Towards a Flexible Solution in Knowledge-Based Service Organizations: Capability as a Service

Hasan Koç(✉), Kurt Sandkuhl, and Michael Fellmann

Chair of Business Information Systems, Institute of Computer Science, University of Rostock, Albert-Einstein-Str. 22, 18059 Rostock, Germany
{hasan.koc,kurt.sandkuhl, michael.fellmann}@uni-rostock.de

Abstract. To improve their chances of survival, enterprises need to cope with the challenges caused by today's dynamic markets, regulations, customer demands, novel technologies, etc. The shift towards a service-oriented economy as underlined by the concept of Service-Dominant (S-D) Logic is gaining traction within the Service Science community and industry. This shift towards service-dominated business models makes it even more important for enterprises to be agile to meet the rapidly changing customer demands. This demand for agility is reinforced in the context of knowledge-based service organizations (KBSOs), for which flexible service provision is a fundamental requirement. Against this background, we argue that the capability-driven design may provide the KBSOs the required degree of flexibility. Towards this goal, we present an approach for the capability-driven design of enterprises. More precisely, our contributions are (i) the introduction of a capability-based paradigm that contributes to the design of flexible services in KBSOs, (ii) an architecture of our service design and deployment environment and (iii) a demonstration of using our approach based on a real-world use case from an organization in the utilities industry. Our results and insights are based on our work in the EU-FP7 project "Capability as a Service in Digital Enterprises" (CaaS).

Keywords: Capability modelling · Service science · Context-aware services · Knowledge-based services · Context modelling · Capability as a service

1 Introduction

Service is a broadly applied term in Economics and Computer Science. The roots of service lie in division of labor and development of capabilities by various entities that compete and cooperate [1]. The Service Science field is a relatively young discipline investigating different facets of services. Service-Dominant (S-D) Logic focuses on the shift from the exchange of tangible goods toward the exchange of intangibles, specialized skills and knowledge, and processes and reflects the rise of the Service Economy [3]. Knowledge is an operant resource and it is the foundation of competitive advantage. Enterprises operating in this economy have to adapt their services frequently due to increased competition, changes in the market and technology. Capabilities are

T. Borangiu et al. (Eds.): IESS 2016, LNBIP 247, pp. 100–111, 2016.
DOI: 10.1007/978-3-319-32689-4_8

enablers of competitive advantage, they help companies to continuously deliver a certain business value in dynamically changing circumstances [4], which allows adaptation and dynamic design of services.

An important aspect in the Service Science research is the knowledge-based service organization (KBSO). Despite being defined in different ways, KBSOs are characterized with the need of higher flexibility in the service provision, i.e. the *ability to cope with specific needs, expectations and preferences of the clients* [5]. Vargo and Lusch define services as *the application of specialized competences (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself* [3] and Alter argues that capabilities reside in the “competences” part [6]. Based on these interpretations, the work focuses on how KBSOs can exploit their capabilities to deliver flexible services by applying the Capability-driven Development (CDD) method, which is being developed in an EU FP-7 research project “Capability as a Service” (CaaS).

The main contributions of the paper are (i) the introduction of a capability-based paradigm that contributes to the flexible service design in KBSOs, (ii) the architecture of service design and deployment environment and (iii) showcasing the approach in a use case from an organization in utilities industry.

The remainder of the paper is structured as follows. In Sect. 2, we analyze the literature in order to state the problem, report on research streams suggesting solutions towards flexible service provision and finally derive requirements for a possible solution. Against this background, in Sect. 3, the CaaS paradigm as well as its methodology, capability-driven development (CDD), are introduced. Following that, Sect. 4 presents a real-world use case in a KBSO to showcase our approach. Finally, Sect. 5 concludes the work and discusses our findings.

2 Problem Relevance and Related Work

Two of five foundations of service systems as studied by Spohrer et al. include an adaptive internal organization responding to the dynamic external environment and a dynamic configuration of resources [1]. This view is supported by Tohidi [7], which emphasizes the role of flexibility in today’s enterprises offering services. Flexibility in this case means having a higher degree of adaptability in responding to new demands of the market. Furthermore, Hachani, Gzara and Verjus [8] emphasize that enterprises operate in a changing environment, both the external and internal constraints have an influence on service delivery and remaining competitive is even a harder challenge.

Different research streams relate to the design of flexible services. Among these, Business Process Management (BPM) in general and context-aware business processes and application of business rules in particular are considered most relevant. BPM is increasingly used in cross-boundary organizational processes including unstructured components [9]. In order to reach a higher degree of flexibility, the first possible solution could be adding functionality to BPM languages and making them context-aware, as suggested in [10, 11]. The second possible solution could be extending the boundaries of business process modelling languages, such as using approaches for modelling process variants [12] and relating them to the business rules, for instance by using DROOLS [13].

Furthermore, declarative approaches for business process modeling could also be considered (e.g. [25, 26]). However, declarative process specifications are hard to create and read in contrast to imperative approaches, especially for non-modelling experts.

We have identified a number of publications that list the shortcomings of such solutions in KBSO such as [5, 9] and enriched them based on our experiences in the CaaS project [14, 15]. First of all, it is a challenging task to capture all relevant knowledge in a formal business process model since (i) the processes are complex and have many variants and (ii) organizations and their processes are evolving to adapt to the changes. Second, it is hard to focus on the flexible parts of the services in enterprise models that capture all possible variations. Third, in a KBSO, knowledge workers experience and tacit knowledge is vital for service provision and it is hard to capture this knowledge with business process models. Fourth and last, the solution approaches cannot close the gap between business requirements and technical implementations. This necessarily has a negative influence on the communication between the stakeholders on various levels, such as business analysts, solution engineers and operators.

The Computer Science literature adopts the black box view of services. Proposals in this field seem to focus on aspects related to data and control flow [16, 17]. Thus, the definitions put emphasis on technical services, such as web services and IT-infrastructures needed to realize them. They neglect internal details and how service is performed, which obviously is relevant from the business point of view [18]. We regard services as a socio-technical aspect of organizations rather than a purely technical aspect. Based on this view and the identified problems above, two main requirements that the solution should fulfill have been derived:

- REQ 1: Business services should be designed in an understandable way for the stakeholders, who do not necessarily have a deep IT knowledge. To facilitate understanding, the solution should benefit from enterprise modelling approaches and should not only cover the technical aspects of service delivery.
- REQ 2: To support different roles participating to service design and delivery, a method should be engineered that addresses the modelling of business services depending on the application context, enterprise goals and operational standards.

3 Capability as a Service: A Novel Paradigm

This section introduces an approach that contributes to the flexible design of IT-based business services by explicitly taking into account business goals and delivery contexts, which is being developed in the CaaS project.

The main objective of CaaS-project is the creation of an integrated approach consisting of a *method*, *tools* and *best practices* that enable enterprises to sense and take advantage of changes in business context. The CaaS method for capability-driven development (CDD) consists of various components addressing different modelling aspects, such as context modelling, business process modelling and pattern modelling. The method is supported by the CDD environment, i.e. the *Capability Design Tool* (CDT), the *Capability Context Platform* (CCP) and the *Capability Delivery Navigation Application* (CNA) (cf. Section 3.2). Three industrial cases from e-government, finance and business process outsourcing (BPO) serve as basis for developing and validating the CDD approach. Section 3.1 focuses on the methodological and design aspects of CDD whereas Sect. 3.2 presents the architecture of the environment.

3.1 Context-Aware Design of Services

The CDD follows a component-oriented method conceptualization, each addressing different modeling aspects (REQ 2). The *enterprise modeling* component captures strategic objectives related to the service in a transparent and measurable way. The component specifies the business services by using a model-based approach and represents them with a visual modeling language to support understandability (REQ 1). The *Context modeling* method component identifies potential application context(s) where the business service is supposed to be deployed. Finally, the *Pattern elicitation* component investigates reusable design-time or run-time elements. The method components are based on the CaaS meta-model [2], which is illustrated in Fig. 1 and described below.

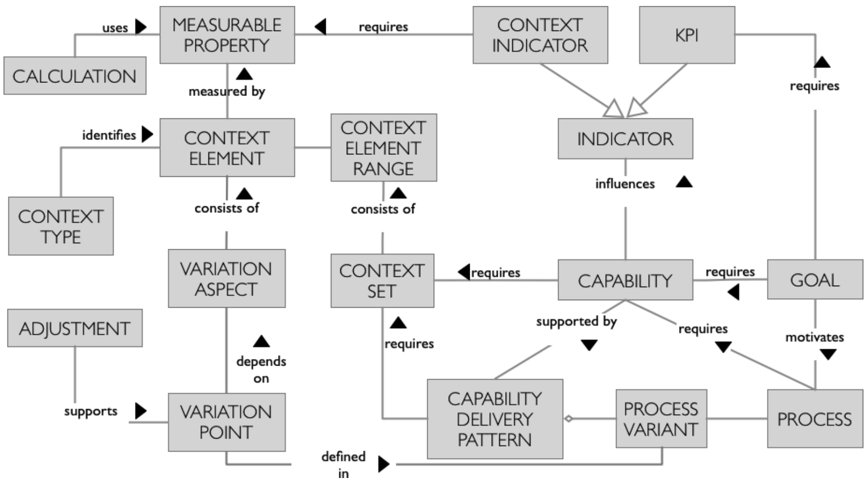

Fig. 1. Refined CDD meta-model based on [2]

Capability is defined as an ability to continuously deliver business value in dynamically changing circumstances (cf. [19]). In the literature, capabilities are perceived as fundamental abstraction instruments in business service design and they are related to the organizational strategies [4]. Vargo and Lusch [3] define services as *the application of specialized competences (knowledge and skills) through deeds, processes, and performances for the benefit of another entity or the entity itself*. Based on this definition, Alter [6] argues that capabilities reside in the "competences" part. Due to its root in strategic management, the notion of capability is a less technical-oriented concept and takes a business point of view. On the other side, services usually take a technical point of view and are concerned about the implementation aspects (cf. Sect. 2 and [17]). Unlike services, capabilities are perceived to be easier to link to the drivers and goals of the business and they are becoming a useful concept to business stakeholders [20]. This relates to the fact that it is still not clear how Service Science should address the managerial aspects of service-oriented technologies whereas capabilities provide an abstraction for business stakeholders on technical and IT-Services perspectives [21].

In terms of service design, capabilities enable flexible utilization of resources in various contexts and aim to support a model-based configuration of services. This part is represented in CDD meta-model with the concepts of *Context element* (the contextual factor), *Context element range* (configurable and relevant value ranges of a context element for the service at hand), *Context set* (a set of relevant ranges, which are monitored during runtime), *Measurable property* (attributes used to calculate the context value), *Variation point* (gateways that require contextual knowledge at runtime to be resolved) and *Adjustment* (adaptation of the service to the new situation by selecting an appropriate process variant). The detailed explanation of the CDD meta-model concepts can be found in [19].

3.2 The CDD Environment

In order to implement the CDD approach, a capability development and delivery environment was designed and deployed in the industrial cases of the CaaS project, which is depicted in Fig. 2. The numbers between the arrows are added to refer to the dependencies between the components in the following text.

Capability Design Tool (CDT). Provides modeling environment based on the CDD meta-model, i.e., capabilities can be modeled including business services (e.g., business process models), business goals, context and relations to patterns. The model designed in the CDT is uploaded to the CNA along with the adjustments code (1).

Capability Context Platform (CCP). Captures data from external data sources, such as sensing hardware and social networks (5). At runtime, CCP aggregates data and provides them to the CNA for further calculations (4). Also, at design time, the information about the context sources can be imported as new objects into CDT (2).

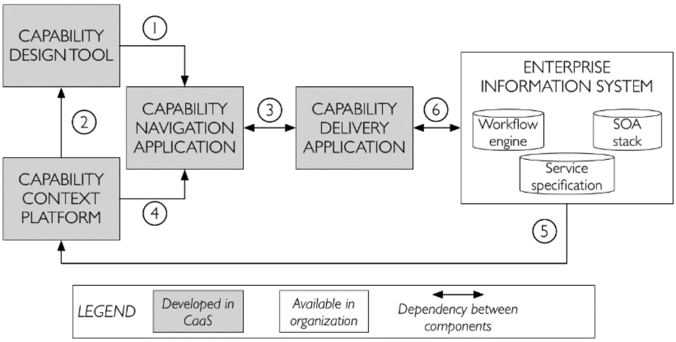

Fig. 2. The Architecture of the CDD Environment

Capability Navigation Application (CNA). Includes a module for monitoring context and KPIs. Moreover, the capability delivery adjustment algorithms are built-in here. The algorithms continuously evaluate necessary adjustments and pass capability delivery adjustment commands to the CDA for adapting the service to its application context (3).

Capability Delivery Application (CDA). An interface between the CNA and the Enterprise Information System of the organization in order to receive capability delivery adjustment commands from the CNA (3) and to provide the adjustment commands in line with the actual service application context (6).

4 Application of the CDD in a KBSO

4.1 BPO Use Case from Utilities Industry

This section reports from the application of the CDD method in SIV group, a vertically integrated enterprise from Rostock, Germany. The group specifically serves the utility industry and acts in the market mainly in two different roles. As an *ISV*, the group has a long-standing market presence in developing and selling the industry-specific ERP platform kVASy®, a software product that provides support functions specifically for the utility services industry.

As a *BSP*, the group uses the kVASy platform to offer digital services to utility companies. Within the European Union, the commodity markets are strictly regulated. Given the strict regulation and growing complexity, public utilities increasingly consider outsourcing of their business processes to external service providers. In this respect, the subsidiary of the SIV group, SIV Utility Services GmbH, is a KBSO and offers digital services to the players (market roles) of the energy sector running kVASy®.

The purpose of the use case is the transmission of energy consumption data from one market role to another. Energy data is exchanged in MSCONS format, which is a member of the EDIFACT specification family. Messages received by a market role are validated against an underlying informal data model and then the transmitted values are imported into kVASy®. Basically, this process includes a file-level check, a validation step and the processing of the individual meter readings. The exceptions in the former two steps can be remedied automatically by the ERP system.

However, the problems in the processing of the meter readings cannot be resolved without a manual intervention. For any exception, the BSP acts as a clearing center, i.e. having a direct access to the client's environment, a knowledge worker regularly checks the client's Business Activity Monitor (BAM) for failed MSCONS import processes. This causes organizational efforts, such as the arrangement of BSP's human resources schedule. Then, based on the contractual agreement between the BSP and the client, the responsible party clears the exceptional message. This decision depends on various factors such as the backlog size of the customer, message type that has thrown an exception or the type of the commodity, which are captured in the contractual agreement. The agreement as such cannot respond to customer demands, particularly when certain deadlines must apply to the message that are specified by the regulatory authority. The organization aims to design a flexible solution related to the routing of the exceptional cases. The envisioned solution introduced in the next section has to support a dynamic behavior in order to decide whether or not an individual clearing case has to be cleared by the client or by the BSP.

4.2 Dynamic and Flexible Clearing Services

SIV group has established business process models underlying the offered services, hence for the capability-based design of the clearing services, a slightly updated version of the process-first capability design method is applied [22]. The method consists of four components, each addressing important concepts (cf. Section 3.1), the notation to represent them as well as the sequence of activities (procedures) to be followed. The method components as well as their application in the use case are explained in the following.

Component 1– Define Scope. In order to design the capabilities, the method user first selects the service and sets the scope of the capability design. The selection can depend on various factors, such as optimizing the services with high process costs or managing services that are affected from the changes. The scope of the capability delivery is set to increase the throughput of MSCONS messages as well as the rate of automation by adjusting the service to the application context. To select the related services, the method proposes the approach introduced in [23], which basically classifies the services in a matrix based on four enhancers, “quality, time, flexibility, and cost”.

After consultation with domain experts and business service managers, the “exception clearing in market communication” service is selected, which should deliver “dynamic BSP clearing” as a capability. This capability envisions offering context-aware services to support flexible exchange of messages, where faulty processes must be cleared.

Component 2– Develop or Update Enterprise Models. This method component ensures that selected business process models are up-to-date and applies changes if required. Moreover, to check if business goals are satisfied during the capability delivery, KPIs are developed. If no goals model is available, they are developed based on the guidelines proposed in [24]. Since an alignment of the goals is required on the business service level, method user should rather model the capability related goals and not the enterprise objectives on a general basis. Finally, goals, business process models, KPIs and capabilities are related, which is used as input in the next method component. This method component uses BPMN 2.0 and 4EM Notation [24] to represent the important concepts, such as goals, processes and KPIs.

Clearing of the faulty cases is a knowledge-centric task, hence the BSP’s domain experts are predominantly *knowledge workers*. Nevertheless, we observed during the method application that the knowledge workers follow predefined *case handling instructions*, when clearing the messages. The handling instructions include service specifications and best-practices in service delivery in a textual form. This caused ambiguous definitions of the solution implementation as well as low degree of formality, hindering the envisioned flexible service provision. Using the guidelines provided by the method component, we created business process models of case handling instructions. Moreover, we developed the goals model by involving the domain experts in the modelling sessions. In the last step, we related goals, business process models and the "dynamic BSP clearing" capability, which should help in understanding the influence of the contextual factors to the processes.

Component 3 – Context Modelling. This method component models the context of the capability delivery, which is the potential application context of the deployed service. For this purpose, the designer executes three activities subsequently, "find variations, capture context element" and "design context", which are defined in the following.

Find Variations. Identifies the variability in the business process models and focuses on the reasons causing such variations. By further analyzing variability in the subsequent activity, the method user aims to elicit a context element. The activity also provides the method user with the guidelines on what constitutes a process variant and how to distinguish variability from standard decision points.

For the SIV use case, the variations in the business process models were analyzed. To do so, we used the guidelines provided by the method, (e.g. "*different than a decision point, a process variant is always relevant for capability delivery. For each decision point, evaluate the condition expression at the gateway and determine how the decision is met, i.e. data-based, event-based or context-based"*). As a result, 15 process variants were identified, which are being executed by the knowledge worker at BSP when clearing a message.

Capture Context Element. We argue that contextual information can stem from the factors of change, since they mainly cause variations in the service provision. Thus a substantial analysis of process variants underlying the service is required to capture a context element. The method proposes three guidelines for this activity, namely (i) to be classified as a context element, the change factor must be measurable, i.e. its value must be retrieved from an information system, (ii) context element is an external influence on the process, which should not exist as a process instance or data in the system and (iii) context elements are decisive for the resolution of variation points.

In the SIV use case, we investigated the factors influencing the case clearing of a knowledge worker. Alongside with the schedule of the knowledge workers in BSP, the clearing policy establishes the main factor for resolving the variation point, which is a gateway that is resolved in accordance with the service context. A clearing policy typically includes specifications about the message types, market roles, exception clearing type and critical backlog. Each time a knowledge worker clears the case, (s)he compares the actual values of these parameters with the values from the clearing policy. As such, these are external to the processes and thus selected as context elements, which are decisive for capability delivery (cf. Fig. 3).

Design Context. The activity defines ways to measure the context elements and to identify their relevant ranges, which are then put in a container (context set). The context set is linked to the goals and business process models, which is the capability

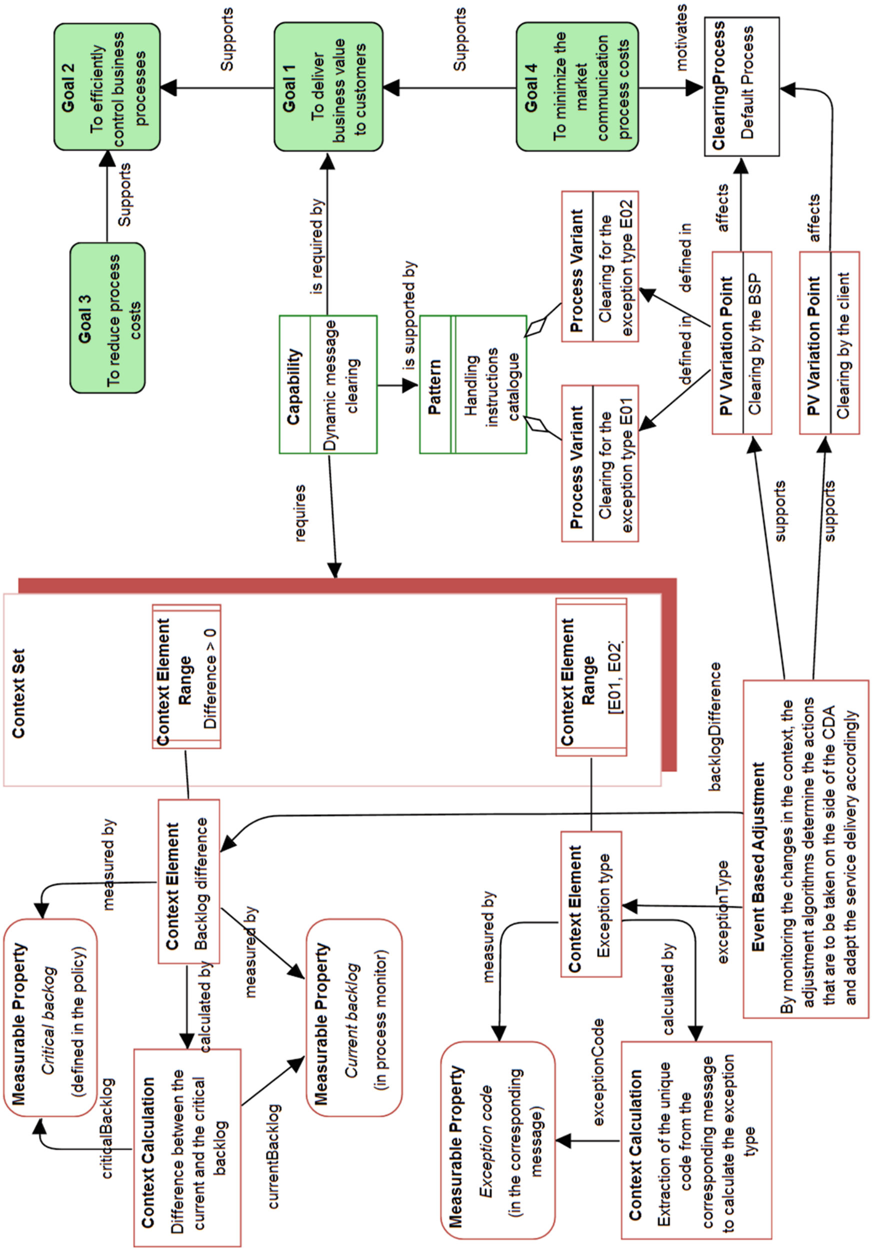

Fig. 3. Context-aware Business Service in a Capability Model

model of the selected service. In SIV use case, we conducted workshops and expert interviews to define the context element ranges. As the context elements have a common degree of similarity in each of the clients, we benefited from defining context elements only once and updating their valid ranges for different clients (cf. Fig. 3).

Component 4– Adjustments Modelling. Specifies the calculations of context values. Based on such calculations, the CNA adjusts the capability delivery in line with the service application context (cf. **Fig. 2**). After completing this method component, the capability model is now ready to be implemented, monitored and adjusted at runtime.

The developed capability model within the Capability Design Tool (CDT) is simplified and illustrated in Fig. 3. The right side of the model captures the organizational goals. The middle part represents the required business processes, reusable solutions and their specifications. Last but not least, left side of the model includes context modelling and adjustment modelling parts, i.e. which configurations and rules are necessary to adapt the service delivery to its application context and which processes are influenced by the changes.

5 Discussion and Outlook

Enterprises operate in changing environments and the shift towards a service-oriented economy based on S-D logic makes it even more important for enterprises to be agile. As stated in the 4th foundational premise of S-D logic, *knowledge is an operant resource* and *it is the foundation of competitive advantage* [3]. In this work, we focused on KBSOs as they require higher flexibility in the provision of services.

In terms of services notion, Computer Science literature adopts a rather technical view and neglects business needs. Necessarily, a solely technical view hinders reaching a common understanding in an enterprise, which is a prerequisite for offering flexible services. To close this gap, we used the notion of *capabilities*, which has its roots in strategic management and adopt a business perspective. Hence, this work contributes to context-aware business service modelling with a recent approach developed in an EU FP-7 research project “Capability as a Service” (CaaS).

The engineering of the CDD method was necessarily an iterative process and the method has undergone several versions, each of which has been evaluated by using different approaches. Currently, the industrial project partners prepare for a successful implementation of CaaS results, including the runtime environment as described in Sect. 3.2. During the application of the CDD method, we have observed that the capability-driven design may provide the KBSOs the required degree of flexibility in the following ways:

- By relating the business services to the application contexts and the goals of the enterprise to be reached, the notion of capability provides an abstraction from technical concepts of services.
- By reducing the gap between various stakeholders due to leveraging a model-based approach, i.e. the business analysts, solution engineers and knowledge workers understand what part of the service provision is flexible (context model), how they

are configured (adjustment model) and how it all contributes to the enterprise strategy (goals model). Moreover, due to the methodological support, the modelling endeavors can be documented systematically and made transparent for all stakeholder types.

- By representing the knowledge workers' tacit knowledge to a great extent in the capability models. Still, a complete automatization of the clearing procedures is not possible, which is a limitation of the approach.
- By not requiring changes in service provision to trace back the influenced business processes and updating them. Instead, the context model and the adjustments model are updated and the capability model is redeployed to the CNA (cf. arrow 1 in Fig. 2.). Hence, the number of process variants as well as the complexity of business process models are expected to diminish.

These merits of our approach should be validated by analyzing qualitative and quantitative data available when the CaaS results are implemented. Therefore, we are currently working on a final evaluation of the artefact. Another limitation is the modelling of the reusable solutions part, i.e. patterns modelling. In SIV use case, the case handling instructions deliver best-practices applicable to the clearing services offered to the clients. Although such instructions might include initial pattern candidates, we think that further analysis is required concerning this part of the method which will be part of our future work.

References

1. Spohrer, J., Golinelli, G.M., Piciocchi, P., et al.: An integrated SS-VSA analysis of changing job roles. Serv. Sci. **2**(1–2), 1–20 (2010)
2. 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, O. (eds.) CAiSE 2013. LNCS, vol. 7908, pp. 369–383. Springer, Heidelberg (2013)
3. Vargo, S.L., Lusch, R.F.: Evolving to a new dominant logic for marketing. J. Mark. **68**(1), 1–17 (2004)
4. Cutter Consortium. http://www.cutter.com/content-and-analysis/resource-centers/enterprise-architecture/sample-our-research/ea110504.html
5. Faria, J.A., Nóvoa, H.: An agile BPM system for knowledge-based service organizations. In: Nóvoa, H., Drăgoicea, M. (eds.) Exploring Services Science. LNBIP, vol. 201, pp. 65–79. Springer International Publishing, Cham (2015)
6. Alter, S.: Making a science of service systems practical: seeking usefulness and understandability while avoiding unnecessary assumptions and restrictions. In: Demirkan, H., Spohrer, J.C., Krishna, V. (eds.) The Science of Service Systems, pp. 61–72. Springer, Boston (2011)
7. Tohidi, H.: Modelling of business services in service oriented enterprises., Procedia Comput. Sci. **3**, 1147–1156 (2011). World Conference on Information Technology
8. Hachani, S., Gzara, L., Verjus, H.: Business process flexibility in service composition: experiment using a PLM-based scenario. In: van der Aalst, W.M.P., Mylopoulos, J., Rosemann, M., et al. (eds.) Exploring Services Science. LNBIP, vol. **82**, pp. 158–172. Springer, Berlin Heidelberg (2011)

9. Stavenko, Y., Kazantsev, N., Gromoff, A.: Business process model reasoning: from workflow to case management. In: CENTERIS 2013 - International Conference on Health and Social Care Information Systems and Technologies, Procedia Technology, vol. 9, pp. 806–811 (2013)
10. Rosemann, M., Recker, J., Flender, C.: Contextualisation of business processes. Int. J. Bus. Process Integr. Manage. **3**(1), 47–60 (2008)
11. van der Aalst, W.M.P.: Business process configuration in the cloud: how to support and analyze multi-tenant processes. In: 2011 IEEE 9th European Conference on Web Services (ECOWS), pp. 3–10. IEEE Computer Society Press, New York (2011)
12. Hallerbach, A., Bauer, T., Reichert, M.: Context-based configuration of process variants. In: 3rd International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008), pp. 31–40 (2008)
13. Browne, P.: JBoss Drools Business Rules: Capture, Automate, and Reuse Your Business Processes in a Clear English Language That Your Computer Can Understand. From Technologies to Solutions. Packt Pub, Birmingham (2009)
14. Bravos, G., Grabis, J., Henkel, M., Jokste, L., Kampars, J.: Supporting evolving organizations: IS development methodology goals. In: Johansson, B., Andersson, B., Holmberg, N. (eds.) BIR 2014. LNBIP, vol. 194, pp. 158–171. Springer, Heidelberg (2014)
15. Koç, Hasan, Sandkuhl, Kurt: A business process based method for capability modelling. In: Matulevičius, Raimundas, Dumas, Marlon (eds.) BIR 2015. LNBIP, vol. 229, pp. 257–264. Springer, Heidelberg (2015)
16. Ferrario, R., Guarino, N.: Towards an ontological foundation for services science. In: Domingue, J., Fensel, D., Traverso, P. (eds.) Future Internet - FIS 2008: First Future Internet Symposium, FIS 2008, pp. 152–169. Springer, Berlin (2009)
17. Polyvyanyy, A., Weske, M.: Flexible service systems. In: Demirkan, H., Spohrer, J.C., Krishna, V. (eds.) The Science of Service Systems, pp. 73–90. Springer, Boston (2011)
18. Ferrario, R., Guarino, N.: Commitment-based modeling of service systems. In: Snene, M. (ed.) Exploring Services Science. LNBIP, vol. 103, pp. 170–185. Springer, Heidelberg (2012)
19. Bērziša, S., Bravos, G., Gonzalez, T.C., et al.: Capability driven development: an approach to designing digital enterprises. Bus Inf Syst Eng. **57**(1), 15–25 (2015)
20. Antunes, G., Borbinha, J.: Capabilities in systems engineering: an overview. In: Cunha, J.F., Snene, M., Nóvoa, H. (eds.) Exploring Services Science. LNBIP, vol. 143, pp. 29–42. Springer, Berlin Heidelberg (2013)
21. Mikalef, P.: Developing IT-enabled dynamic capabilities: a service science approach. In: Johansson, B., Andersson, B., Holmberg, N. (eds.) BIR 2014. LNBIP, vol. 194, pp. 87–100. Springer, Heidelberg (2014)
22. España, S., Grabis, J., Henkel, M., Koç, H., Sandkuhl, K., Stirna, J., Zdravkovic, J.: Strategies for capability modelling: analysis based on initial experiences. In: Persson, A., Stirna, J. (eds.) CAiSE. LNBIP, vol. 215, pp. 40–52. Springer, Heidelberg (2015)
23. Andersson, B., Johannesson, P., Zdravkovic, J.: Aligning goals and services through goal and business modelling. Inf. Syst. E-Bus. Manage. **7**(2), 143–169 (2009)
24. Sandkuhl, K., Stirna, J., Persson, A.: Enterprise Modeling: Tackling Business Challenges with the 4EM Method. The Enterprise Engineering Series. Springer, Heidelberg (2014)
25. van der Aalst, W.M.P., Pesic, M.: A declarative approach for flexible business processes management. In: Eder, J., Dustdar, S. (eds.) BPM 2006 Workshops. LNCS, vol. 4103, pp. 169–180. Springer, Berlin Heidelberg (2006)
26. van der Aalst, W.M.P., Pesic, M.: DecSerFlow: towards a truly declarative service flow language. In: Bravetti, M., Nunez, M., Zavattaro, G. (eds.) Web Services and Formal Methods. LNCS, vol. 4148, pp. 1–23. Springer, Heidelberg (2006)