RESEARCH PAPER

Capability Driven Development: An Approach to Designing Digital Enterprises

Solvita Bērziša · George Bravos · Tania Cardona Gonzalez · Ulrich Czubayko · Sergio España · Jānis Grabis · Martin Henkel · Lauma Jokste · Jânis Kampars · Hasan Koç · Jan-Christian Kuhr · Carlos Llorca · Pericles Loucopoulos · Raul Juanes Pascual · Oscar Pastor · Kurt Sandkuhl · Hrvoje Simic · Janis Stirna · Francisco Giromé Valverde · Jelena Zdravkovic

Abstract The need for organizations to operate in changing environments is addressed by proposing an approach that integrates organizational development with information system (IS) development taking into account changes in the application context of the solution. This is referred to as Capability Driven Development (CDD). A meta-model representing business and IS designs consisting of goals, key performance indicators, capabilities, context and capability delivery patterns, is being proposed. The use of the meta-model is validated in three industrial case studies as part of an ongoing collaboration project, whereas one case is presented in the paper. Issues related to the use of the CDD approach, namely, CDD methodology and tool support are also discussed.

Authors are listed alphabetically to reflect equal contributions.

Dr. S. Bērziša · Prof. Dr. J. Grabis · L. Jokste · Dr. J. Kampars
Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia

Dr. G. Bravos · Prof. Dr. P. Loucopoulos
The University of Manchester, Oxford Rd, Manchester M13 9PL, Manchester, UK

T. C. Gonzalez · R. J. Pascual
Everis, Av. Cortes Valencianas, 39, Valencia, Spain

Dr. U. Czubayko · Dr. J.-C. Kuhr
SIV AG, Konrad-Zuse-Str. 1, 18184 Roggentin, Germany

Dr. S. España · Prof. Dr. O. Pastor · Dr. F. G. Valverde
PROS Center, Universitat Politècnica de València, Camino de Vera S/N, València, Spain

Dr. M. Henkel (✉) · Assoc. Prof. J. Stirna · Assoc. Prof. J. Zdravkovic
Department of Computer and Systems Sciences, Stockholm University, Forum 100, 16440 Kista, Sweden
e-mail: martinh@dsv.su.se

H. Koç · Prof. Dr. K. Sandkuhl
Chair Business Information Systems, University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, Germany

C. Llorca
Fresh T Limited, Keckwick Lane, Daresbury, UK

H. Simic
Croz d.o.o, Lastovska 23, 10000 Zagreb, Croatia

1 Introduction

To improve alignment between business and information technology, information system (IS) developers continuously strive to increase the level of abstraction of development artifacts. A key focus area is making the IS designs more accessible to business stakeholders to articulate their business needs more efficiently. These developments include object-orientation, component based development, business process modeling, enterprise modeling (EM) and software services design. These techniques are mainly aimed at capturing relatively stable, core properties of business problems and on representing functional aspects of the IS (Wesenberg 2011). However, the prevalence and volatility of the Internet shifts the problem solving focus to capturing instantaneous business opportunities (Deloitte 2009). Furthermore, the context of use for modern IS is not always predictable at the time of design; instead an IS should have the capability to support different contexts which implies that, we should consider the context

of use and under which circumstances the IS, in congruence with the business system, can provide the needed business *capability*. Hence, an IS capability is determined not only during the design-time but also at run-time when its ability to handle changes in different contexts is put to test. As an example, airport operations use different patterns to cope with different levels of passenger flow at times of different events, e.g., when many passengers are stranded at once which leads to surges in demand for various services. Tackling the issues of IS supporting different contexts is especially relevant for modern digital enterprises. Digital enterprises are enterprises possessing digital resources and providing an important share of its services digitally by customizing the essential services to meet requirements of customers facing specific operating circumstances. Such enterprises continuously monitor changes in the business context in order to identify opportunities for capitalizing on these changes.

A capability-driven approach to development should be able to alleviate all such issues and to produce solutions that fit the actual application context.

From the business perspective, we define a capability as the ability and capacity that enable an enterprise to achieve a business goal in a certain context. Ability refers to the level of available competence, where competence is understood as talent intelligence and disposition, of a subject or enterprise to accomplish a goal; capacity means availability of resources, e.g., money, time, personnel, tools. Note here that capacity, being the amount of available resources, is an integral part of a capability. IS applications (and their execution environments) can be an important part of capabilities. This means that it is important to tailor these applications with regard to functionality, usability, reliability, and other factors required by users operating in varying contexts. This puts pressure on IS development and delivery methods. The IS development industry has responded by elaborating Model Driven Development (MDD) methods and by adopting standardized design and delivery approaches such as service-oriented architecture and cloud computing. However, there are a number of major challenges when it comes to making use of MDD to address business capabilities:

- *The gap between business requirements and current MDD techniques*. MDD approaches and tools still operate with artifacts defined on a relatively low abstraction level.
- *Inability to model execution contexts*. In complex and dynamically changing business environments, modeling just a service providing business functionality in a limited context of execution is not sufficient.
- *High cost for developing IS that work in different business contexts*. Developers, especially SMEs, have difficulties to market their software globally because of the effort it takes to adhere to localization requirements and constraints in the business context of where the software will be used.
- *Limited support for modeling changes in non-functional requirements*. Model-driven approaches focus on functional aspects at a given time point, rather than representing evolution of both functional and non-functional IS requirements over time.
- *Limited support for "plasticity" in applications*. The current context-aware and front-end adaptation systems focus mainly on technical aspects (e.g., location awareness and using different devices) rather than on business context awareness.
- *Limited platform usage*. Limited modeling support for defining the ability of the IS to make use of new platforms, such as cloud computing platforms because it is a technology driven phenomenon, and there is little guidance for development of cloud based business applications.

We propose to support the development of capabilities by using EM techniques as a starting point of the development process and to use model-based patterns to describe how the software application can adhere to changes in the execution context. Our vision is to apply enterprise models representing enterprise capabilities to create executable software with built-in contextualization thus leading to *Capability Driven Development (CDD)*.

The objective of this paper is to present the CDD meta-model, to discuss its feasibility by using an example case, and to outline a number of open development issues related to practical adoption of the CDD approach.

The research approach taken in this paper is conceptual and argumentative. Concepts used in EM, context representation and service specification are combined together to establish the CDD meta-model. Validation of the meta-model is performed using the cases of companies in the fields of e-government, compliance, and business process outsourcing. Application of the meta-model is outlined by analyzing its role in development of capability delivery applications. The CDD methodology is proposed following the principles of agile and iterative IS development methodologies. The work presented in this paper is a continuation of our work presented in (Stirna et al. 2012) and (Zdravkovic et al. 2013). Compared to these publications, in this paper we present a refined CDD meta-model, a new case study where the CDD approach has been applied, and a validation of the approach.

The remainder of the paper is organized as follows. Section 2 sets the scene for the CDD approach in terms of related work and the requirements for CDD. In Sect. 3 requirements for CDD are discussed. Section 4 presents the CDD meta-model. It is applied to a validation case in

Sect. 5. Section 6 discusses aspects of development methodology need for the CDD approach. The paper concludes in Sect. 6 with a number of reflective remarks.

2 Background

2.1 Related Work

In the strategic management discipline, a company's resources and capabilities are, for a long time, seen as the primary source of profitability and competitive advantage. Barney has united them into what has become known as the resource-based view of the company (Barney 1991). Accordingly, Michael Porter's value chain identifies top-level activities with the capabilities needed to accomplish them (Porter 1985). In Strategy Maps and Balanced Scorecards, Kaplan and Norton also analyze capabilities through the company's perspectives, e.g., financial, customers', and other (Kaplan and Norton 2004).

In research within Business-IT alignment, there have been attempts to consider resources and capabilities as the core components in enterprise models, more specifically, in business value models (Osterwalder and Pigneur 2003; De Kinderen et al. 2009). However, in none of these works, capabilities are linked to IS models in a structured and systematic way. In the SOA reference architecture (OASIS 2011) a capability is described as a business functionality that, through a service, delivers a well-defined user need. However, in the specification, little attention is given to the modeling of capabilities. In Web Service research, capability is considered purely on the technical level, through service level agreements and policy specifications (Papazoglou and Yang 2002).

In order to reduce development time, to improve software quality, and to increase development flexibility, MDD has established itself as a promising IS development approach. However, Asadi and Ramsin (Asadi and Ramsin 2008) show that the Model Driven Architecture (Kleppe et al. 2013), a de-facto standard for MDD, and its associated methodologies mainly assume requirements as given a priori. Loniewski et al. (2010) and (Yue et al. 2011) indicate that MDA starts with system analysis' models. They also survey various methods for integrating requirements into an overall model-driven framework, but do not address the issue of requirements origination. There is a limited evidence of MDA providing the promised benefits (Mohagheghi and Dehlen 2008). Complexity of tools, their methodological weaknesses, and too low abstraction level of development artifacts are among the main areas of improvement for MDD tools (Henkel and Stirna 2010).

EM has been used for business development and early requirements elicitation for many years (Nilsson et al.

1999). However, a smooth, nearly automated, transition to IS development has not been achieved due to immaturity of the existing approaches and lack of tools. Enterprise-wide models are also found in various Enterprise Architecture development approaches, for example, where the enterprise architecture of ArchiMate (ArchiMate 2013) is extended with an intentional aspect capturing the goals and requirements for creating an enterprise system. A comparable solution is developed in (Pastor and Giachetti 2010), where a generic process is presented for linking i* and the OO-Method as two representatives of Goal-Oriented Requirements Engineering (GORE) and MDD, respectively. Furthermore, a systematic transition from goals to business process models (Ruiz et al. 2014), and from business process models to UML-based specifications (González et al. 2011) is being developed, but its industrial application and adoption is yet to be demonstrated.

Methods for capturing context in applications and services have achieved high level of maturity and they provide a basis for application of context information in IS development and execution. Vale and Hammoudi (Vale and Hammoudi 2009) describe MDD for context-aware applications, where the context model is bound to a business model, encompassing information about user's location, time, profile, etc. Context awareness has been extensively explored for Web Services, both methods and architectures, as reported in (Sheng et al. 2010). It is also studied in relation to workflow adaptation (Smanchat et al. 2008); Hervas et al. (2010) have suggested a formal context model, compounded by ontologies describing users, devices, environment and services.

In summary, there are a number or contributions to addressing the problem of adjusting IS depending on the context, however the concept of business capability is not explicitly addressed.

2.2 Requirements for Capability Driven Development

Currently the business situation in which an IS will be used is predetermined at design time. At run-time, only adaptations that are within the scope of the planned situation can usually be made. But in many cases we need rapid response to changes in the business context and development of new capabilities, which also requires run-time configuration and adjustment of the IS. In this respect a meta-model for capability modeling, linking business designs with application contexts and IS components is needed.

Designing capabilities is a task that combines both business and IS knowledge. A review of existing practice (Jarke et al. 2011) shows that implementation and its requirements specification are now closely intertwined. Hence both domains need to be integrated in such a way

that allows establishing IS support for the business capabilities.

Current EM and business development approaches have grown from the principle that a single business model is owned by a single company. In spite of distributed value chains and virtual organizations this way of designing organizations and their IS still prevails. The CDD approach aims to support co-development and co-existence of several business models by providing "connection points" between business models based on goals and business capabilities.

Most of the current implementations of MDD approaches do not support development of more advanced features e.g., complex calculations, advanced user interfaces, scalability of the application in the cloud. CDD should contribute to the state of art by supporting the modeling of the application execution context; this includes modeling the ability to switch service providers and platforms.

3 Foundations for Capability Driven Development

The CDD meta-model presented in this section provides the methodological foundation for the CDD. In this paper, the term CDD meta-model is used to denote the high-level language defined for expressing the models as used when applying the CDD. The meta-model thus contains the central concepts that are used, and their relationships; therefore it specifies the abstract syntax and the semantics for modeling during CDD. The meta-model is developed on the basis of industrial requirements and related research

on capabilities. The CDD meta-model is shown in Fig. 1. The meta-model has three main parts:

- *Enterprise and capability modeling* for developing organizational designs that can be configured according to the context dependent capabilities in which they will be used. This part of the meta-model captures a set of generic solutions applicable in many different business situations.
- *Capability delivery context modeling* of situations under which the solutions should be applied including indicators for measuring the context properties.
- *Capability delivery patterns* representing reusable solutions for reaching business goals under different contexts. The context defined for the capability should match the context in which the pattern is applicable in.

3.1 Enterprise and Capability Modeling

This part covers modeling of business *goals*, *key performanceindicators* (*KPI*), and business *processes* needed to accomplish the goals. *KPI*s are performance measurements used for monitoring *goal* fulfillment. We also specify *resources* required to perform *processes*. The associations between these modeling components are based on the meta-model of the EM approach EKD (Bubenko et al. 2001). The concept of *capability* extends EKD towards being suitable for CDD.

Capability is the ability and capacity that enable an enterprise to achieve a business goal in a certain context. Capability is the core element that describes the part of the
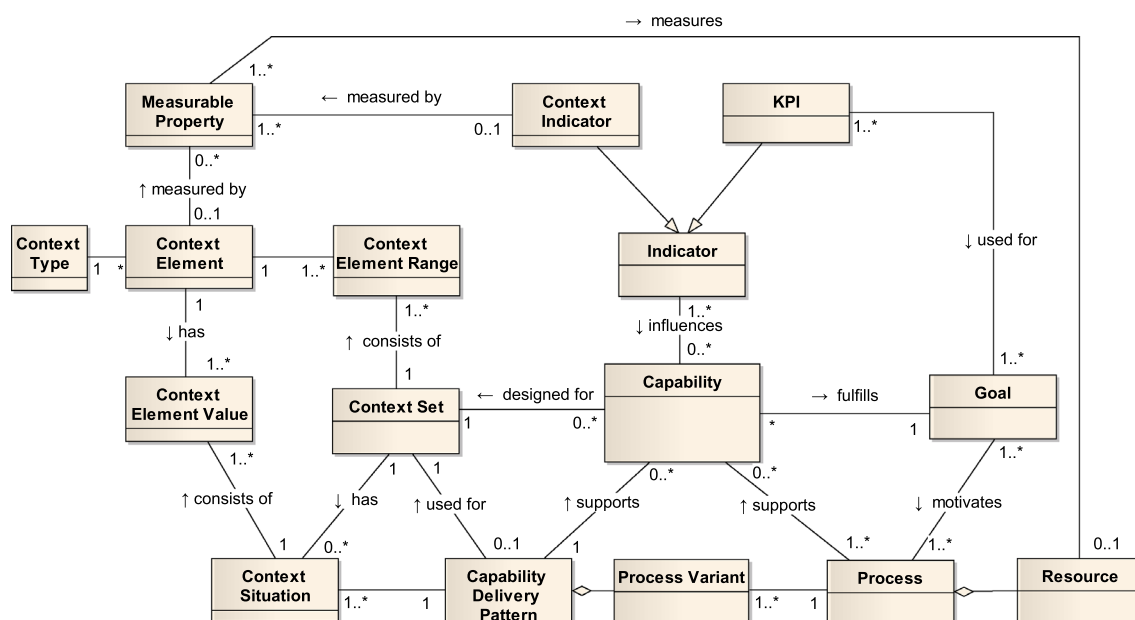

Fig. 1 The CDD meta-model

business that will be designed and delivered by the CDD approach. *Capability* formulates the requirements for the ability of accomplishing a business *goal*, realized by applying a solution described by a capability delivery *pattern*.

Each *capability* requires or is motivated by one business *goal*. In principle business goals can be seen as internal means for designing and managing the organization and capabilities as offerings to external customers. A *capability* requires or is supported by specific business *processes*, each process utilizing a set of *resources*. The distinguishing characteristic of a *capability* is that it is designed to be provided in a specific *context*. The desired *goal* fulfillment levels can be defined by using a set of *indicators* in the form of *KPI*s.

3.2 Context Modeling

The context modeling part consists of *context elements* to describe the context constituents, as well as *indicators* in the form of *measurable properties* that can be used to monitor a specific *context situation*. The context is any information that can be used to characterize the situation (Dey 2001), in which the capability can be provided. In the CDD meta-model the *context set* denotes a set of circumstances, such as geographical location, platforms and devices used, as well as business conditions and environment. These circumstances are described by the use of *context elements*, categorized by different *context types*. Each *context element*, such as geographical location, has a range of valid values as identified by its *context element range*. The purpose of *context element range* is to represent the actual ranges of value of relevant *context elements* for a specific *context set*. The *context situation* represents the current context status.

Each capability delivery *pattern* requires a specific *context set* as to be possible to apply. The *context indicators* represent context measurements, which are of importance for the capability delivery.

The *context indicators* are used to monitor whether the design for capability delivery is still valid for the current *context situation*. If it is not valid, then capability delivery should be dynamically adjusted (i.e., changing delivery process, reassigning resources, etc). Technically, the context information is captured using a context platform in a standardized format, e.g., XCoA (Gomes et al. 2010).

3.3 Capability Delivery Pattern

In the CDD approach we amalgamate the principle of reuse and execution of patterns as good designs (c.f., for instance, Gamma et al. 1995) with the principle of sharing best practices in the form of organizational patterns. Hence,

capability delivery *patterns* are reusable solutions for reaching business *goals* under specific *context situation*. The context defined for the *capability* (by the context set) should match the context in which the *pattern* is applicable. Patterns will represent reusable solutions in terms of business *process variants*, including resources, roles and supporting IT components (e.g., code fragments, web service definitions) for delivering a specific type of capability in a given context.

Each pattern describes how a certain capability is to be delivered within a certain context and what processes and resources are needed. In order to provide a fit between required resources and available resources, *KPI*s for monitoring capability delivery quality are defined in accordance with organization's *goals*. *KPI*s measure whether currently available resources are sufficient in the current context.

4 Validation of the CDD Meta-Model

The goal of the meta-model validation is to test the suitability of the CDD meta-model in a realistic case. Correctness and expressiveness of the meta-model were the fundamental aspects to evaluate. The meta-model has been validated by modeling three companies in different fields: (1) e-government, (2) compliance, and (3) business process outsourcing. More specifically, we developed capability models for the following industrial use cases:

1. At Everis (Spain) for service promotion capability, marriage registration capability, SOA platform capability
2. Fresh T Limited (UK) for compliance capability
3. SIV AG (Germany) for standard business processes execution capability.

The capability models were developed using the Enterprise Architect tool (Sparx Systems 2013). The model elements are represented using UML 2.0 (OMG 2011a), while process models are elaborated using BPMN 2.0 (OMG 2011b). Due to space limitations this section only presents one case; the complete set of models is available in (Berziša et al. 2014).

4.1 Application Case at Everis – Service Promotion Capability

The purpose of the Service Promotion (SP) is to encourage the use of a service on municipality's e-government web portal. In particular, this is done in situations where a service is highly used in a municipality with a similar profile (number of citizens, location near the sea or inland, etc.). Each municipality using e-services has a service catalogue on its home page and several services can be

promoted by highlighting them on the main page. Each municipality can follow different approaches on how to implement the highlighting. Some home pages have an ability to run automatic service highlighting (showing service in the main page), while some municipalities' home pages do not. The capability model of the SP case is shown in Fig. 2, using the UML object diagram notation.

The main goal of the SP capability is to promote the usage of the online services. It is supported by several sub-goals, see Fig. 3.

Each of the goals has one or more KPIs associated with it. A list of KPIs is given in Table 1.

The context representation is elaborated using a composite context model here summarized in Table 2. It includes the list of context elements and their defined types. The measurable property concept defines the measurements for the specific context elements. The context element value and the context situation concepts are not shown here because they are instantiated only during the capability delivery phase. The context element has zero or more measurable properties because in the design time ways to measure some of the context elements might be unknown.

The SP process has two main process variants in respect to how the services with a high usage in one municipality can be promoted in similar municipalities: if the municipality's home page has automatic service highlighting ability, then *service highlighted procedure* is executed. Depending on different context data, service highlighted procedure can be run once every 24 h, or once every 72 h. If automatic highlighting is not possible, or another municipality with similar profile does not have particular service, then an email is sent to the municipality, or to the project management office (PMO) about service

promotion. Such factors or business drivers influencing the execution of the relevant process variant are summarized in Table 2, where context elements and their measurable properties are specified. Service usage (context) is monitored during the run time according to the following principles:

- If service usage is high, then service can be highlighted in similar municipalities (similar size and profile);
- If citizens' feedback about service usage in social networks is positive, then service can be highlighted in similar municipalities (similar size and profile);
- If a municipality's portal does not have an ability to automatically highlight the service, an e-mail to the municipality is sent about the high usage of particular services.
- If a municipality with similar profile does not have a particular service which has a high usage in other municipality, then the information about this service is sent to PMO about service dissemination.

The process model with process variants and capability delivery patterns is shown in Fig. 4. In the capability model, process variants are presented as a separate concept from the main process, but in the process model they are included as the alternative paths to the main scenario. After the starting event of the process, an evaluation is done by an expression taking the context elements as inputs. In this case, the context elements used are the municipality size, usage of the service in other municipalities and feedback in social networks. The conditional expression uses these context elements to determine the need for running the service promotion process.

For the presented case the following modeling observations can be made:

Fig. 2 Service Promotion (*SP*) Capability model
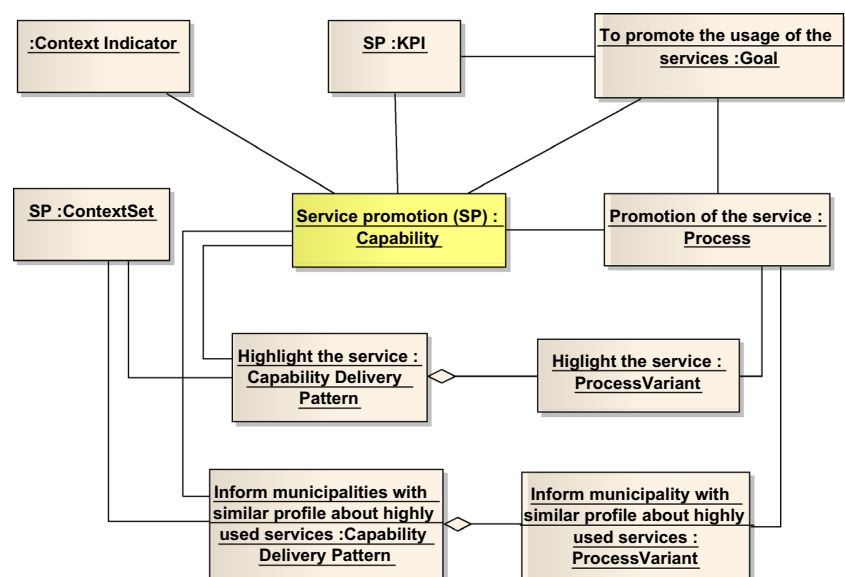
Fig. 3 SP Goals model
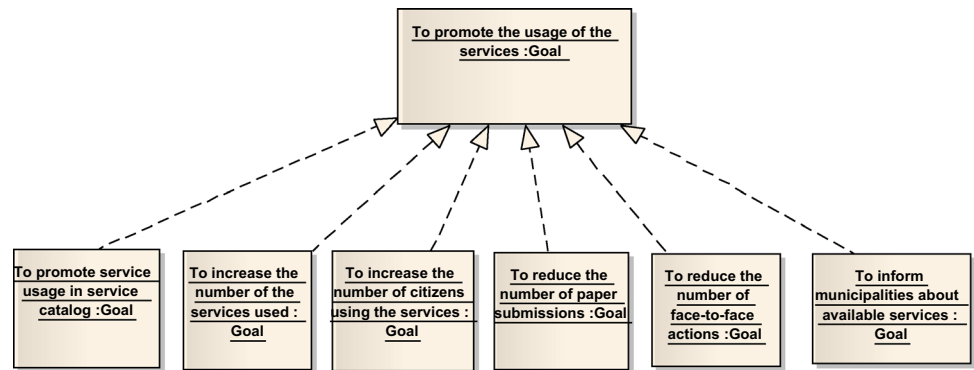
Table 1 SP Goals KPI

KPI name	Target value	Segments	Related goal
Citizens consuming the services (%)	25	Idle ≥ 25 Acceptable: 5–25 Warning < 5	To promote the usage of the services
Services in active use from all services provided by municipality (%)	100	Idle > 90 Acceptable: 70–90 Improvable 50–69 Warning < 50	To increase the number of the services used
Growth of the number of citizens using the services per month (%)	5	Idle ≥ 5 Acceptable: 1–4 Improvable 0–1 Warning < 0	To increase the number of citizens using the services
Completed service actions/submissions (%)	90	Idle ≥ 90 Acceptable: 50–90 Improvable < 50	To increase the number of completed service actions/submissions
Municipalities starting to use a service after received information (%)	90	Idle ≥ 90 Acceptable: 60–90 Improvable < 60	To inform municipalities about available services
Face-to-face actions (from all actions where online service is available) (%)	30	Idle: 0–30 Acceptable: 31–40 Improvable > 40	To reduce the number of face-to-face actions
Paper submissions (from all submissions where online submission is available) (%)	30	Idle: 0–30 Acceptable: 31–40 Improvable > 40	To reduce the number of paper submissions

1. The SP capability depends on the usage data for a large number of services, meaning that the usage of the service in other municipalities need to be monitored.
2. Process and process variants are dependent on the context data, thus the association between process and context situation (which describes the context data in capability delivery phase) is highly important for the SP capability.
3. Resources should be included in the capability model only if it has a significant role for delivering the capability. In this case, services are located in the SOA platform and highlighting is done automatically by a database procedure which is not treated as a resource.

4.2 Validation Results

The capability models were developed for the three use cases and the companies have approved the models as sufficiently describing their business problems. However, several differences have been observed that should be taken into account during the further elaboration of the CDD methodology.

A majority of CDD meta-model concepts and their associations have been used in the instantiation of the meta-model. A detailed overview of usage of the concepts and associations is provided in (Berziša et al. 2014). The resource concept was not used in some of the capability

Table 2 SP Context elements in tabular view

Context element	Context type	Possible values	Measurable properties	Mapping measurable property to value
Municipality size	Static	{Small, medium, large}	Number of citizens	If number of citizens <10,000 then 'small'
				If number of citizens 10,000–30,000 then 'medium'
				If number of citizens >30,000 then 'large'
Usage of the service in other municipalities	Dynamic	{High, medium, low}	Percentage of municipalities using the service	If municipalities using the service <20 %, then 'low'
				If municipalities using the service 20–50 % then 'medium'
				If municipalities using the service >50 % then 'high'
Amount of actions/ submissions per month	Dynamic	[0…10,000]	Number of actions/ submissions per month	n/a
Feedback in social networks	Dynamic	{Very negative, negative, neutral, positive, very positive}	Number of positive remarks in Facebook, Twitter	List of expressions/key words describing online services feedback
Type of highlighting	Static	{Automatic, manual}	Type of highlighting	n/a

models. The context element and measurable property were related one-to-one, especially, for the context elements known at the design time. The context type was not always used in the model because its efficient use depends on the availability of a taxonomy of context types.

The process variant was used: (1) to represent variants of the capability process; and (2) to represent variation within the process variant itself. The former can be considered as global process variants while the latter can be seen as local process variants. The global process variants were represented using the process variant concept. The local process variants can be perceived as non-standard ways of business process modeling. Therefore, it is not directly represented in the CDD meta-model. It is assumed that local process variants are useful if there are many different process variants. Rules for developing process variants should be defined at the design time of the capability.

The pattern concept was used to represent: (1) the solution supporting the capability; and (2) reusable components used in design of the process variants. Similarly to the process variants the former can be seen as a global pattern supporting the capability as a whole and the latter can be seen as a local pattern supporting parts of the process. The global pattern is created for every new capability developed and latter can be used in design of new capabilities. The local patterns are retrieved from the pattern repository and are used to create process variants. The patterns also contain information about how to perform the run-time adjustment.

The validation results show that different approaches to specialization of capabilities are possible. The standard process delivery capabilities at SIV AG are further specialized depending on design time context situation while Everis SP capability is defined as an individual capability and context dependence occurs only within the capability definition. The CDD methodology should provide guidance for developing and managing hierarchy of related capabilities. The capability oriented specialization shows context dependence at the business level and process variants show context dependence at the business and conceptual solution level.

5 Capability Driven Development Approach

The CDD methodology is based on agile and model driven IS development principles and consists of the CDD development process, a language for representing capabilities according to the CDD meta-model, as well as modeling tools. The main principles of the CDD methodology are:

- Use of enterprise models understandable to business stakeholders,
- Support for a heterogeneous development environment as opposed to a single vendor platform,
- Equal importance of both design-time and run-time activities with clear focus on different development artifacts,
- Rapid development of applications specific to a business challenge,
- Search for the most economically and technically advantageous solution,

Fig. 4 SP Process model
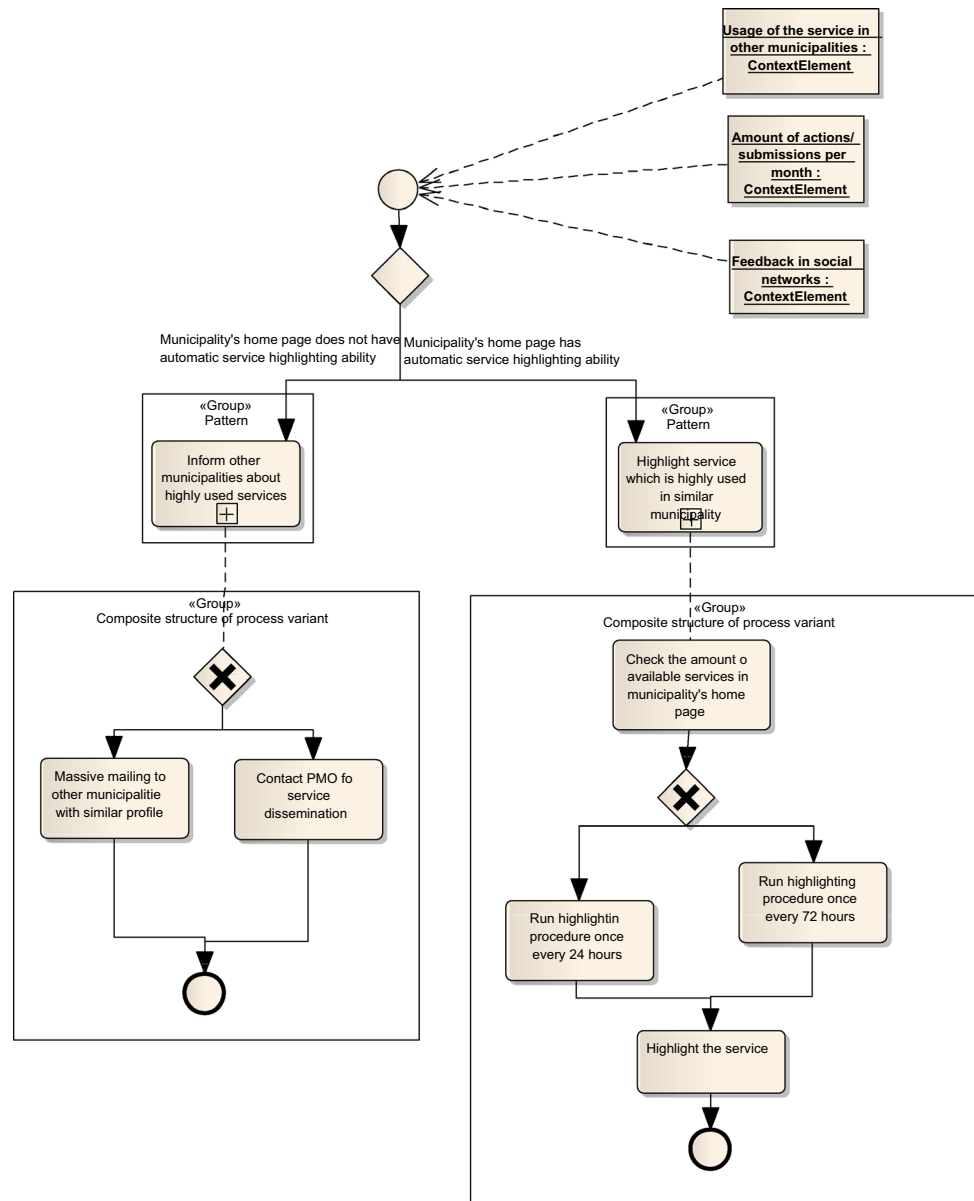
An overview of the envisioned CDD process is shown in Fig. 5. It includes three phases (Enterprise Modeling, Design, and Delivery) as well as two supporting activities (Management and Feedback).

Enterprise Modeling helps defining the overall business design and its result serves as input for the capability design. Capabilities usually serve business objectives and are linked to business strategies. The interrelations between objectives, strategies, structures and processes are captured in enterprise models. Thus, capabilities often are designed on the basis of enterprise models. If the company already has some models in place, this phase can mostly focus on reviewing and, if needed, refining them. In other cases, if the capability to be designed addresses new business area and/or

solutions, then the whole business design might need to be modeled first.

Design – the capability design explicitly focuses on evaluation of different business service designs in various delivery contexts as capabilities are being customized to specific requirements.

Delivery – the delivery phase concerns the actual utilization of the capability enabled by supporting information systems (i.e., capability delivery environment) with the intention to meet company's business goals in continuously evolving circumstances.

Feedback activity is used mainly due to the need to take into account additional context factors because not all relevant factors can be identified during the first development iteration.
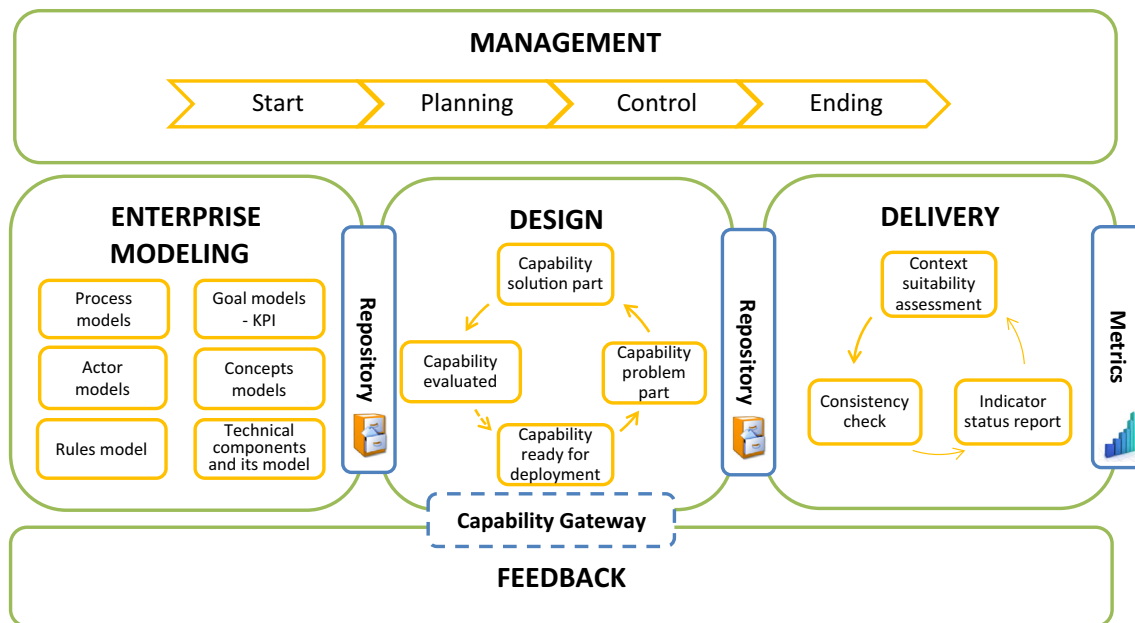
Fig. 5 CDD methodology

A *management* activity is explicitly represented in the CDD method to support its use in a project setting by supporting the capability development and management life-cycle including project planning, performing, control and ending activities.

6 Concluding Remarks and Future Work

We have proposed an approach, Capability Driven Development (CDD), that integrates organizational development with IS development taking into account changes in the application context of the solution. The approach is based on an EM process – it is based on EM components understandable to business stakeholders, such as goals, KPIs, processes, and resources and in principle is independent of any specific EM language. The linkage of the available enterprise components with different business contexts is done relying on the principle of reusing organizational patterns. In the CDD meta-model, patterns represent reusable solutions in terms of the design of business process, resources, and supporting IT components for delivering a specific type of capability in a given context. We have presented a validation case from the e-government field.

Two important challenges to be addressed are the availability of patterns and the implementation of algorithms for dynamic adjustment of the capability delivery application. In order to ensure pattern availability an infrastructure and methods for life-cycle management of patterns is required. In some cases, incentives for sharing patterns among companies can be devised. There could also be a selection of different adjustment algorithms. Elaboration and implementation should follow a set of general, open principles for incorporating algorithms developed by third parties.

References

ArchiMate (2013) An enterprise modeling language from the Open Group. http://www.opengroup.org/archimate/. Accessed 3 Dec 2014

Asadi M, Ramsin R (2008) MDA-based methodologies: an analytical survey. In: Proceedings Model driven architecture – foundations and applications (ECMDA-FA 2008), LNCS 5095, pp 419–431

Barney JB (1991) Firm resources and sustained competitive advantage. J Manag 17(1):99–120

Bērziša S, Bravos G, Gonzalez Cardona T, Czubayko U, España S, Grabis J, Henke lM, Jokste L, Kampars J, Koc H, Kuhr J, Llorca C, Loucopoulos P, Juanes Pascua lR, Sandkuh lK, Simic H, Stirna J, Zdravkovic J (2014) Deliverable 1.4: Requirements specification for CDD, CaaS – capability as a service for digital enterprises. FP7 project no 611351, Riga Technical University, Latvia. Submitted for review

Bubenko JA Jr, Persson A, Stirna J (2001) User guide of the knowledge management approach using enterprise knowledge patterns. Deliverable D3, IST programme project hypermedia and pattern based knowledge management for smart organisations. project no. IST-2000-28401, Royal Institute of Technology, Sweden

Briand LC, Yue T, Labiche Y (2011) A systematic review of transformation approaches between user requirements and analysis models. Requir Eng 16:75–99

De Kinderen S, Gordijn J, Akkermans H (2009) Reasoning about customer needs in multi-supplier ICT service bundles using decision models. In: Proceedings 11th international conference on enterprise information systems (ICEIS 2009), pp 131–136

Deloitte (2009) Cloud computing: forecasting change. Deloitte Consulting, New York. http://public.deloitte.no/dokumenter/2_Cloud_Computing%5B1%5D.pdf. Accessed 3 Dec 2014

Dey A (2001) Understanding and using context. Pers Ubiquitous Comput 5(1):4–7

Gamma E, Helm R, Johnson R, Vlissides J (1995) Design patterns: elements of reusable object-oriented software architecture. Addison-Wesley, Boston

Gomes D, Gonçalves JM, Santos R, Aguiar R (2010) XMPP based context management architecture. In: Proceedings GLOBECOM workshop, IEEE, pp 1372–1377

González A, España S, Ruiz M, Pastor Ó (2011) Systematic derivation of class diagrams from communication-oriented business process models. In: 12th working conference on business process modeling, development, and support (BPMDS'11). Springer LNBIP 81, pp 246–260

Henkel M, Stirna J (2010) Pondering on the key functionality of model driven development tools: the case of mendix. In: Forbrig P, Günther H (eds) Proceedings business informatics research (BIR 2010), Springer LNBIP 64, pp 146–160

Hervas R, Bravo J, Fontecha J (2010) A context model based on ontological languages – a proposal for information visualisation. J Univers Comput Sci 16(12):1539–1555

Jarke M, Loucopoulos P, Lyytinen K, Mylopoulos J, Robinson W (2011) The brave new world of design requirements. Information Syst 36(7):992–1008

Kaplan RS, Norton DP (2004) Strategy maps: converting intangible assets into tangible outcomes. Harvard Business School Press, Boston

Kleppe A, Warmer J, Bast W (2013) MDA explained. Addison-Wesley, Boston

Loniewski G, Insfran E, Abrahao L (2010) A systematic review of the use of requirements engineering techniques in model-driven development. In: Proceedings model driven engineering languages and systems (MODELS 2010), Part II, LNCS 6395, pp 213–227

Mohagheghi P, Dehlen V (2008) Where is the proof? - a review of experiences from applying MDE in industry. In: Proceedings model driven architecture – foundations and applications (ECMDA-FA 2008). LNCS 5095. Springer, Heidelberg, pp 432–443

Nilsson AG, Tolis C, Nellborn C (eds) (1999) Perspectives on business modelling: understanding and changing organisations. Springer, Heidelberg

OASIS (2011) Reference architecture foundation for service oriented architecture version 1.0, committee specification draft 03/public review draft 02 06 July 2011. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra.pdf. Accessed 3 Dec 2014

OMG (2011a) UML superstructure. http://www.omg.org/spec/UML/2.4.1/. Accessed 3 Dec 2014

OMG (2011b) Business process model and notation. http://www.omg.org/spec/BPMN/2.0/. Accessed 3 Dec 2014

Osterwalder A, Pigneur Y (2003) Modeling value propositions in e-business. Proc 5th international conference on electronic commerce (ICEC 2003). ACM Conference Proceedings Series 50

Papazoglou MP, Yang J (2002) Design methodology for web services and business processes. In: Proceedings 3rd international workshop on technologies for e-services (TES 03). LNCS 2444. Springer, Heidelberg, pp 54–64

Pastor O, Giachetti G (2010) Linking goal-oriented requirements and model-driven development. In: Intentional perspectives on information systems engineering. Springer, Heidelberg, pp 257–276

Porter ME (1985) Competitive advantage: creating and sustaining superior performance. Free Press, New York

Ruiz M, Costal D, España S, Franch X, Pastor Ó (2014) Integrating the goal and business process perspectives in information system analysis. In: 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014). LNCS 8484. Springer, Heidelberg, pp 332–346

Sheng Q, Yu J, Dustar S (eds) (2010) Enabling context-aware web services: methods, architectures, and technologies. Chapman & Hall/CRC, Boca Raton

Smanchat S, Ling S, Indrawan M (2008) A survey on context-aware workflow adaptations. In: Proceedings 6th international conference on advances in mobile computing and multimedia (MoMM 2008), New York, pp 414–417

Stirna J, Grabis J, Henkel M, Zdravkovic J (2012) Capability driven development – an approach to support evolving organizations. In: The practice of enterprise modeling (PoEM 2012). LNBIP 134. Springer, Heidelberg, pp 117–131

Vale S, Hammoudi S (2009) COMODE: a framework for the development of context-aware applications in the context of MDE. In: Proceedings 4th international conference on internet and web applications and services (ICIW 2009). IEEE, Venice, pp 261–266

Wesenberg H (2011) Enterprise modeling in an agile world. In: Johannesson P, Krogstie J, Opdahl AL (eds) The practice of enterprise modeling (PoEM 2011). LNBIP 92. Springer, Heidelberg, pp 126–130

Zdravkovic J, Stirna J, Henkel M, Grabis J (2013) Modeling business capabilities and context dependent delivery by cloud services. In: Proceedings advanced information systems engineering (CAiSE 2013). LNCS 7908. Springer, Heidelberg, pp 369–383