Capability-Driven Development of a SOA Platform: A Case Study*

Sergio España[1], Tania González[2], Jānis Grabis[3], Lauma Jokste[3], Raúl Juanes[2], and Francisco Valverde[1]

[1] Research Centre on Software Production Methods (PROS)
Universidad Politécnica de Valencia, Spain
{sergio.espana,fvalverde}@pros.upv.es
[2] everis, Spain
{tania.gonzalez.cardona,raul.juanes.pascual}@everis.com
[3] Information Technology Institute
Riga Technical University, Latvia
{grabis,lauma.jokste}@rtu.lv

Abstract. Capability-driven development (CDD) is a novel paradigm for organisational modelling and information technology development. Its cornerstones are capability modelling (including goals, context, processes), pattern-based design, and runtime context awareness and service delivery adjustment. There is a lack of empirical studies regarding the industrial application of CDD. This paper reports on a case study that focuses on capability modelling within a service-oriented architecture development project. We have collected lessons learned, as well as open challenges to feedback the improvement of the CDD methodology.

Keywords: Information systems, capability-driven development, enterprise architecture, case study, context modelling, business process modelling.

1 Introduction

Capability is a concept that has been used for some time in disciplines such as organisational management [1] and welfare economics [2], and it is used in defence technology development. However, when applied to information technology (IT) development, there is much debate on how the concept of capability relates to other widely used concepts, such as business process, business service, goals, etc. [3, 4].

Recently, a metamodel for capability modelling has been proposed [5]. Within the European Commission FP7 Project CaaS, a methodology and tools to support capability-driven development (CDD) are being developed.

Despite the growing use of the capability concept, there are no empirical validations of its application to IT developments. This paper presents a case study research that reports on a software project undertaken in everis, a multinational firm. everis applied a preliminary version of the CDD methodology and supported it with the modelling tools the team had at hand. The paper contributions are the following:

* This work has been partially supported by the EU-FP7 funded project no: 611351 CaaS – Capability as a Service in Digital Enterprises.

L. Iliadis, M. Papazoglou, and K. Pohl (Eds.): CAiSE 2014 Workshops, LNBIP 178, pp. 100–111, 2014.

- We report on the case study, its protocol and qualitative findings.
- We discuss the lessons learned about the application of capability modelling to IT development and we highlight challenges for improving the CDD methodology.

The rest of the paper is structured as follows. Section 2 overviews the CDD paradigm. Section 3 presents the research methodology. Section 4 reports on the case study (an e-government service platform), including a discussion on lessons learned and open challenges. Section 5 discusses the validity of the results. Section 6 concludes.

2 Capability-Driven Development of Information Technology

From the business perspective, a capability is the ability and capacity that enables an enterprise to achieve a business goal in a certain context. From the technical perspective, capability delivery requires dynamic utilisation of resources and services in dynamically changing environments. For instance, if we provide an e-government service to a given municipality, we need to react to changes that might happen throughout the year, and we may also want to provide the same service to other municipalities with a different context (e.g. different population, laws).

This principle of describing a reusable solution to a recurrent problem in a given context has been adopted in domains such as organisational design [6], business modelling [7], knowledge management [8], and workflow management [9]. Open challenges are the proper integration of conceptual reuse approaches (e.g. patterns, components) with business design and the provision of an adequate tool support.

The specification of context-aware business capabilities, by using enterprise modelling techniques, can be the starting point of the development process. Following this approach, business services are configured by enterprise models and built-in algorithms that provide context information. Capability-driven development (CDD) is a novel paradigm where services are customised on the basis of the essential business capabilities and delivery is adjusted according to the current context [5, 10]. For supporting CDD, the CaaS project has envisioned the following main components:

- *CDD methodology*: an agile methodology for identification, design and delivery of context aware business models. It formalizes the notion of capability by means of a metamodel that comprises the following elements [10]:
  - *Goal*: desired state of affairs that needs to be attained.
  - Key performance indicator (KPI): for monitoring the achievement of a goal.
  - *Context*: characterisation of situations in which a capability should be provided.
  - *Capacity*: resources (e.g. money, time, staff, tools) for delivering the capability
  - *Ability*: competence (i.e. talent, intelligence and disposition), skills, processes.
- *Capability delivery patterns*: they are generic organisational designs and business processes that can be easily adapted, reused, and executed.
- *CDD environment*: tool support for design (e.g. capability modelling) and runtime (e.g. the context platform monitors changes and the capability navigation delivery application calculates KPIs and selects the most suitable pattern) of CDD solutions.

3 Research Goal and Methodology

Our goal is gathering knowledge on the results of the application of capability modelling in industry. We specifically target the lessons learned during the application of the CDD methodology, as well as identifying current challenges that ought to be addressed in future improvements. We have structured the research methodology following the Design Science approach [11] (see Figure 1).

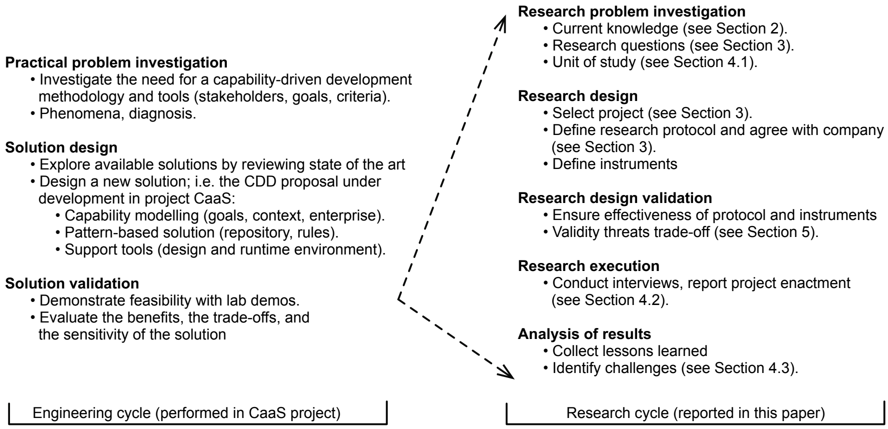


Fig. 1. Overall structure of the research methodology

The investigated project has been enacted in industry, so we have selected the case study research methodology. The actual interaction between the company and the researchers is closer to a case study research than to an action research since (i) the participation of the company in CaaS project meetings prior to the application of the CDD methodology made a method transfer process unnecessary, (ii) the company had much degree of freedom to apply the methodology, and (iii) the researchers mainly acted to solve some doubts formulated by the company and to conduct interviews and gather the data that is reported in Sections 4 and 5. The checklist by Runeson and Höst [12] served as guideline for conducting and reporting the case study research. The reader should consider the exploratory nature of this case study.

4 Case Study

4.1 The Company and the Project

The **case-study company** is everis, a multinational firm offering business consulting, as well as development, maintenance and improvement of IT. Within the public administration sector, everis has wide experience in projects related to modernisation of public procurement management, education, e-government, health, justice, etc.

The **unit of analysis** is a project to improve a service-oriented architecture (SOA) platform for e-government. It aligns with the Spanish administration goal of sharing human resources, software and hardware to support e-government. The most valuable feature of the SOA platform is offering electronic services provided by municipalities to citizens and companies. By the end of 2013, the platform provided a service catalogue of around 200 services (e.g. marriage registration application, public pool booking, taxes). Approximately 50 of them are in active use in 250 municipalities. As a result, over 1 million Spanish citizens benefit from using the SOA platform.

We selected this project because the platform context is complex and volatile; for instance, each municipality has a distinct profile, citizens have different interests, and laws and regulations change frequently. everis has to adapt the electronic services when the platform is deployed for a new municipality and whenever the context changes. For the time being, service customisation is done at code level.

The **main challenges** are (i) to perform organisational actions tailored for a specific municipality in a given moment in time (i.e. taking into account the period of the year, real-time usage indicators, calendar events, or most requested services in a certain period of time), and (ii) to automate the adaptation of the supporting IT.

By means of applying CDD methodology and tools, everis intends to adapt its way of working and to evolve the SOA platform into a context-aware, self-adaptive platform. In this first attempt to apply CDD, everis set up the following team:

- A *Public Sector and R&D Manager*, has over 12 years of experience in the IT sector for public administrations and that has led several innovation projects. This role has a mixture of knowledge about the SOA platform, the CDD methodology, and also of the results expected by public administrations. He is author number 5.
- A *Business Consultant*, with concrete expertise in the CDD methodology, who is willing to apply the CDD paradigm to several projects, and with little initial knowledge of the use case domain (i.e. the SOA platform). She is author number 2.
- A *Technological Consultant*, with concrete expertise in the SOA Platform, whose responsibility is improving the services provided by the SOA platform, but with no initial knowledge about the CDD Approach.

This team had the support of academic partners that are part of the CaaS consortium. Authors 1, 3, 4 and 6 are among them.

4.2 The Application of the CDD Methodology

During this project, everis has approached CDD from a goal-first perspective. This means that the goal model was created in the first place and then the rest of the models (e.g. stakeholders, context) were reasoned taking the goal model as input. However, this was not the initial intention and the approach rather emerged as capability modelling turned out to be more complex than expected. Initially, the Public Sector and R&D Manager and the Business Consultant organised a brainstorming session in which the overarching questions were: What type of new or adapted solutions can everis provide their customers by applying CDD? How can everis measure the accomplishment and the benefits of these solutions? Several staff members envisioned possible capabilities and specified them using a textual template.

Table 1. Initial capability drafts expressed during brainstorming

Capacity: IT infrastructure, monitoring tool, developers, technicians.	*Capacity*: swimming pool facilities, swimming coaches.
Ability: being able to deploy a maintenance portal.	*Ability*: offer the electronic service to request swimming course registration.
Enterprise: everis	*Enterprise*: municipality
Goal: keep services available despite platform errors.	*Goal*: reduce cost of service provision.
Context: loss of connectivity w. other subsystems.	*Context*: amount of requests.
Goal KPI: time service available / time error in platform	*Goal KPI*: amount of money saved

Table 1 shows a sample of two out of the eight capability descriptions. Such descriptions were sent to the academic partners in UPV, along with an invitation to meet in order to discuss goal modelling. The joint meeting clarified the different perspectives and granularities from which capabilities can be conceived. This discussion paved the way for everis to focus on the business goals and, therefore, a goal-first approach was adopted. Figure 2 depicts the resulting flow of capability modelling activities enacted during the project. For the moment, CDD methodology is notation-agnostic (e.g. for business process modelling one can use either BPMN, Activity Diagrams, Communicative Event Diagrams [13], etc.).

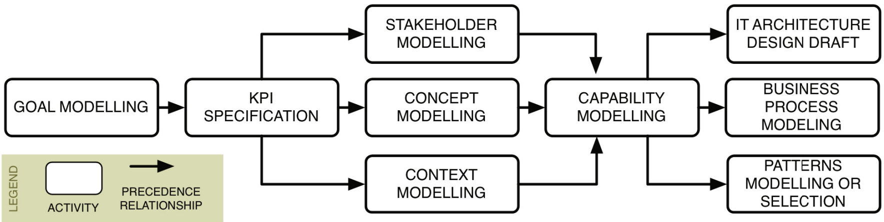


Fig. 2. Flow of activities enacted during the project

As first step towards clarifying the capabilities pursued in the SOA platform project, everis performed a modelling session in which the goals were elicited and modelled graphically (see Figure 3).

The model was mainly created from the perspective of everis objectives towards the project. Table 2 shows a sample of goal specifications. In order to facilitate reasoning, goals were classified into five categories:

- *Strategic goals* refer to improving services and their usage (G-1 to G-5).
- *Business goals* are mostly related to the ability to identify changes in usage of services and changes in services themselves (G-6 to G-9).
- *Technical goals* relate to service usage and platform collocation (G-10).
- *Design time goals* relate to service design requirements and to the identification of change patterns (G-11 to G-13).
- *Run-time goals* relate to the run time of the SOA platform, such as usage of patterns, dynamic adjustment, automated responses, etc. (G-14 to G-18).

In order to measure goal achievement, key performance indicators (KPIs) were defined using templates. For the sake of brevity, we include only a few in Table 2.

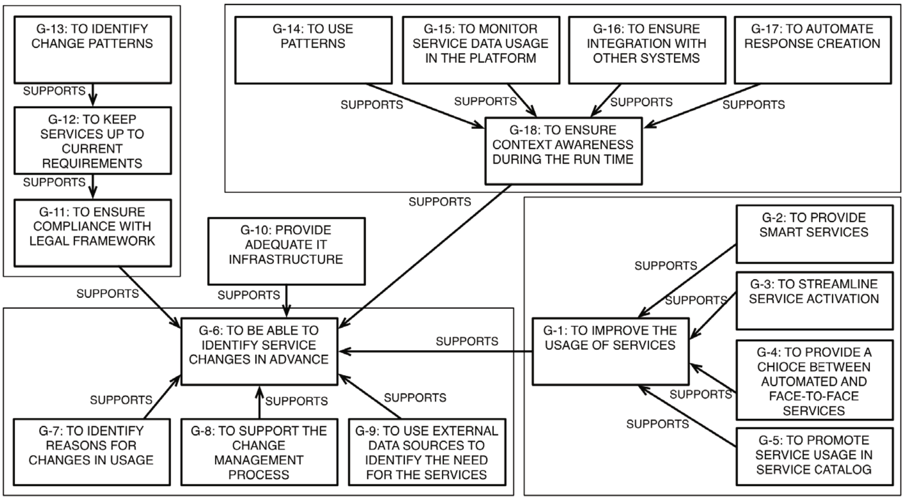


Fig. 3. Goal model of the project

Table 2. Sample of goals and KPIs of the SOA platform project

G-1. To improve the usage of the services At the time being EVR provide up to 200 services for 250 municipalities, but only 100 services are in active use and not in all municipalities. The goal is to improve the usage of the services. This goal is supported by other strategic and business goals. *Category*: Strategic goal *Stakeholder*: S-3. EVR *KPIs*: Percentage of citizens consuming the services (*target*=25%) Percentage of completed service actions / submissions (*target*=90%)	
G-6. To identify service changes in advance The services provided by the SOA platform are affected by changes. These include changes in requirements, environment and other aspects. The goal is to proactively identify possible changes in the services. This goal includes sub-goals G-2, G-3 and G-4. *Category*: Business goal *Stakeholder*: S-3. EVR *KPIs*: Frequency of change in current services	

Stakeholders were identified from the current business processes. They are considered responsible for reaching the goals described above. Three important stakeholders are the *end users* (companies and citizens), the *project management office* (PMO, who is responsible for coordination in collaborative projects) and *municipalities* (a general-purpose administrative subdivision -as opposed to a special-purpose district- and the smallest administrative unit in a province). Municipalities carry out the services provided to end users (e.g. registering marriage applications).

With regards to the concepts model, it contains the main concepts that are used to describe the SOA platform, and not those related to individual services. For the sake of brevity, we do not include the stakeholder model and tables, or the concepts model.

From the point of view of everis, the main goal of the project is to improve service usage in the SOA platform (G-1); one of the mechanisms to achieve this is by service promotion (G-5). The purpose is to highlight services in the municipality homepage in case this service is highly used in municipalities with similar profile (e.g. number of citizens, location -coast or inland-) or if the context is favourable (e.g. hot weather increases pool booking, marriage applications increase on the week of Valentine's day). Due to technological development decisions, some homepages cannot automatically highlight services. The graphical context model is omitted for reasons of space. Table 3 presents a sample of three out the fourteen context elements. A set of rules maps contextual indicators with measurable properties. Other elements refer to the legislation, the time of the year and week, social network information, pool visitor data, weather, etc.

Table 3. Sample of context element specifications

Element	Values	Measur. prop.	Mapping rules
Municipality size	{Small, Medium, Large}	Number of citizens	If number of citizens <10 000 then 'small' If number of citizens 10000- 30000 then 'medium' If number of citizens >30000 then 'large'
Service usage in other muni-cipalities	{High, Medium, Low}	Percentage of municipalities using the service	If municipalities using service < 20%, then 'low' If municipalities using service between 20 and 50% then 'medium' If municipalities using service >50% then 'high'
Type of highlighting	{Automatic, Manual}	*NA*	*NA (unknown at design time)*

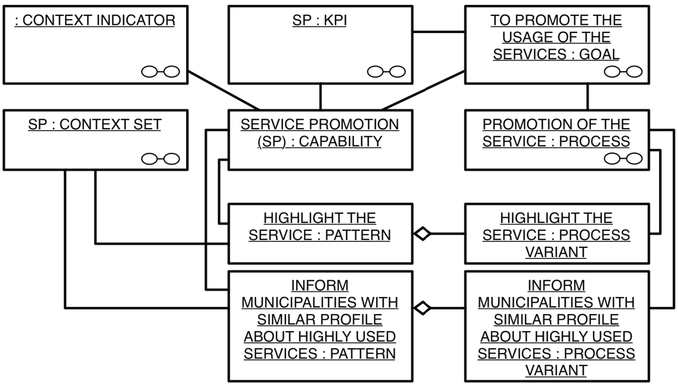


Fig. 4. Service promotion capability model

Figure 4 presents the service promotion capability model, which graphically summarises the capability by aggregating all its related elements. It includes the process Promotion of a service, which promotes services in one municipality whenever that service is being highly used in similar municipalities. This process is detailed in Figure 5 and has two main process variants:

- If the municipality homepage has automatic service highlighting then service highlight procedure is executed. Depending on different context data, service highlight procedure can be run once every 24 hours or once every 72 hours.
- If automatic highlighting is not possible or municipality with similar profile does not have that particular service, then an email is sent to municipality or to the PMO recommending service promotion.

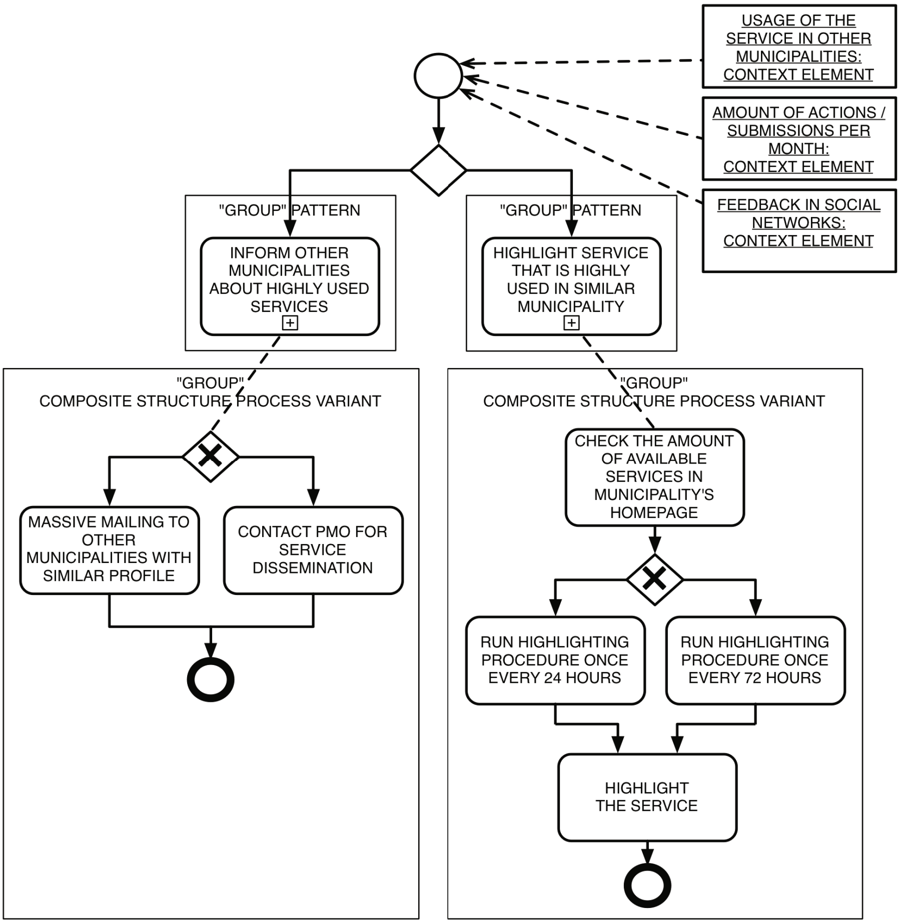


Fig. 5. Service promotion process model

Service usage context is monitored at run time:

- If service usage is high, then service can be highlighted in similar municipalities (similar size and profile);
- If the citizens feedback about service usage in social networks is positive, then the service is highlighted in similar municipalities (similar size and profile);
- If the municipality homepage is not able to automatically highlight the service, an email is sent to the municipality reporting on the high usage of services.
- If municipality A is not offering a specific service that has a high usage in other municipalities that have a similar profile to A, then the PMO is sent an email recommending deploying the service in A.

The process model with its process variants and capability delivery patterns is shown in Figure 5, the BPMN notation has been used. Note that, while process variants are depicted as separate elements in capability models (see Figure 4), in the process models the variants are included in the same diagram. The starting event of the process is conditionally evaluated by an expression taking the context elements as inputs. In this case, the context elements used are municipality size, usage of the service in other municipalities and social networks feedback. The expression uses these context elements to determine a need for running the service highlight process. The reader should take into consideration that everis slightly extended the BPMN notation in an exploratory attempt to model aspects of the capability solution that are currently not covered by business process modelling notations. We further discuss this issue in Section 4.3.

4.3 Lessons Learned and Open Challenges

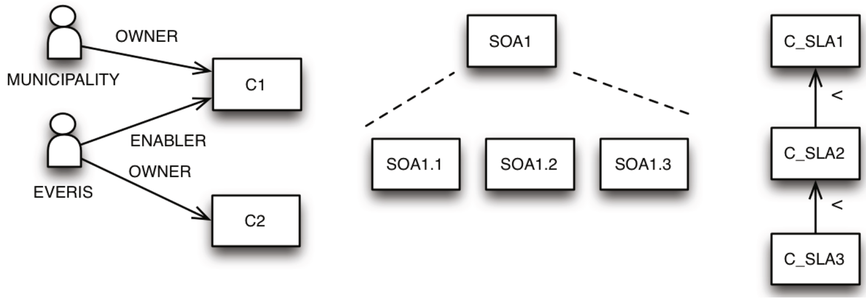
We now discuss on some lessons learned from observing everis apply CDD to the SOA platform project. During the project, everis team members were motivated to enact the methodology, but also found several difficulties regarding the instantiation of the capability metamodel.

Regarding the **motivation to adopt CDD**, we noted that the driver for improving the SOA platform was twofold: not only the perception of current limitations in the platform, but also the expectations of new features that the CDD methodology and tools can enable. For instance, the industrial stakeholders feel confident that the CDD runtime environment will be able to automatically adapt the SOA platform to changing contexts. They enter into capability modelling with the intention to characterise such contexts and specify the rules for self-adaptation.

As mentioned above, the initial descriptions of capabilities related to the SOA platform differed in perspective (e.g. EVR vs. municipality) and granularity (e.g. related to the SOA platform as a whole or to an individual service).

The subsequent meeting with academic partners from UPV and later project plenary meetings revealed that capabilities have relationships among them. An initial characterisation of such relationships was done. We envision the need for at least three types of **relationships among capabilities** (see Figure 6). More research is needed regarding capability relationships related to:

- *Perspective*. For instance, municipalities are the owners of some capabilities (e.g. C1), while everis are the enablers of such capabilities and, in turn, owners of other capabilities related to the prior (C2).
- *Refinement*. Some capabilities (e.g. SOA1) must be refined in smaller ones (SOA1.1 to SOA1.3) in order to handle them more easily.
- *Context or quality levels*. Some capabilities need to be ordered because they refer to different levels of the context (e.g. high attendance to a cultural event vs. small events) or the agreed quality (C_SLA1 to CSLA3).



a) Perspective b) Refinement c) Quality level

Fig. 6. Illustration of types of capability relationships

The Public Sector and R&D Manager observed during the brainstorming that, depending on the profile of the analyst the outlined capability had a different perspective and granularity. His impression was that there is a need for guidelines.

By the time the goals model, the KPIs, the stakeholders model and the concepts model were created, the analysts had already realised that they could conceive capabilities (i) from the point of view of everis and (ii) from the point of view of the municipalities. The former are capabilities possessed by everis as PMO and the SOA platform provider; the later are capabilities possessed by the municipalities although delivered by everis using the SOA platform on behalf of the municipalities.

Based on our observations during this case study and comments of other CaaS industrial stakeholders during project meetings, there is an open challenge related to **how to start capability modelling**. everis performed a goal-first capability modelling, but one could also start CDD by modelling context or resources. Also, if current business process models exist, CDD enters a reengineering scenario that may differ in terms of the flow of modelling activities and guidelines. We plan to compare these starting points in future work.

In the SOA platform project, everis brought together a consultant that was knowledgeable in the CDD paradigm, a consultant that was knowledgeable in the SOA platform and its business process, and a manager who bridged the gap by standing somewhere in the middle. Although such collaboration for capability modelling worked well in this case, both the company and the researchers wonder whether this third role is needed. Also, in case it is indeed needed, an open question is who should play it (someone from the organisation, an external stakeholder) and whether his/her competence can be reused for different projects.

Regarding **modelling notations**, an open research challenge is discovering which notations are more suitable for each of the model views, whether different notations require specialised guidelines or extensions (e.g. everis connected context elements with elements of the BPMN diagram, see Figure 5), and whether situational guidelines are needed to adapt to project contingencies.

The identification and modelling of **variability** is key to CDD. To avoid manual customisation of services software code, everis intends to apply CDD so as to identify the variability in the context and, in design time, define solution patterns that deal with such variability. Following the CDD vision, at run-time, a context platform will enable the SOA platform to be context-aware and automatically select the patterns that suit the context. Above, we have shown the variability related to the automation (or lack of it) of service highlighting. Other main factors of variability in the project are the existence of different facilities provided by a municipality (e.g. public pool, marriage registration institution), the characteristics of the facilities (e.g. pool size, opening hours), and the legislation affecting the services. Variability brings challenges to CDD that need further investigation.

5 Discussion on Validity and Ethical Concerns

This case study is one of many milestones that are planned during the three-year span of CaaS project. Both the capability-modelling endeavour by everis and the case study research were of exploratory nature. Two facts make us cautious regarding everis perception of the utility of the approach. First, the development team is motivated to apply the CDD methodology. Second, they have high expectations towards the CDD runtime environment. Also, to obtain evidences and evaluate the benefits and drawbacks of CDD, further research is needed, especially when new versions of CDD are issued and applied. Instead, we focused on identifying the key lessons learned and future challenges.

As mentioned above, the SOA platform project was selected because its characteristics (project size, dynamic and changing context, high variability) suited our research goals. The interviews were not recorded; instead, the researchers made annotations using note-taking software (e.g. Evernote), conceived hypotheses (e.g. the consultants had troubles related to the perspectives of capabilities) and formulated additional questions to verify them. After every interview, the minutes were collaboratively edited. To mitigate threats to the validity of our conclusions, several researchers were involved in later discussions, so as to reduce researcher bias and achieve inter-subjective agreement. The lessons learned and open challenges were subjected for the consideration of the Public Sector and R&D Manager and the Business Consultant and they expressed their agreement.

We are aware that only one case has been analysed and, in order to avoid threats to the external validity of the results, other case study researches should be conducted. In any case, we argue that the results are a valuable feedback for CDD methodology improvement.

With regards to ethical concerns, everis team members and managerial staff were aware of the goals of this research and consented on publicly reporting the results.

6 Conclusions

Capability modelling is central to capability-driven development (CDD). We have conducted a case study research on the industrial application of capability modelling in a SOA platform project. By observing the process and the results, we can conclude that CDD facilitates a systematic analysis of organisational needs and designing an IT solution that is aware of the context so as to adjust the business services to changes in the environment. The CDD methodology still needs improvement and we have identified some open challenges, such as the need to provide relationships among capabilities, and the need for guidance with regards to the flow of modelling activities or how to use modelling notations.

As future work, we plan to design guidelines for goal-first capability modelling, to conduct a controlled experiment with students to validate some aspects of the guidelines and to conduct additional case studies to assess the evolution of the CDD.

References

1. Barney, J.: Firm resources and sustained competitive advantage. Journal of Management 17(1), 99–120 (1991)
2. Sen, A.: Development as freedom. Oxford University Press (1999)
3. Sharp, A.: Capabilities, agile and “process blindess”. In: BPTrends (November 2011)
4. Sharp, A.: Peace accord reached! Process vs. capability debate ends with a whimper. BPTrends (October 2013)
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. LNBIP, vol. 134, pp. 117–131. Springer, Heidelberg (2012)
6. Niwe, M., Stirna, J.: Pattern approach to business-to-business transactions. In: International Conference for Internet Technology and Secured Transactions (ICITST 2009). IEEE, London (2009)
7. Business Model Innovation Hub (2013), http://businessmodelhub.com/ (cited March 2014)
8. Persson, A., Stirna, J., Aggestam, L.: How to disseminate professional knowledge in healthcare: the case of Skaraborg Hospital. Journal of Cases on Information Technology 10(4), 41–64 (2008)
9. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M.: Process-aware information systems: bridging people and software through process technology. Wiley (2005)
10. 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)
11. Wieringa, R.: Design science as nested problem solving. In: 4th International Conference on Design Science Research in Information Systems and Technology, pp. 1–12. ACM, Philadelphia (2009)
12. Runeson, P., Höst, M.: Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering 14(2), 131–164 (2009)
13. España, S., González, A., Pastor, Ó.: Communication analysis: A requirements engineering method for information systems. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 530–545. Springer, Heidelberg (2009)