On the Applicability of Concepts from Variability Modelling in Capability Modelling: Experiences from a Case in Business Process Outsourcing

Kurt Sandkuhl and Hasan Koc

University of Rostock, Albert-Einstein-Straße 22,
18059 Rostock, Germany
{kurt.sandkuhl,hasan.koc}@uni-rostock.de

Abstract. Efficient and effective value creation and service delivery processes are considered as the key factor to competitiveness in a globalized market environment. Capability management contributes to this goal by considering an integrated view of the ability to deliver a certain service with the capacity to do so. In this paper, we focus on the aspect of variability in capabilities. Starting from an industrial case from business process outsourcing, we propose to introduce concepts from variability modelling, i.e. variation points and variation aspects, into modelling and representation of capabilities. The main contributions of this paper are the introduction of variability points into capability modelling, a proposal for further formalizing the term capability, and an industrial case showing the use of variability points.

Keywords: capability, variability modelling, variation point, enterprise modelling, business process outsourcing.

1 Introduction

In many industrial sectors, efficient and effective value creation and service delivery processes are considered as the key factor to competitiveness in a globalized market environment. Systematic management of enterprise architectures including the technical, application and business architecture is emerging into a key discipline in enterprises. One of the objectives of this discipline is to manage and systematically develop the capabilities of an enterprise, which often are reflected in the business services offered to customers and the technical services associated to them. In this context, networked enterprises [1], value networks [2] and extended enterprises [3] massively use service-oriented and process-oriented architectures.

The term capability is used in various industrial and academic contexts with often different meanings (see section 2.1 for a discussion). Most conceptualizations of the term agree that capability includes the ability to do something (know-how, organisational preparedness, appropriate competences) and the capacity for actual delivery in an application context. This indicates that flexibility, dynamics and variation are attributes associated with capability. In this paper, we focus on the aspect of variability in the context of capabilities and we propose to introduce

L. Iliadis, M. Papazoglou, and K. Pohl (Eds.): CAiSE 2014 Workshops, LNBIP 178, pp. 65–76, 2014.

variation points and variation aspects into modelling and representation of capabilities. For this purpose, a certain degree of formality is required in capability models, which will also be subject of the paper. The main contributions of this paper are the introduction of variability points into capability modelling, a proposal for further formalizing the term capability, and an industrial case showing the use of variability points.

The remaining part of the paper is structured as follows: Section 2 will give a brief overview to background for this work including existing capability modelling approaches, a meta-model proposed for formalizing capabilities and feature modelling. Section 3 presents an industrial case for motivating extension and formalization of capability definition. Section 4 introduces the concept of variation points in capability modelling including a formalization and an example from the industrial case illustrating the context of business service delivery together with the variation points. Finally, section 5 summarizes the work and discusses future work.

2 Background

This section summarizes the conceptual background including capability definitions and existing modelling approaches (2.1), variability modelling in product lines (2.3) and an analysis of improvement potential of a selected capability meta-model (2.2).

2.1 Capability Definitions

The term capability is used in different areas of business information systems. In the literature there seems to be an agreement about the characteristics of the capability, still there is no generally acceptance of the term. The definitions mainly put the focus on “combination of resources” [8], “capacity to execute an activity”[7], “perform better than competitors” [10] and “possessed ability [6]”.

The capabilities must be enablers of competitive advantage; they should help companies to continuously deliver a certain business value in dynamically changing circumstances [11]. They can be perceived from different organizational levels and thus utilized for different purposes. According to [12] the firm performance is the greatest, when the enterprises map their capabilities to IT applications. In this perspective the capabilities are provided as Business Services, i.e. they are designed and delivered in a process-oriented fashion. Capabilities as such are directly related to business processes that are affected from the changes in context such as regulations, customer preferences and system performance. As companies in rapidly changing environments need to anticipate variations and respond to them [9], the affected processes/ services need to be adjusted quickly. In other words the changes in context can be realized if the variations to the standard processes are promptly instantiated.

A capability definition is proposed by the EU-FP7 project Capabilities-as-a-Service in Digital Enterprises (CaaS)[1]. In the CaaS project capability is defined as *the*

[1] See `http:// http://caas-project.eu/`

ability and capacity that enable an enterprise to achieve a business goal in a certain context. Ability refers to the level of available competence, where competence is understood as talent intelligence and disposition, of a subject or enterprise to accomplish a goal; capacity means availability of resources, e.g. money, time, personnel, tools. This definition focuses on the components of enterprise modelling such as goal modelling and utilizes the notion of context, thus stresses the variations of the standard processes.

2.2 Improvement Needs in Existing Capability Meta-models

Since textual definitions leave room for ambiguity some capability modelling approaches define meta-models. Meta-models in general define the elements of a modelling language with their relationships and structural constraints. The meaning of relationships often is expressed by the name of the relationship, which leaves room for interpretations. An exception might be the taxonomic and aggregation relationships, which are available in many modelling languages and established with respect to their meaning. Some meta-models also include behavioural constraints to be observed during runtime, like the meta-model of MEMO [4]. However, even the existence of behavioural constraints does not fully specify how the model language elements have to be interpreted at runtime, i.e. the operational semantics (or runtime semantics) also should be defined, if the model is meant to be enactable.

In the area of capability modelling a recently published meta-model was proposed by Stirna et al. [13] together with the capability definition from the CaaS project presented above. Figure 1 depicts an excerpt from this meta-model, which shows all concepts directly related to Capability. The capability notion is the core element of this approach and related to other important components. In order to provide a capability some goals have to be fulfilled. These goals are operationalized via processes. As a result, capabilities require processes to be executed. During the capability delivery context indicators are measured in order to adjust the delivery to anticipated changes. This implies that the capability must be adequately delivered for certain context situations represented as context set. In order to react to the anticipated changes in context and to adjust the capability delivery process variations are used. These are modelled as specialisations of the processes.

Although this meta-model contains a quite detailed conceptualization of capability, some aspects need to be further specified in order to avoid ambiguities. Examples are:

- Each Capability "requires" at least one Process and exactly one ContextSet and one Goal. Semantically, this indicates that a capability cannot exist if there either is no Process or no Goal or no ContextSet, but it does not further specify the exact meaning of the relationship between Capability and these other concepts. From the textual definition of capability, it could be concluded that the ContextSet, the Process and the Goal specify the capability or are even part of it, but this cannot be derived from the meta-model.
- A Capability is supported by exactly one Pattern. Each Pattern is an aggregation of ProcessVariants, which in turn are specializations of Process. What is the

relation of the ProcessVariants of the Processes required for a Capability and the ProcessVariants aggregated in the one Pattern supporting a Capability? One interpretation would be that all ProcessVariants of all Processes required by a Capability have to be aggregated in the one Pattern required by the Capability. Other interpretations would be that only selected ProcessVariants or only one ProcessVariant per Process would be part of the one Pattern for a Capability.

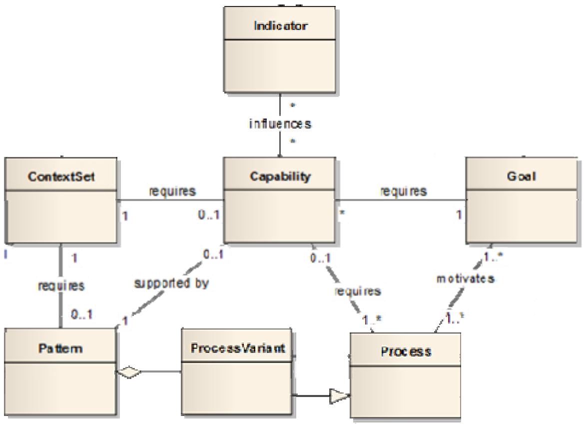

Fig. 1. Excerpt of the Capability Meta-Model introduced by [13]

- Not only Capability but also Process has a relation to Goal, which is that each process is motivated by one or many goals and that each goal motivates one ore many processes. Has the one Goal "requiring" a certain Capability to be among the Goals motivating the Processes required by the same Capability? Intuitively, this seems to be obvious, but it again is not expressed in the model.

The above examples show the general problem of most meta-models, the insufficient expressivity when it comes to runtime and behavioural characteristics also applies for the given meta-model. As a contribution and first step to address this issue, we propose to further specify the relations between Capability, Process and Context.

2.3 Variability Modelling in Product Line Engineering

Discussion of the term capability and existing meta-models in capability modeling indicates that capturing and representing variations in processes, business services or other elements of capability including the relationships or dependencies to context sets is an essential task in capability modeling. The area of variability modeling offers concepts how to deal with variability in complex systems, which might be applicable for capability modeling and will be briefly presented in this section.

Variability modeling offers an important contribution to limit the variety of the variants of systems by capturing and visualizing commonalities and dependencies between features and between the components providing feature implementations.

Since more than a decade, variability is frequently used in the area of technical systems and as element of software product line implementations. Among the variability modeling approaches, feature models are considered as in particular promising. The purpose of a feature model is to extract, structure and visualize the commonality and variability of a domain or set of products. Commonalities are the properties of products that are shared among all the products in a set, placing the products in the same category or family. Variability are the elements of the products that differentiate and show the configuration options, choices and variation points that are possible between variants of the product, aimed to satisfy customer needs and requirements. The variability and commonality is modelled as features and organized into a hierarchy of features and sub features, sometimes called feature tree, in the feature model. The hierarchy and other properties of the feature model are visualized in a feature diagram. Feature diagrams express the relation between features with the relation types mandatory, optional, alternative, required and mutually-exclusive. The exact syntax of feature diagrams is explained in [5].

3 Industrial Case

Work in this paper is motivated by an industrial case originating from the EU-FP7 project "Capability-as-a-Service in Digital Enterprises (CaaS)". This section introduces the case with its general characteristics (3.1) and shows an example for a business service offered (3.2)

3.1 Business Process Outsourcing of Energy Distribution Companies

SIV.AG from Rostock (Germany) offers business process outsourcing services to a variety of medium-sized utility providers and other market roles of the energy sector in Germany, Bulgaria, Macedonia and other European countries.

Energy distribution companies are facing a continuously changing business environment due to new regulations and bylaws from regulating authorities and due to competitors implementing innovative technical solutions in grid operations or metering services. In this context, both the business processes in organizations and information systems supporting these processes need to be quickly adaptive to changing organizational needs.

Business process outsourcing, i.e. the performance of a complete business process for a business function by a service provider outside an organization, has to offer and implement solutions for different cases. One variation is inherent in the business process as such. Even though core processes can be defined and implemented in standard software systems, configurations and adjustments for the organization in question are needed. The second cause of variation is the configuration for the country of use, which not only includes the usual localization, but also includes implementation of the actual regulations and bylaws. The third variation is related to the resource use for implementing the actual business process for the customer, i.e. the provision of technical and organizational capacities and capabilities.

3.2 Business Service MSCONS

The MSCONS (Metered Service Consumption Report Message) use case is viewed from a global perspective. The purpose of the global process in MSCONS use case is the transmission of energy consumption data from one market role to another role. By regulatory requirement, all data must be sent by e-mail and its format must comply with the international UN/EDIFACT standard. In addition to this requirement, national variants of the EDIFACT standard may exist that add further constraints to the syntactical structure of exchanged messages. Thus, messages must not only comply with the international but also with the national EDIFACT standard, which are subject to periodical change by the regulatory authorities, with usually two releases per year.

The process is triggered with a received MSCONS message after which the first syntax check happens. The second check is the examination of model error. In this task the rules force the declaration of a unique transaction ID between communicating partners. If there is no model error, the messages are classified. After this, a processability error check per message is performed. Messages may be invalid though syntactically correct. An invalid message causes an exception to be thrown. Currently, all of these exceptions are treated manually, involving the role of a knowledge worker. In the future it is possible to offer dynamic capabilities that routes the exception handling processes depending on the context in which the exception is thrown (see also section 4.2). If the message is processable, then the reading reason has to be determined since the MSCONS message is triggered due to a change of meter, installation of meter or period meter reading. Each reading reason has specific processes, still some components are recurring. After all messages are processed, they have to be archived. Fig. 2 illustrates only the "happy path" in the process of MSCONS Validation excluding the error conditions. For the sake of brevity the activities specifying which tasks should be executed when exceptions occur are omitted from the use case description and model.

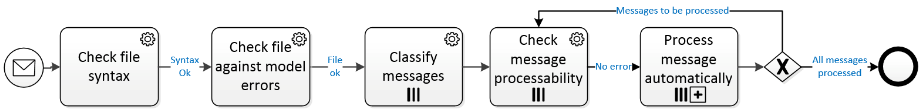

Fig. 2. Process of MSCONS Validation

4 Variation Points for Capability Modelling

This section contains a proposal how to apply concepts from variability modelling, i.e. variation points and variation aspects, for modelling capabilities. At the same time, a formalization of relationships between the core concepts of a capability is proposed (section 4.1) which extends what is defined in the meta-model introduced in section 2. Section 4.2 and 4.3 show the use of variation points and aspects in the industrial case; section 4.4 discusses initial experiences.

4.1 Variation Points and Variation Aspects in Capability Models

As discussed in section 2.2, a further formalization of the term capability and its relations to related concepts is supposed to reduce ambiguities and improve preciseness. At the same time, variability is considered an important aspect in capability modelling and management. This is why we will focus our proposal for a more detailed formalization on aspects related to variability. The basic idea of our approach is inspired by variability modelling in general and feature modelling in particular (see section 2.3). We propose to identify those aspects in business services of an enterprises where alternative flows, functions or procedures are possible and to identify cause and type of variations. For this purpose, we introduce *variation aspects* as the cause of variations and *variation points* as the locations of the variations in the business service model. Variations in behavior, functionality or content can be caused by different aspects, like performance indicators, exception types or information input. Since the context of a capability and its elements already have been introduced as characteristics of adapting capabilities, our approach is that these variation aspects correspond to context elements. Variation aspects can be relevant for different business services and at different positions in the business service model. A variation point identifies the business service model element where a variation with respect to a specific variation aspect occurs.

The formalization introduced in the following reflects the above ideas and starts with defining capability, context, business service, pattern and variant. Based on this definition, we introduce variation aspects and variation points and interlink them. Regarding the variants, our proposal is to also consider the fact, that variants can be composed of different alternating, optional or mandatory sub-variants. Decomposing variants in such a way will ease adaptation to different context.

A capability structure is a tuple $Cap := \{B, C, P, V, BT, CT, PT, VAss, VS, type\}$, consisting of

- disjoint sets B and BT whose elements are called business service model elements and business service model element types respectively; and a function $type_B : BT \to B$, that assigns a type $bt_i \in BT$ to an element $b_i \in B$ We assume that an enterprise's services offered to their customers can be modelled. B represents the different elements of such a model and BT the types used. The inner structure of the model could be further specified, for example using OMG's MOF. We by intention do not further specify the content of B since this is not required for our approach. B could be a business process model with attached web services or a software specification in UML, to just name two examples. With respect to the meta-model from section 2.1, B represents the Process concept.
- disjoint sets C, and CT whose elements are called context elements and context element types, respectively; and a function $type_C : CT \to C$, that assigns a type $ct_i \in CT$ to each $c_i \in C$. C and CT represent the ContextSet concept from the meta-model.

- disjoint sets P, and PT whose elements are called pattern elements and pattern element types, respectively, with $PT \subseteq BT$ and a function $type_P : PT \rightarrow P$, that assigns a type $pt_i \in PT$ to each $p_i \in P$. P and PT represent the Pattern concept from the meta-model.
- a set of variants V with $\forall v_i \in V : v_i \subset B \vee v_i \subset P$. V represents the ProcessVariant concept from the meta-model.

Based on the above capability structure, we define a variability assignment as a tuple $VAss := \{VA, VP, VPT, R, \mathrm{var}\}$, consisting of

- Sets of variation aspect VA with $VA \subseteq C$ which means that each $va_i \in VA$ consists of one or several $c_i \in C$.
- Sets of variation point types VPT with $VPT \subseteq PT$ and set of variation points VP with $VP \subseteq P$ and $\forall vp_i \in VP : type_V(vp_i) \in VPT$
- Set of relations R and a function $\mathrm{var} : R \mapsto VA \times VP$ with $\forall vp_i \in VP : \exists va_i \in VA : \mathrm{var}(R)(va_i, vp_i)$. This function relates variation aspects from the context to variation points in a pattern or variant. With var(R) = (va_i,vp_j) we define vp_j as a variation dependant on va_i.

Furthermore, we define a variability specification as a tuple $VS := \{V, R, man, opt, alt, req, excl\}$, consisting of

- the variation set V introduced above and a set R whose elements are called relations; V and R are disjoined sets.
- A function $man : R \mapsto V \times V$ that relates mandatory variants. With man(R) = (V_1,V_2) we define V_2 as a mandatory sub-variant of V_1.
- A function $opt : R \mapsto V \times V$ that relates optional variants. With opt(R) = (V_1,V_2) we define V_2 as an optional sub-variant of V_1.
- A function $alt : R \mapsto V \times V$ that relates alternative variants. With alt(R) = (V_1,V_2) we define V_2 as an alternative sub-variant of V_1.
- A function $req : R \mapsto V \times V$ that relates required features. With req(R) = (V_1,V_2) we define V_2 as a required variant for V_1.
- A function $excl : R \mapsto V \times V$ that relates mutually-exclusive features. With excl(R) = (V_1,V_2) we define V_2 is mutual-exclusive to V_1.

The functions are inspired by the relation types used in feature modeling for expressing relations between features (see section 2.3).

4.2 Context of Business Service Delivery

Context is a term that is used in many domains of computer science like artificial intelligence, operating systems, software engineering, databases, knowledge representation etc. The concept of context is also adapted by different disciplines other than computer sciences such as cognitive or social sciences. Thus the various definitions of context arise due to its widespread use. According to the framework of context use, the definitions and characteristics vary [14]. It is important that one should speak of the context in reference to its use [15], since there is no real consensual definition. Hence interpretation of context depends on the field of knowledge that it belongs to [16]. In accordance with [17] context is defined in this work as "any information that can be used to characterize the situation of any entity".

In the business process outsourcing use case described above, it is possible to identify context sets that consist of different context elements. These context elements define variation aspects introduced in section 4.1 and are related to the variation points in the process models (see section 4.3). For instance in order to execute the process "check file syntax" properly, the information about the country where the company operates is needed so that the appropriate service of the application is activated and parses the message. In addition other information like the role of the issuer and the addressee, the type of the message, the message version as well as the energy commodity needs to be acquired to "classify messages" before checking their processability (see also Fig. 2). The affects of the context elements to the process execution are demonstrated in the following section.

The values of such context elements form a context set, which is required to realize the capability "MSCONS processes supporting automated validation & exception handling". This capability is required to ensure correct exchange of messages between market roles. Different context sets arise due to various constellations of context elements. Since these context sets require the adoption of process variations, they are the main causes of process variability. This list of context elements derived from the use case and their ranges are illustrated in Table 1.

Table 1. Context Elements as Causes of Variability

ContextElement	**Range**
Country	{EU, Non EU}
Role	{Grid Operator, Balance Supplier}
Service Contract	{Types of Exceptions, Backlog size}
Application Reference	{LG, EM, VL, TL}
Process Execution	{Cloud, Customer, SIV}
Commodity	{Gas, Electric, Water}
Message Type	{MSCONS, UTILMD}
Message Version	{2.2a, 2.2b, 5.0, 5.1}
Exception Handling	{Routine, Knowledge Worker}

4.3 Variation Points in the Industrial Case

This section aims at applying the idea of variation points and variation aspects to the industrial case presented in section 3. Since the business services in the case are defined and modelled in a process-centred way, we will consider processes and process variants instead of business services in general.

Process variations are used to react to the anticipated changes in context and to adjust the capability delivery process as the following scenario demonstrates: the system imports an MSCONS message sent from the market role “grid operator” to another market role “balance supplier” with the message type “change of balancing area”. In this case the application does not execute the standard process that changes the balancing area, but instead it changes the tariff that the customer uses. If the same message type was sent from the balance supplier to grid operator, then the standard process that changes the balancing area had to be executed. Thus two context elements “market role” and “message type” form the context set “CS1”. The context set information is applied to complex gateway in Fig. 3.

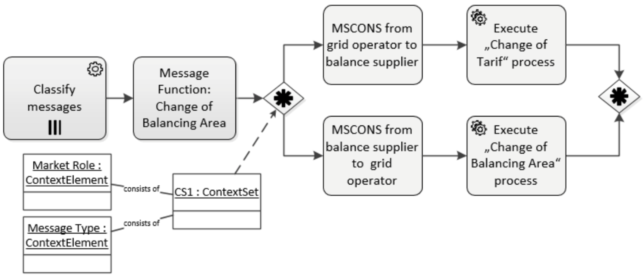

Fig. 3. Variation Points and Context Set

Regarding the formalization of capability structure, following observations were made. The business service model elements (B) are activities in MSCONS Validation processes. The context elements (C) that are required to realize the capability are listed in Table 1. In accordance with the actual values of context elements, different process variants (V) can be executed to deliver the capability. In order to deliver the capability, the variability should be assigned ($VAss$) by applying patterns (P). The variability assignment was also defined as a tuple. The variation points (VP) are those activities in the process model directly affected by the context set, hence they are related to one or more context elements. Moreover, the context set defines the variation aspects (VA). In this example we have a variation aspect of a “process execution” depending on context elements like roles and message types. This shows that the variation aspects are related to the variability points via context. Last but not least, different relations between process variants can be specified (VS), e.g. the process variants in Fig. 3 are mutually exclusive, since only one of them can occur at a time.

4.4 Discussion

The application of variation points and variations aspects in the industrial case confirmed that these concepts originating from feature modelling in principle can be transferred to capability modelling. Variation aspects, which in feature modelling are the characteristics deciding about mandatory, optional, mutually-exclusive or required features correspond to context elements in capability modelling; variation points in feature models are the different feature nodes which corresponds to business service model elements in capability models, e.g. activities in the process models of the industrial use case. However, the utility of variant hierarchies from feature modelling intuitively makes sense, but still has to be investigated in capability modelling.

The formalization of the relations between context, process, pattern, variant and capability resolves some of the ambiguities discussed in section 2, but not all of them. The definition of operational semantics still is required if capability models are meant to transformed to executable or enactable representations. The way of formalization presented in section 4.1 is not the only possibility. Using OCL constraints or other formal languages as annotations to the existing relationships between concepts in combination with further refinement of the concepts' attributes would be another option.

Although the formalization does not assume a process-oriented perspective on business services, the practical use with other business service representations, e.g. in functional or declarative paradigms, will have to be investigated in future work, since the industrial case only confirmed the use process-oriented business services.

5 Summary

New situations in business environments arise due to changes in regulations, bylaws and customer preferences. The capabilities help companies to continuously deliver a certain business value in these dynamically changing circumstances by adjusting the service delivery to different contexts. This paper focuses on the variation aspects of the business services which are triggered by the changes in context. As a starting point the capability meta-model introduced in [13] is taken.

In section 2 the relations between the concepts concerning the capability are discussed. These discussions addressed the need to contribute to capability modelling by specifying the relations of the meta-model components, particularly Capability, Process and Context. At the same time, we suggested that variability is considered as important aspect in capability modelling and introduced an example from an industrial use case. The variation aspects are mainly reflected on the business processes since the use case is modelled in a process-centred way. In the future, the relation between variation aspects and different paradigms such as service-oriented-architectures can be further investigated.

Another contribution of this work is an initial formalization approach that is proposed to increase preciseness of the term capability and to specify how the term is related to the aspects in the meta-model concerning variability. In that way we aim to avoid ambiguities identified in the underlying capability meta-model and enumerated in section 2.2. Future challenges may include attempts to specify or extend the formal capability definition to emphasize the semantics of associations between the capability components.

Acknowledgments. This work has been performed in the EU-FP7 funded project no. 611351 CaaS - Capability as a Service in Digital Enterprises.

References

1. Gunasekaran, A., McGaughey, R., Wolstencroft, V.: Agile Manufacturing: Concepts and Framework. In: Gunasekaran, A. (ed.) Agile Manufacturing: The 21st Century Competitive Strategy, pp. 25–49. Elsevier (2001)
2. Tapscott, D., Lowy, A., Ticoll, D.: Digital capital. Harnessing the power of business webs. Harvard Business School Press, Boston (2000)
3. Fong, M.W.L.: E-collaborations and virtual organizations. IRM Press, Hershey (2005)
4. Frank, U.: Multi-perspective enterprise modeling: foundational concepts, prospects and future research challenges. Softw. Syst. Model. (2012), doi:10.1007/s10270-012-0273-9
5. Thörn, C., Sandkuhl, K.: Feature Modeling: Managing Variability in Complex Systems. In: Tolk, A., Jain, L.C. (eds.) Complex systems in knowledge-based environments. SCI, vol. 168, pp. 129–162. Springer, Heidelberg (2009)
6. Open Group Standard TOGAF Version 9.1, `http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap03.html` (accessed March 06, 2014)
7. Jiang, Y., Zhao, J.: An empirical research of the forming process of Firm inter-organizational e-business capability: Based on the supply chain processes. In: 2010 2nd International Conference on Information Science and Engineering (ICISE), pp. 2603–2606 (2010)
8. Antunes, G., Barateiro, J., Becker, C., et al.: Modeling Contextual Concerns in Enterprise Architecture. In: 2011 15th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW), pp. 3–10 (2011)
9. Eriksson, T.: Processes, antecedents and outcomes of dynamic capabilities. Scandinavian Journal of Management (2013), doi: 10.1016/j.scaman, 05.001
10. Boonpattarakan, A.: Model of Thai Small and Medium Sized Enterprises' Organizational Capabilities: Review and Verification. JMR 4(3) (2012), doi: 10.5296/jmr.v4i3.1557
11. 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)
12. Chen, J., Tsou, H.: Performance effects of 5IT6 capability, service process innovation, and the mediating role of customer service. Journal of Engineering and Technology Management 29(1), 71–94 (2012), doi:10.1016/j.jengtecman.2011.09.007
13. 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)
14. Ben Mena, T., Bellamine-Ben Saoud, N., Ben Ahmed, M., Pavard, B.: Towards a Methodology for Context Sensitive Systems Development. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) CONTEXT 2007. LNCS (LNAI), vol. 4635, pp. 56–68. Springer, Heidelberg (2007)
15. Brézillon, P., Cavalcanti, M.: Modeling and Using Context. Knowl. Eng. Rev. 13(2), 185–194 (1998), doi:10.1017/S0269888998004044
16. Bazire, M., Brézillon, P.: Understanding Context Before Using It. In: Dey, A.K., Kokinov, B., Leake, D.B., Turner, R. (eds.) CONTEXT 2005. LNCS (LNAI), vol. 3554, pp. 29–40. Springer, Heidelberg (2005)
17. Dey, A.K.: Understanding and using context. Personal and Ubiquitous Computing 5(1), 4–7 (2001)