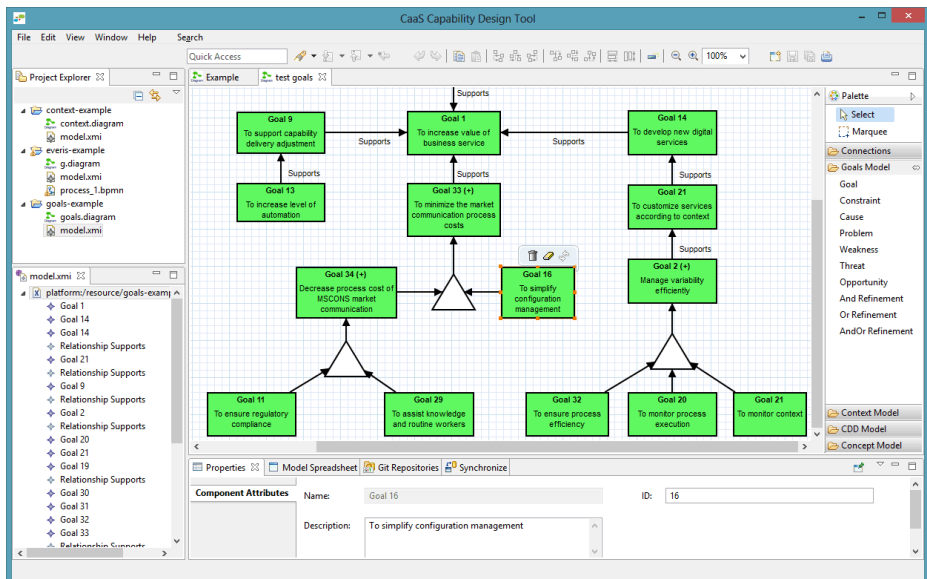


Fig. 3. Documenting Capability according to CMM; goal modeling (center pane), the language (right pane), descriptions (bottom pane), navigation (left pane), and search (top pane)

Requirements documented in the tool are further transformed to development artifacts by augmenting the descriptions of the models with technology-related data – for example, business process can be modeled with BPMN2 where the modeling details can be added in any of the development phases.

3.4 Validation

In contrast to analysis where the assessment is set to the correctness of individual capabilities, the objective of the validation in the traditional view is to examine the quality of the requirements specification *as a whole*, where the major criteria are outlined in section 2.1; considering them, as well as the advices from agile practices for using iterative-type of actions we propose the validation of a capability as following:

- Use review meetings for validation after each or every few iterations, and involve developers in addition to the main RE stakeholders (Table 2). During the meeting, the specified capabilities are demonstrated from the tool, and Q&A on their quality and usefulness are used to obtain feedback and to emphasize problems early. In addition, the demonstration has the objective to increase customer trust and confidence to the current results.
- Use expert reviewing to assess the quality of the current models in terms of, e.g. understandability, completeness, and consistence, as well as the links for traceability among them. The reviewing is organized as a session in conjunction with the main meeting, and where the models are given to individual experts (Table 2) for assessment and proposals for improvements for next iterations.

3.5 Change Management

In contrast to traditional way of managing changes in requirements, where formal requests for change are submitted and a well-documented analysis of the change is performed by specially assigned roles, agile practices recommend to consider changes in requirements simply as new requirements which will be according to priorities elaborated in one of the regular iterations.

Following the agile approach, the management of a business change depicted in Figure 2) is considered as it occurs. Depending on the cause of the change (such as inference of a new context element), the key stakeholder will be the main responsible to assess the importance and priority for the change; thereafter changes in the sub-models of CMM will be elicited, where the traceability of intra- and inter- model links will be used to identify the influenced models and concepts. For example, if for the previously mentioned “marriage registration” capability, a new context element “period” is added because of a heavier load during springs than in winters, consequently a process model within the capability has to change to include needed variation points, as well as needed variants of execution.

4 An Application Case at SIV

The CaaS partner SIV is a Germany-based independent software vendor (ISV) and a business process outsourcing (BPO) provider for the utilities industry. SIV has developed a domain-specific ERP platform kVASy[®] that supports all relevant value-added processes of market players. All BPO services offered to SIV’s customers – mostly grid access providers and balance suppliers – are based on the functionalities of kVASy[®]. SIV’s business goal is to deliver a maximum of business value to its customers by to combining best practice business processes with compliance to the market’s ever changing business rules and regulatory requirements.

4.1 Requirements Engineering for Capabilities in SIV

In SIV’s approach, key capabilities are elicited following the principle of the *Goals first* strategy, i.e. the RE process starts by defining goals for offering BPO services

(Figure 4). The reason for choosing this approach lies in the fact that over the years the company's role in the market has arrived at a mature stage, where the ERP platform kVASy[®] is well established in the industry. Therefore, the challenge of keeping existing and acquiring new customers in a highly competitive utility market is a top priority. Performing an enterprise-level analysis with a clear focus on business goals, and linking them to appropriate BPO capabilities is a way to deliver business value to kVASy[®] users, thereby sustainably strengthening SIV's market position.

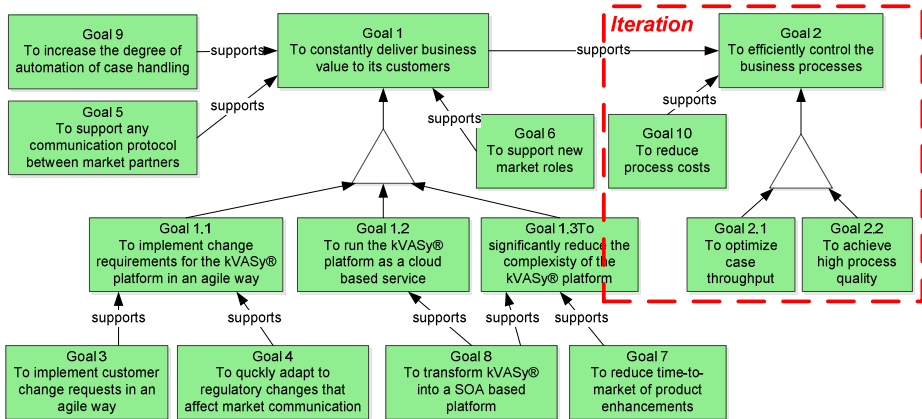


Fig. 4. The goal model of SIV

The goals of the industrial use case are modeled using the 4EM approach. Due to the relationship between SIV and its customers, a distinction was made between customer goals and SIV goals, i.e. customer goals are more *operational* whereas SIV follows *strategic* business goals, since SIV has no operative business. For SIV's customers, however it is of prime importance running their business processes as efficiently as possible, thus keeping process costs low. Therefore, these organizations are predominantly driven by operative factors such as workload, backlog, and the availability of human resources. By contrast, SIV is more oriented to strategic goals, such as to constantly meet the customer's requirements and to exploit opportunities of the market. Thus, there are two aspects in the overall goal model that are distinct, yet not separable. In this respect, the main goal of SIV is *to deliver constant business value to its customers*, thereby supporting the customer's goal *to efficiently control business processes*. Also, goals can be refined into sub-goals, forming a hierarchically organized goal model. To efficiently control business processes, customers aim *to optimize case throughput and to achieve a high process quality*. Both can be considered as subordinate goals in the use case.

Elicitation: Following the guidelines of the *Goal-first* approach, the objective for business analysts of the utility companies and of SIV was to iteratively analyze the goals to identify capabilities. In an iteration, a single goal-pathway is elicited, such as for example Goal 2 supported by Goal 1, further requiring the achievement of Goal 2.1 and Goal 2.2. According to the CaaS approach, achievement of business goals

requires specific capabilities. In the following we exemplify the linkage of capabilities to goals, using Goal 2.1 “*To optimize case throughput*”. Enabling customers to optimize their case throughput, requires SIV to deliver capabilities to route their processes in accordance with the workload. For the industrial case at hand, “*Dynamic business service provider (BSP) support*” is identified as a key capability that supports this goal. This connection is illustrated in Figure 5.

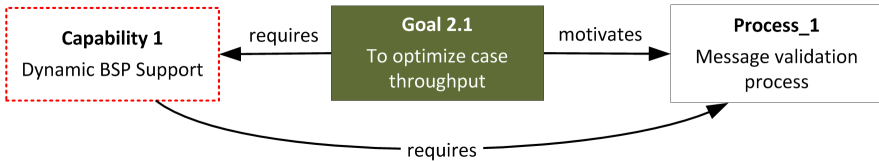


Fig. 5. Specification of a capability from a goal, supported by a process

The capability comprises the business process triggered when partner A sends a message about energy consumption to partner B (Figure 6). The process involves a validation of the message, and a processing step. The purpose of validation is to ascertain the message is syntactically and semantically correct with respect to an underlying informal data model. Upon validating the message, exceptions can occur, triggering in turn another process to remedy this exception. If a faulty message could be remedied, the processing step may be successfully re-executed. If this is not the case, then the process has to be aborted.

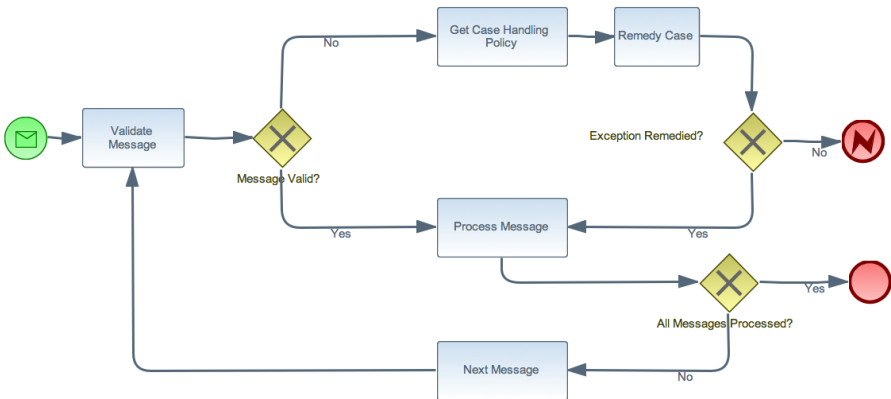


Fig. 6. “Message Validation” business process

To implement the “*Dynamic BSP support*” capability, which is required by the “*To optimize case throughput*” goal of the customer, SIV must offer instruments enabling customers to automatically route individual process instances to an external business service provider in order to achieve the desired case throughput. Since such routing decisions depend on many operational factors such as the customer’s workload, the current backlog size and exception type, the associated capability requires evaluation

of complex situations, which are represented by context models. Depending on the concrete context situation, the task “Remedy case” can be dynamically routed to the external business service provider (BSP), or left with the customer. This accordingly leads to the elicitation of two process variants: either the customer handles the faulty case on his own, or the case is treated by the BSP’s highly trained experts (Figure 7).

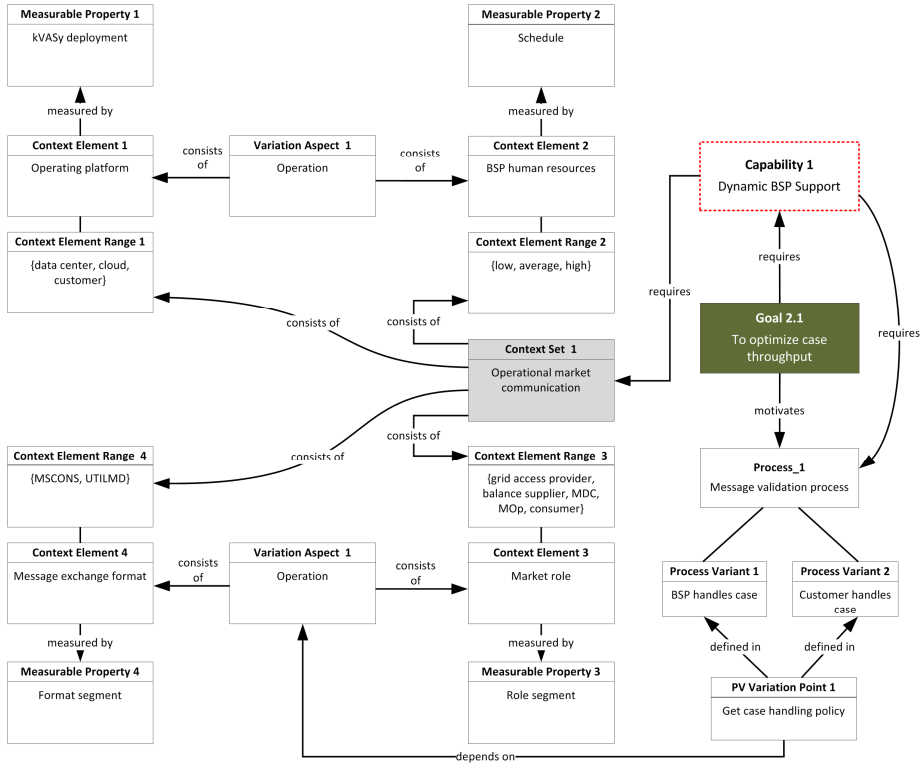


Fig. 7. Context model for the *Dynamic BSP Support* capability

As a result, the process variation point is located in “*Get case handling policy*” process. The decision which variant must be executed depends on the business context. Acquisition of such situation during runtime has to take place for each individual case to feed the decision logic component with sufficient input data to enable a correct routing scenario for the case in question. Three main business drivers are identified, which cause variations and thus provide stimulus for changes. The first business driver is the contractual aspect, which specifies parameters such as backlog threshold as well as the process variant to be implemented regarding the backlog size, such as “if the backlog size exceeds the agreed threshold, then the case is routed to customer”. The second business driver, payload aspect, includes information of the service call such as the market role, the faulty message and the exception type etc. The last driver, the operational aspect, is related to both SIV Services personnel deployment plan and the kVASy-operating environment. These are captured as variation aspects, which are

further elaborated to identify context elements. A capability offered by an enterprise requires an application context and this is captured as a context set. “Operational market” communication context set is required by the capability and comprises of the possible ranges of four context elements such as operating platform, BSP human resources, market role and message exchange format.

With the completion of context elicitation, the capability becomes specified. As illustrated in Figure 5 and motivated above, the elicited capability is needed in respect to the business goals. Delivering such capability is context-dependent, i.e. dynamic BSP support capability brings value to the customer in a specific situation that is captured with Operational market communication context set.

Analysis. Although we have not conducted a feasibility study, the stakeholders perceived the development of such capability in given project conditions in a positive manner and collaborated actively on the development activities both in workshops and in distributed project teams. It is still early to make statements about the consistency of the capability models since at the time being we have developed only one capability. We are expecting in the future more observations on this.

Documentation has been successfully done in the integrated modeling environment (section 3), with the outcomes represented in the figures above. The tool uses natural language annotations for model descriptions and allows for linking different parts of the enterprise models such as connecting a business process model to a goal and to a capability. It was also possible to apply a notation to represent the context model, which have been created and developed following the principles of Moody [14].

Validation has been performed in an extended expert group, in the form of reviews as explained in section 3. Change management has not yet been practiced, because there was not any business change occurring during the period of requirements’ specification.

5 Conclusions and Future Work

We have proposed an approach for Requirements Engineering for Capability Driven Development, which specifies the notion of capability and related concepts and integrates organizational analysis with IS development taking into account changes in the organization’s business context.

The main objective of the process is to facilitate the specification for capability requirements in an integrated way following the multi-perspective views defined in the CMM to enable efficient application development. Therefore the proposed process consists of guidelines for systematic management of requirements combined with agile principles of working and model based documentation of requirements. The requirements are proposed to be elicited starting either from business goals, services, or from relevant business contexts. In any of the three strategies, both the functionality and the quality aspects of capability are captured, where the first are dictated by the CM, and latter by the setting of the goals and KPIs.

The proposed process and its guidelines were tested in a real case concerning business process outsourcing (BPO) services carried out within the company SIV. The results have confirmed the functionality of the process, where both its systematic and

agile aspects have been practiced. The model-based documentation of requirements for capabilities using the presented tool has been widely accepted by the stakeholders. Efficiency in communication, rapid specification and analysis, and effective validation principles, were the key aspects to accept the process, even over traditional, text-oriented and more sequential way of doing RE

For the future work, we plan to define a process for managing the requirements for capability patterns. A composite structure of the patterns requires specific guidelines and a language for their specification, storing, search, combining, etc. Another challenge to address concerns the elicitation of capabilities in the situations where a supporting base of software services has been previously developed, and capability requirements need therefore to consider these services as existing “IT capabilities”.

References

1. EU FP7 CaaS Project. Capability as a service for digital enterprises., <http://caas-project.eu/>
2. Ulrich, W., Rosen, M.: The business capability map: Building a foundation for business/it alignment. Cutter Consortium for Business and Enterprise Architecture (2012)
3. OPENGROUP TOGAF - enterprise architecture methodology, version 9.1 (2012), <http://www.opengroup.org/togaf/>
4. OPENGROUP Archimate - modelling language for enterprise architecture, v2.0 (2012), <https://www2.opengroup.org/ogsys/catalog/c118>
5. Stirna, J., Grabis, J., Henkel, M., Zdravkovic, J.: Capability Driven Development – An Approach to Support Evolving Organizations. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) PoEM 2012. LNBIIP, vol. 134, pp. 117–131. Springer, Heidelberg (2012)
6. Zdravkovic, J., Stirna, J., Henkel, M., Grabis, J.: Modeling Business Capabilities and Context Dependent Delivery by Cloud Services. In: Salinesi, C., Norrie, M.C., Pastor, Ó. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 369–383. Springer, Heidelberg (2013)
7. SIV. AG <http://www.siv.de/> (last accessed July 16, 2014)
8. International Institute of Business Analysis (IBA). A Guide to the Business Analysis Body of Knowledge® (BABOK® Guide) Version 2.0 (2009), <http://www.theiiba.org>
9. Dorfman, M., Thayer, R.H.: Software Requirements Engineering. IEEE Computer Society Press, Los Alamitos (1997)
10. Kotonya, G., Sommerville, I.: Requirements Engineering: Process and Techniques. John Wiley and Sons, New York (2002)
11. Fowler, M., Highsmith, J.: The agile manifesto. *Software Development* 9(8), 28–35 (2001)
12. Ramesh, B., Cao, L., Baskerville, R.: Agile requirements engineering practices and challenges: an empirical study. *Information Systems Journal* 20(5), 449–480 (2007)
13. Leffingwell, D.: Agile software requirements: Lean Requirements Practices for Teams, Programs, and Enterprise. Addison-Wesley (2011) ISBN-10: 0321635841
14. Moody, D.L.: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. *IEEE Transactions on Software Engineering* 35(6), 756–779 (2009)