An Empirical Evaluation of Capability Modelling using Design Rationale

George Bravos[1,2], Pericles Loucopoulos[1,2,3], Christina Stratigaki[1], Dimitris Valvis[1]

[1] Fresh TL, The Innovation Centre, Sci-Tech Daresbury, Daresbury, U.K.
[2] Department of Informatics and Telematics, Harokopio University of Athens, Athens, Greece
[3] Manchester Business School, University of Manchester, U.K.
{gebravos, ploucopoulos}@hua.gr, {xristinastrathgakh, d.valvis}@gmail.com

Abstract. Capability-driven development (CDD) is an emerging research field aimed at aligning information technology (IT) to business evolution. From a methodological perspective a designer using CDD is faced with the challenge of reasoning about phenomena present in the business domain, capturing user requirements and developing an IT solution that reflects these phenomena and meets user expectation. Central to this process is the methodology meta-model, which is intended to define both the key concepts on which the designer has to focus and the process to be followed. The purpose of this paper is to report on an investigation on the utility of a specific meta-model in terms of these two aspects. This investigation was carried out through a use case that involved capability modelling on the same application, by different designers. Each approach was documented using design rationale techniques. The two efforts were then analysed and observations about the capability driven design activities were defined. The output of this work has provided feedback to enhancing the capability meta-model and consequently the capability driven design activities in a number of important ways.

Keywords: Capability – driven development, Design rationale, reasoning cycle

1 Introduction

In today's dynamic business environment software systems need to evolve to reflect changes in user requirements [1, 2]. Against this dynamic business backdrop, emergent application software is regarded as a key component in the service industry of tomorrow [3]. The effective and efficient development of such systems can have a major impact on the economic value of digital companies – that is companies for which enterprise software becomes the decisive driver behind product and service innovation [4]. One important factor in achieving this is to ensure that the design and evolution of the software system can comply with the design and evolution of the enterprise. To this end, focus of development lifecycle model has shifted in recent years towards dynamic configuration using approaches such as agile methods, method-driven development and software-oriented architectures. These efforts are complemented by a response to adaptation at operational level by exploiting context-awareness [5]. However, there still exists a gap between enterprise requirements and software solutions [6, 7]. One recent development that attempts to bridge this gap is the use of capability-driven approaches.

The notion of 'capability' has been traditionally used in non-technical domains such as in socio-economic analyses [8-10], organizational studies [11, 12] and arguably most intensively in strategic management with a variety of theories that fall into three categories namely resource based value [13, 14], absorptive capability [15, 16], and dynamic capability [17-21]. Recently, a number of initiatives have begun to use the concept of 'business capability' as a way of linking enterprise aspects to software solutions. These efforts include information system agility [22, 23], service-orientation [24, 25], software process improvement [26] and business-IT alignment [27-29].

This paper focuses on one particular approach that falls in the last category specifically that of the CaaS (capability as a service) approach. The motivation of the work presented in this paper is the need to examine an initial version of the CaaS meta-model [29] about its utility to modelling business capabilities. More specifically, the question being answered is "to what extent does the meta-model support modelling activities in a consistent and generic manner?" To this end, a single use case was used where two designers completely independently, using the capability meta-model, attempted to capture and represent the enterprise requirements and their business setting.

The two different design efforts were formally documented using the generic model for reflective design, known as the *reasoning loop*, originally presented in [30]. This model is non-descriptive and affects minimally the design process, whist it provides the means of capturing and recording the design goals, the hypotheses for meeting these goals, the arguments for or against accepting a particular hypothesis and the actions that are needed in proceeding in the design tree. In order to record and analyse the design decisions of each one of the two modelers, the authors deployed the Compendium tool [31, 32].

The analysis that resulted from this study has given a number of key insights into the utility of the meta-model and has proved valuable in the meta-model's evolution as well as in impacting in the core elements of the methodology for a capability-driven development.

2 The Key Elements of the Evaluation Study

This section discusses the three essential elements involved in this study: (a) the meta-model being tested (section 2.1), (b) the use case used for the testing (section 2.2) and (c) the method by which design decisions were captured and subsequently analysed (section 2.3).

2.1 An Overview of the Meta-model

The capability meta-model that was examined was an early version of the CaaS framework [29]. This is graphically shown in Fig. 1. The model focuses on the representation of Goals, and the Processes realizing these goals using required Resources. These are essential components of business planning, and are common to many Enterprise Modelling (EM) approaches. In the CaaS project, the EM approach is based on the Enterprise Knowledge Development (EKD) paradigm [33, 34]. Furthermore, Key Performance Indicators (KPIs) are used to measure the achievements of goals. The main components in the meta-model needed for planning business variability are Capability and Context. In essence, Capability formulates the requirements for the ability of accomplishing a Goal, realized by applying a solution described by a capability delivery Pattern. This realization requires certain business Processes, Process Variants and Resources, such as infrastructure or IT components.

The distinguishing characteristic of Capability is that it is designed to deliver a business solution for specific Context Sets that are represented by Context Situations at runtime. It essentially links together business Goals, related business Processes and Resources, with delivery solutions by distinguishing the business contexts in which certain Patterns capturing business Process Variants should be applied.

In this framework, the specific paper focuses mainly on the terms of Context and Capability, also taking into consideration the Variability and the business Goals.

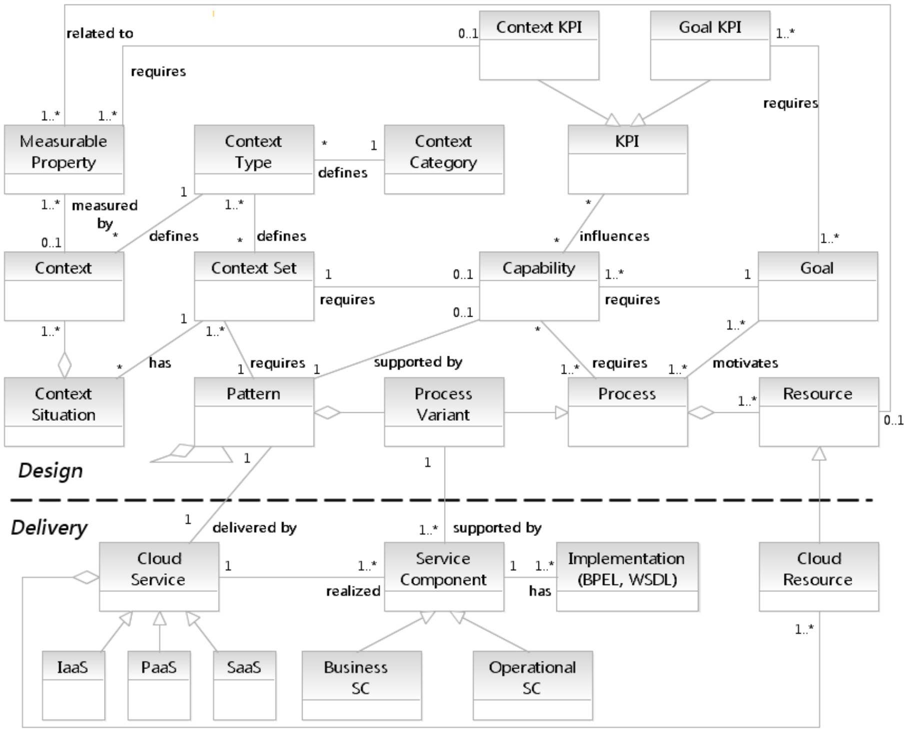

Fig. 1 An early version of the CaaS meta-model [29]

2.2 A brief description of the use-case enterprise

The use case was based on an enterprise that is one of the UK's leading Independent Alternative Investment promoters. For reasons of anonymity this will be referred henceforth as the “Enterprise”. The mission of the Enterprise is to help investors capitalize on their investments by striving for excellence with their investment options and offering the best customer service.

Being truly independent, the Enterprise is able to consider a diverse range of investment opportunities to meet their clients’ requirements. Whilst the Enterprise does hold agent agreements with certain investment providers to ensure access to their full range of products, it strives to ensure that its work ethic and selection process remains unbiased and client focused. The Enterprise’s commitment to its customers is to provide a friendly, professional service, deliver results that are consistent with their expectations and fall within the customers’ investment remit, handled by specialist staff to take the job in-hand.

The investments promoted are designed to provide the Enterprise’s clients with attractive income streams and investments with high potential for capital growth. Everyone’s needs and objectives are unique. This is why the Enterprise’s belief in tailor-made personal service lies at the heart of the Enterprise approach.

The Enterprise’s operational model is depicted in Fig. 2. The Enterprise operates as a connector between various investment products and distributors, who are responsible for contacting the investors either directly or through a sub – distributor.

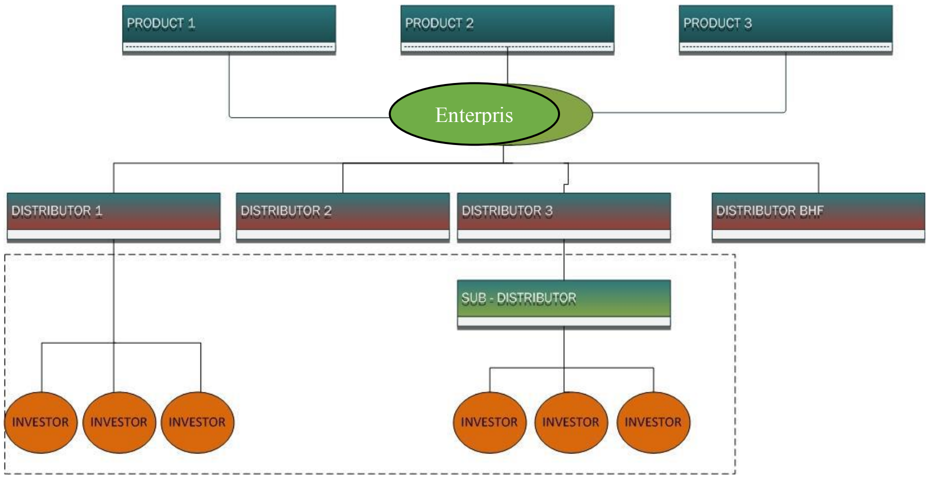

Fig. 2 The Enterprise's operational model

Regarding the specific use case, currently compliance procedures and pressures from industry bodies prevent the pension transfer process being managed through digital means. There are multiple compliance check points by all stakeholders (insurance companies, IFAs, introducers, agents, SIPP Trustees, Investment products) and all pension transfer documentation, SIPP establishment documentation and investment application documentation need to be manually signed by the client.

If the industry and compliance procedures allowed digital submissions of such documentation, the Enterprise would like to be capable of getting such documentation signed via a secure portal on the system either whilst speaking to a client face to face (iPad digital signature) or over the phone (computer digital signature). Subsequently all documentation to be automatically sent to all relevant parties for processing, ideally their systems also being automated so as to speed up the pension transfer and investment process.

In this case, capability-driven modelling is required in order to get all necessary documentation signed via a secure portal on the system either whilst speaking to a client face to face or over the phone and automatically sent to all relevant parties for processing. The overall goal is to design a thorough system that will enable compatibility within several different parties (insurance companies, IFAs, introducers, agents, SIPP Trustees, Investment products) regarding transfer and verification of all necessary documentation for cash and pension investments. In such a use case, the context may include the type of investment, type of product, the client's characteristics, requirements by insurance companies and / or agents.

In order to gather the necessary information from the enterprise based on a capability-driven methodology, the enterprise's goals were first identified and modelled. These goals were identified with respect to (i) the current capabilities of the enterprise, (ii) the envisioned capabilities for the near future and (iii) the context within which these capabilities would help towards the achievement of the main goals.

2.3 An overview of the 'reasoning cycle' approach to recording design decisions

The theoretical foundation of the design meta-process was based on the design reasoning framework introduced in [30] and used in various analyses of modelling techniques *c.f.* [35]. Briefly, the design reasoning cycle is depicted as in Fig. 3.

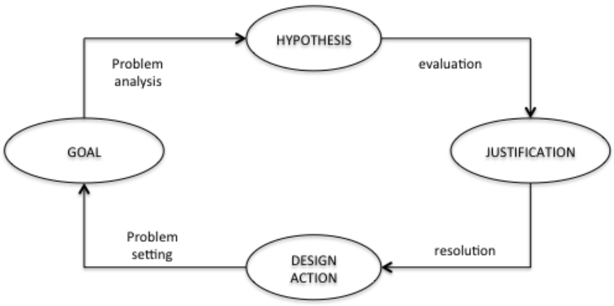

Fig. 3 Design reasoning cycle

The reasoning loop model for reflective design is a composition of a set of static primitives capturing the information content of the design process, and a set of operational primitives capturing its dynamic component. The former is concerned with *what* is done; the latter with *how* this is done. A synthesis of the two shows the interaction between process and substance. The following correspondences are identified:

- *Problem Setting* generates *Goals*. Designers set the objectives to be reached, the demands to be satisfied, the problems to be resolved. These include anything that may arise in the design process and requires effort in order to solve it.
- *Problem Analysis* generates *Hypotheses*. Designers analyse the problem domain to arrive at suggestions, proposals, or ideas about the resolution of identified problems.
- *Solutions Evaluation* generates *Justifications*. Designers produce claims on the status of the generated alternatives.
- *Problem Resolution* generates *Design Actions*, which alter design artefacts while generating and resolving Goals. Designers take actions affecting the objects used, produced or worked on during the design process.

The design activity is goal-driven, in that it stems from and aims at resolving some design goal. A Goal in this context is the designers' intentions with respect to the design process. The creation of new cycles results in the construction of a network of reasoning loops, each one of them derived to resolve a problem arising in the resolution of another problem.

Reasoning proceeds in a closed loop process, starting with the definition of a problem and closing with its resolution. The notion of a closed loop has two connotations. First, a closed loop is a sequence of steps that must be closed, in the sense of being completed; if it is not, the problem has not been solved. Second, like a programming loop, it is a sequence of steps that is used repeatedly throughout the design process and whose contents take up different values in each application.

4 Analysis and Results

Using the design reasoning cycle, each one of the two modelers described the design goals that were set during the meta-model's instantiation, and for each design goal they defined the relevant hypotheses, evaluated them based on specific justifications, and recorded the decisions reached for an action that would lead to another design goal.

4.1 Presentation of the modelling and the rationale behind the instantiations

The rationale behind the capability meta-model instantiations was captured in two levels for both modelers. The outcome for the 1st level of the rationale of the modelling is presented in Fig. 4 and Fig. 5. The two figures clearly depict the differences in the approaches used from the two modelers in order to instantiate the same use case. More specifically, for the 1st modeler the main goal of instantiating the use case, was analysed in three different sub –

goals, namely to decide the first item / entity to identify, to decide upon the way to proceed with the rest of the entities, and to identify them. On the other hand, the 2nd modeler followed a more analytical approach.

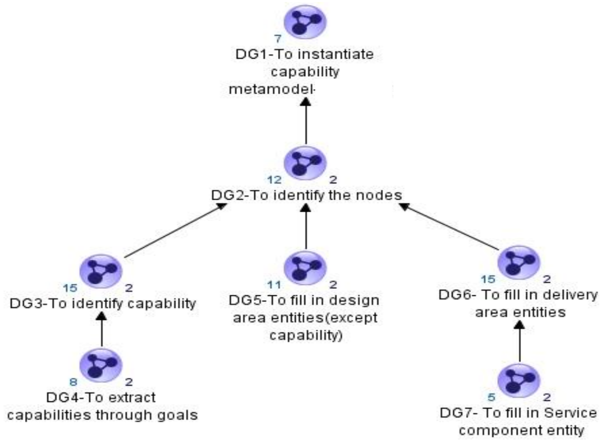

Fig. 4 Goal representation, 1st modeler

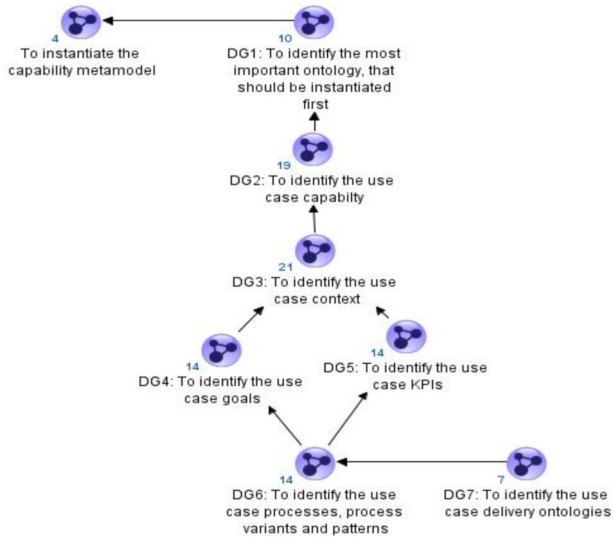

Fig. 5 Goal representation, 2nd modeler

The most important conclusions though may arise from the second level of analysis. For example, let us discuss about the rationale behind the decision of the first and most important entity to be identified; that is, the capability. The following figures depict the rationale behind the decisions of the two modelers. The rationale is based on a question – goal, the fulfilment of which is carried out through arguments and hypotheses. For instance, according to Fig. 6, the 1st modeler's goal was to identify the capability. The first hypothesis was to directly understand capability's definition, but this was rejected based on several arguments and the modeler was not able to provide a straightforward definition. Therefore, he formed another hypothesis, according to which the capability of the enterprise would be defined through its goals, and that led to a capability definition. On the other hand, according to Fig. 7 the 2nd modeler directly identified the use case's capability.

This difference in their approached resulted in different capability definitions and thus differences in the implementation of the same meta–model to the same use case scenario.

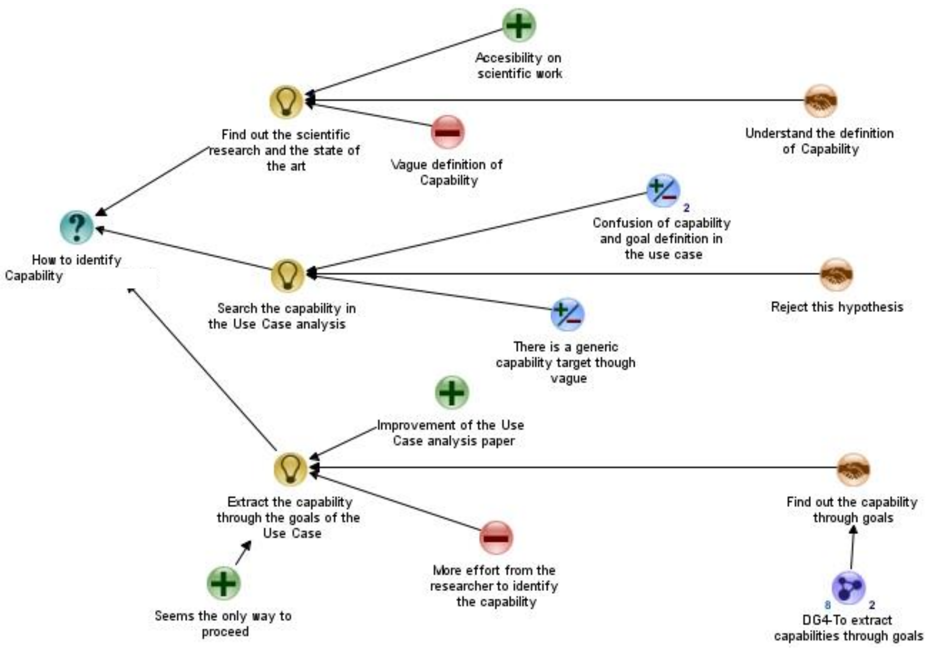

Fig. 6 Enterprise instantiation of 1st researcher, identify capability

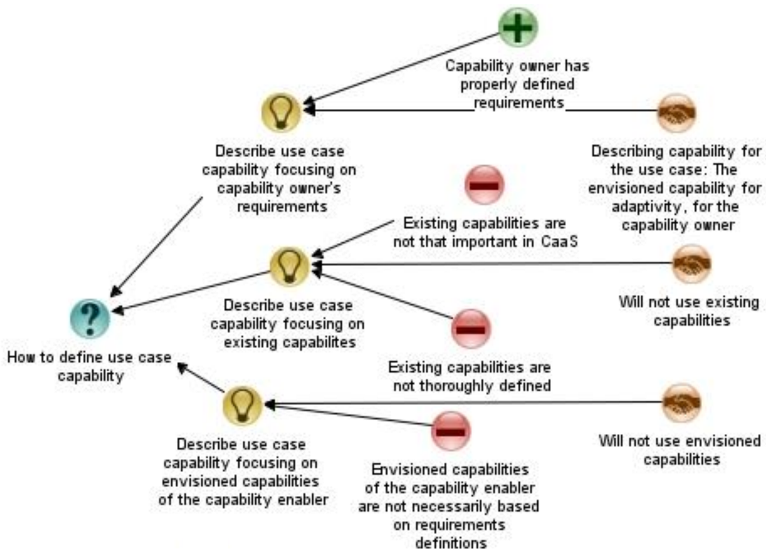

Fig. 7 Enterprise instantiation of 2nd researcher, identify capability

4.2 Presentation of the different analyses

The different analyses carried out by the two different modelers are summarized in Table 1. The summary is carried out with respect to the following key-concepts of the meta-model:

- Capability
- Context
- Goals

- Variability

Table 1 summarizes the way the two modelers dealt with these basic use case concepts throughout the modelling process. The main outcomes of the comparison are briefly described in the *Observation* column. According to the table, similar approaches in defining capability resulted in quite different definitions from the two modelers, while regarding context, both modelers pointed out its strong relationship with capability but once again their definitions were different. As far as the use case's goals are concerned, they were defined in the same way from both modelers, while variability was considered only by one of the two modelers.

Table 1 Treatment of concepts in the meta-model

Concept	*MODELER 1*	*MODELER 2*	*Observation*
Capability	No distinction made between envisioned capabilities of the capability owner and the capability enabler	Chose to define capability based on the capability owner's requirements	Similar approaches in defining capability resulted in quite different definitions.
Context	The context was defined according to the goals.	The use case's context was defined based on the changing context parameters and taking into account variability	For both researchers context is directly related to capability. The two modelers derived two slightly different context definitions
Goals	The goals were defined based on the capability definition	Goals were defined based on the context definition	Despite the difference in the priority given, goals were defined in the same way from both modelers.
Variability	Variability was not taken into account according to the design rationale analysis.	Variability was considered both in context type and in process variant definition	Different rationales regarding use case's variability affected the context definition. Variability may be considered in the meta model, as related both to context and processes.

Similarly, Table 2 summarizes the results regarding the approach used by each modeler regarding the process of modelling. Two separate process phases are investigated: the method guidance and the validation of the model quality. With respect to the method guidance, the main conclusion is that the two modelers were troubled about how to start the process, mainly due to lack of documentation. Thus, different method guidance between modelers finally resulted in different instantiations. Similarly, regarding the validation of the model quality, the lack of ontological definition resulted in alternative approaches even from the same modeler and for the same use case.

Table 2 Approach to the process of modelling

Process Phase	*MODELER 1*	*MODELER 2*	*Observation*
Method guidance	1. Was troubled about which node to start with 2. Capability was chosen as the key	1. Was troubled about which node to start with 2. Capability was chosen as the key entity. Capability was defined	Both modelers were troubled about how to start due to lack of documentation. Different method

Process Phase	*MODELER 1*	*MODELER 2*	*Observation*
	entity. Capability was defined first. 3. Goals were considered directly related to capability and were defined. 4. Context was defined after the use case's goals	first. 3. Context was defined based on capability. 4. Goals were defined based on context and the rest of design nodes were defined as well.	guidance between modelers finally resulted in different instantiations.
Validating model quality	Uncertainty on how to define entities and relationships between them	Uncertainty on how to define entities and relationships between them	Lack of ontological definition leads to alternative approaches even from the same modeler and for the same use case

4.3 Reflections on the use of the capability meta-model

Reflecting on the exercise of instantiating the meta-model and recording the design decisions made one can observe the issues summarized in Table 3.

Table 3 Main observations as a reflection of the procedure

No.	*Observation*
1	Different modelers approached capability and context through different perspectives and as a result they produced different outcomes based on the same use cases.
2	Capability and context are two of the key factors regarding the meta–model and CaaS in general, and should be precisely and accurately defined for every use case.
3	The meta–model validity should be investigated further and tested. The analytical design rationale outputs of two different researchers depict all the differences regarding both the goals set and the rationale that was used in order to achieve them.
4	A key issue that needs to be addressed is the ontological and formal definition of all the concepts in the meta-model. When discussing with users, it was completely subjective and informal how capability was defined. The approach depended almost entirely on the users' interpretation of what they considered as their capability. There is a need for enabling modelers to elicit from users definitions of their capabilities, based on questions that have specific orientation based on key characteristics of capability
5	The idea of variability appears to be closely related to context and by correspondence also to capability. However, in the meta-model the concept of variability is related only to process as a process variant
6	The models finally produced do not appear to have anything that can be regarded as being of 'capability-centric' and indeed one might argue that the models are not much more than enterprise models
7	The different modelling viewpoints (e.g. goal modelling, business process modelling, business rules modelling, context modelling, concept modelling,

No.	Observation
	etc.) need to be supported by modelling languages that have a clear grammar
8	Regarding the ‘designing process’, it was turned out that without some clear guidelines regarding phases, inputs, outputs, constraints etc. it is not possible to have a methodically robust approach

5 Discussion and Conclusions

This paper was motivated by the fact that the Capability-Driven Development (CDD) approach is expected to allow digital enterprises to exploit the notion of 'capability' as a means of both designing for services and operating with services. Capability-driven modelling has been in the centre of research efforts during the last few years, and a few formally defined capability-driven design approaches have been presented recently.

In that framework, the main consideration of this paper was the impact one of these approaches would have on design activities. The main question that was to be answered is whether such a meta-model could provide sufficient guidance for repeatable design activities by different designs working on the same problem. In order to identify that, a use case was exploited which involved capability modelling on the same application by different designers. Each approach was documented using design rationale techniques.

The two efforts were analysed and observations about the capability driven design activities were defined. According to the main results, it is interesting to point out that two researchers in several parts of the process used different rationales and thus were lead to different approaches and different results with respect to main entities’ definitions and representation.

The output of this work has provided feedback to enhancing the capability meta-model and consequently the capability driven design activities in a number of important ways. For example, different modelling viewpoints need to be supported by modelling languages that have a clear grammar. Namely, business process modelling should be based on BPMN and therefore subject to all the concept, rules and constraints defined in the BPMN meta-model; the concepts modelling could be based on UML Class Association Diagram, which again will be subject to UML meta-model. Finally, the goal modelling needs to be more than just ‘boxes and arrows’ since such a description would be entirely based on subjective and individualistic interpretations.

Thus, future work on this aspect would include an accurate and thorough definition of all supportive modelling languages required, towards a complete capability meta-model able to support CDD. In order to efficiently exploit such models it is essential for modelers to ensure that these models provide sufficient guidance for repeatable design activities by different designs working on the same problem, and most research in the near future should focus on that direction.

Acknowledgment. This work has been supported by the European Commission FP7 Project CaaS (611351).

References

1. Bosch, J. *From software product lines to software ecosystems*. in *13th International Software Product Line Conference*. 2009. pp. 111-119, Carnegie Mellon University.
2. Jarke, M., P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. Robinson, *The Brave New World of Design Requirements.* Information Systems, 2011. **36**(7): p. 992-1008.
3. ISTAG, *Software Technologies - The Missing Key Enabling Technology: Toward a Strategic Agenda for Software Technologies in Europe*, 2012, July 2012, ISTAG - Information Society Technologies Advisory Group: Brussels, Belgium.

4. Digital-Europe, *White Paper on 'Digital Europe's Vision 2020: A Transformational Agenda for the Digital Age'*, 2009, Digital Europe.
5. Sandkuhl, K. and U. Borchardt. *Towards an Approach for Context Modeling*. in *International Workshop on Advances in Services DEsign based on the Notion of CApabiliy (ASDENCA'14)*. 2014. pp., Thessaloniki, Greece: Springer.
6. Stirna, J., J. Grabis, M. Henkel, and J. Zdravkovic. *Capability Driven Development: An Approach to Support Evolving Organizations*. in *Practice of Enterprise Modelling (PoEM'12)*. 2012. pp. 117-131: Springer.
7. Danesh, M.H. and E. Yu. *Modeling Enterprise Capabilities with i*: Reasoning on Alternatives*. in *International Workshop on Advances in Services DEsign based on the Notion of CApabiliy (ASDENCA'14)*. 2014. pp., Thessaloniki, Greece: Springer.
8. Deneulin, S., *The Capability Approach and the Praxis of Development*. 2006: Palgrave.
9. Deneulin, S. and L. Shahani, eds. *An Introduction to the Human Development and Capability Approach*. 2009, Earthscan.
10. Deneulin, S., M. Nebel, and N. Sagovsky, eds. *Transforming Unjust Structures - The Capability Approach*. 2006, Springer.
11. Collis, J., *How Valuable are Organizational Capabilities?* Strategic Management Journal, 1994. **15**(Special issue on 'Competitive Organizational Behavior'): p. 143-152.
12. King, A.A. and C.L. Tucci, *Incumbent Entry into New Market Niches: The Role of Experience and Managerial Choice in the Creation of Dynamic Capabilities.* Management Science, 2002. **48**(2): p. 171-186.
13. Barney, J., *Firm Resources and Sustained Competitive Advantage.* Journal of Management, 1991. **17**: p. 99-120.
14. Helfat, C.E. and M.A. Peteraf, *The dynamic resource-based view: capability lifecycles.* Strategic Management Journal, 2003. **24**(997-1010).
15. Cohen, W.M. and D.A. Levinthal, *Absorptive Capacity: A New Perspective on Learning and Innovation.* Administrative Science Quarterly, 1990. **35**(1): p. 128-152.
16. Lane, P.J., B.R. Koka, and S. Pathak, *The Reification of Absorptive Capacity: A critical Review and Rejuvenation of the Construct.* Academy of Management Review, 2006. **31**(4): p. 833-863.
17. Akhtar, I., *Untangling Dynamic Capabilities : How They Change*, *RIBM Doctoral Symposium*, 2013, 14th-15th March.
18. Eisenhardt, K.M. and J.A. Martin, *Dynamic Capabilities: What are They?* Strategic Management Journal, 2000. **21**: p. 1105-1121.
19. Helfat, C.E., S. Finkelstein, W. Mitchell, M. Peteraf, H. Singh, D. Teece, and S.G. Winter, *Dynamic Capabilities- Foundations*, in *Dynamic Capabilities- Understanding Strategic Change in Organizations*. (ed.), 2007, Blackwell Publishing. p. 1-18.
20. Teece, D.J., *Dynamic Capabilities and Strategic Management*. 2009, United States, New York: Oxford University Press.
21. Teece, D.J., G. Pisano, and A. Shuen, *Dynamic Capabilities and Strategic Management.* Strategic Management Journal, 1997. **18**(7): p. 509-533.
22. Sambamurthy, V., A. Bharadwaj, and V. Grover, *Shaping Agility through Digital Options: Reconceptualizing the Role of IT in Contemporary Firms.* MIS Quarterly, 2003. **27**(2): p. 237-263.
23. Weill, P., M. Subramani, and M. Broadbent, *IT Infrastructure for Strategic Agility.* Sloan Management Review, 2002. **44**(1): p. 57-65.
24. OPENGROUP. *ArchiMate - Modelling Language for Enterprise Architecture, V2.0*. 2012; Available from: https://www2.opengroup.org/ogsys/catalog/c118.
25. OPENGROUP. *TOGAF - Enterprise Architecture Methodology, Version 9.1*. 2012; Available from: http://www.opengroup.org/togaf/.
26. Curtis, B., B. Hefley, and S. Miller, *People Capability Maturity Model (P-CMM) Version 2.0, Second Edition*, 2009, SEI Institute, Carnegie Mellon University.
27. Danesh, M.H. and E. Yu, *Modeling Dynamic Capabilities to Reason about Information Systems Flexibility*, *CAiSE 2014 (Submitted)*, 2014.
28. Ulrich, W. and M. Rosen, *The Business Capability Map: Building a Foundation for Business/IT Alignment*, 2012, Cutter Consortium for Business and Enterprise Architecture.
29. Zdravkovic, J., J. Stirna, and M. Henkel, *Modeling Business Capabilities and Context Dependent Delivery by Cloud Services*, *CAiSE 2013*, 2013: Valencia, Spain.

30. Louridas, P. and P. Loucopoulos, *A Generic Model for Reflective Design.* ACM Transactions on Software Engineering and Methodology, 2000. **9**(2): p. 199-237.
31. Buckingham Shum, S. *Augmenting Design Deliberation with Compendium: The Case of Collaborative Ontology Design*. in *ACM Hypertext Conference*. 2002. pp., Maryland.
32. Compendium-Institute. *Compendium*. 2008; Available from: http://compendiuminstitute.net/index.htm.
33. 33. Kavakli, V. and P. Loucopoulos, *Modelling of Organisational Change Using the EKD Framework.* Communications of the Association for Information Systems (http://cais.isworld.org), 1999.
34. 34. Bubenko Jr, J., A. Persson, and J. Stirna, *User Guide of the Knowledge Management Approach Using Enterprise Knowledge Patterns, deliverable D3, IST Programme project HyperKnowledge–Hypermedia and Pattern Based Knowledge Management for Smart Organisations, project no*, 2001, IST-2000-28401, Dept. of Computer and Systems Sciences, Royal Institute of Technology, Stockholm, Sweden, available on http://www. dsv. su. se/~ js/ekd_user_guide. html.
35. 35. Jansen, A.J., *The Goal Argumentation Method Analyzed*, 2013.