A Reference Model Based Design of Supply Chain Management Capabilities

Jānis Grabis, Solvita Bērziša

Information Technology Institute, Riga Technical University, Kalku 1, Riga, Latvia
{grabis, solvita.berzisa}@rtu.lv

Abstract. Capabilities define competitive advantages an organization possesses, and attaining desired capabilities is a challenging task. This paper proposes to use reference models as a basis for the capability design, and it focuses on usage of the SCOR model for designing supply chain management capabilities. The paper outlines a method for the reference model based capability design. The method relies on the correspondence among concepts used in capability modeling and concepts used in the SCOR model. The capability is designed by selecting and combining appropriate process categories and metrics from the reference model. Best practices defined in the referenced model are packaged as capability delivery patterns and are also used in capability design as reusable process fragments. The reference model provides sound foundations for the capability design while capability-oriented view of the reference model enriches it with contextual information.

Key words: Capability, reference model, supply chain, context

1 Introduction

Capabilities describe abilities possessed by an organization [1]. UPDM [2] specifically emphasizes that this ability should be achieved under specified performance requirements and operating conditions. This way capability delivery differs from traditional business services by explicitly taking into account delivery objectives and delivery circumstances. Capabilities can be designed to account for variations in the delivery circumstances and adjusted to improve their delivery performance [3].

Reference models provide a common framework for exploring some phenomena [4]. The SCOR model used in supply chain management [5] is one of the most widely investigated reference models. This model defines core supply chain management processes along with performance measure for supply chain evaluation and best practices for improving the supply chain processes.

There are several investigations on using the SCOR model in enterprise design. The SCOR process models have been used as building blocks for supply chain design [6]. In order to analyze alignment of business processes and information systems, the SCOR model is extended to improve representation of information flows [7]. Medini and Bourey [8] elaborate a methodology for SCOR-based enterprise architecting. The methodology includes steps of IT alignment, As-is analysis, Target top level modeling,

Target sub-level modeling and optimization. The enterprise architecture is created by combing capability maps with the reference model. Applicability of the reference models is hindered by lack of formality [9]. The authors also show that a formalized reference model can be used for rapid supply chain configuration. The SCOR model is also useful for developing supply chain integration solutions [10].

In this paper, it is argued that the reference models and the SCOR model in particular also could be useful source of information for designing capabilities because the reference processes provide an overall solution for achieving the desired capability. The capability design involves specification of the desired organizational capability as well as identification of means for capability delivery in various context situations. Reuse of existing solutions is essential to reduce complexity and control variability. In order to use the reference model in capability design, semantic difference between the SCOR model and capability representation should be addressed and a method for selection of design artifacts from the reference model should be elaborated.

The paper describes an initial proposal for using the SCOR reference model in capability design. The objective of the paper is to identify communalities among the reference model and capability design and to outline the reference model based capability design. The paper contributes both to the fields of capability management and SCOR model development. The SCOR model provides a well-formed basis for designing capabilities. The capability management view of the SCOR model enriches supply chain processes with contextual information.

The rest of the paper is organized as follows. Section 2 briefly discusses capability modeling and structure of the SCOR meta-model. The correspondence between both models is established in Section 3. Steps of the reference-model based capability design are described in Section 4. The capability design example is discussed in Section 5. Section 6 concludes.

2 Analysis Framework

The reference model based capability design is based on an observation that the SCOR model definition template share many similarities with the way capabilities are defined. In order to established foundations for further investigation, this section recaps the capability modeling meta-model adopted from [11] and general structure of the SCOR model.

2.1 Capability Modeling

The capabilities are modeled using concepts defined in the capability model. The simplified version of the capability meta-model is given in Fig. 1. Every capability has goals and achievement of these goals is measured by indicators. The indictors actually can be used as feedback to influence the way capabilities are delivered. The context defines circumstance affecting capability delivery. While the context can assume possibly an infinite set of values, it is assumed that the capability is designed for delivery for a limited range of specific context values, i.e., an organization does not

claim of being able to deliver the capability in all circumstance but only those prescribed in the capability delivery context. Nevertheless, the capabilities are designed to cover as many context situations are possible and reasonable. The capability delivery is supported by a process, which is elaborated more specifically using process variants. The process variants can be constructed for dealing with specific capability delivery context situations. Designing capabilities for various context situations might be a laborious task, and patterns are used as one of the means for reducing design efforts and complexity. The patterns provide reusable solutions for capability delivery. They are also characterized by their context, which defines situation when this pattern is applicable.

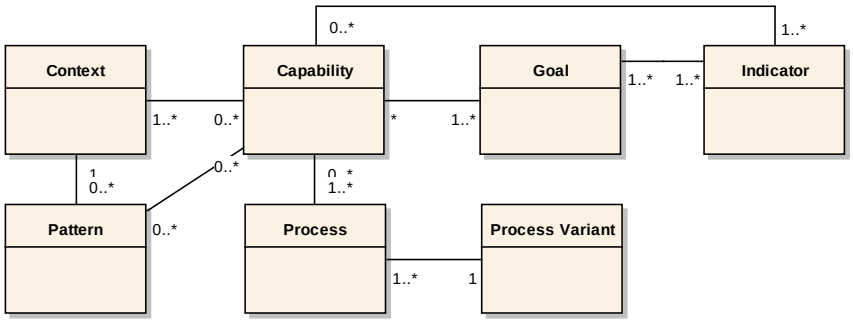


Fig. 1. A simplified capability modeling meta-model

2.2 Structure of the reference models

The SCOR model is widely analyzed [12, 13]. It identifies the key supply chain management processes and elaborates these at several levels of abstraction. This paper focuses on the process element level. At this level, the key supply chain management processes are defined as a sequence of activities. The description of every process is provided. This description includes process category definition, performance attributes and their evaluation metrics, best practices and their features as well as process inputs and outputs. Fig. 2 provides a simplified definition of concepts used in describing the third level processes.

The performance attributes are common for all processes and generally characterize the key dimensions used for supply chain evaluation. These include reliability, responsiveness, flexibility, cost and assets. Every attribute has process specific metrics. The best practices describe suggestions for improving the process and their features specify technologies contributing to successful adoption of the best practices. The input and output elements link together different processes.

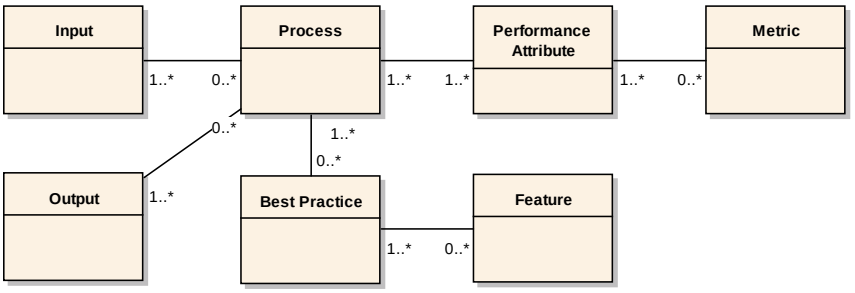


Fig. 2. Concepts used in defining process at the element level

3 Model Mapping

The SCOR model defines the key supply chain management processes to deliver value to supply chain customers. These processes can be expressed in terms of the capability meta-model (Table 1). The capability encompasses a goal-oriented and contextualized top level process. The top level process itself corresponds to the Process concept in the capability meta-model. Elaboration of this process into process categories is mapped to the Process variant concept. That implies that the main underlying process can be executed in differently for every process variant. Performance attributes and metrics are mapped into goals and indicators, respectively.

The best practices defined in the SCOR model are expressed as patterns in the capability model (the guidance for describing capability delivery patterns can be found in [14]). From the perspective of this paper, the most important aspects are that every pattern has its applicability context and the best practice solution is expressed a process fragment. The applicability context defines a range of circumstances this pattern was found to be useful. It is compared with the context of capability to be designed to identify suitable patterns. The patterns are stored in a searchable repository. The repository can contain other patterns beside those derived from the SCOR model.

Table 1. Correspondence between the SCOR model and capability model

SCOR concepts	Capability concepts
Level 1 process	Process
Level 3 process	Process variant
Best practice	Pattern
Performance attribute	Goal
Metric	Indicator

The SCOR model does not describe supply chain management context and there is little work on supply chain contextualization. The related research on risk averse supply chain management [15] suggests that the common high level context factors are location, compliance, weather, traffic, socio-economic and competition data. These

context factors can be measured using various sources and the capability delivery depends on these factors. For instance, many countries maintain important/export restrictions and the process execution depends on origin of materials or destination of products.

4 Capability Design

The capability design involves three main aspects: 1) specification of the desired capability; 2) identification of contextual factors affecting the capability; and 3) elaboration of the capability delivery solution. The reference model primarily addresses the latter concern though it is also helpful in specification of the desired capability. The reference model based capability design includes the following main steps:

1. Naming of the desired capability and definition of the capability goals;
2. Identification of contextual factors affecting the capability collectively referred as to context;
3. Selection of the appropriate process supporting the capability;
4. Selection of the process variants;
5. Specification of capability indicators;
6. Search for appropriate patterns in the pattern repository;
7. Insertion of the selected patterns in the process variants.

The capability is named in high level terms. This name relates to the first level processes in the SCOR model. The capability goals are derived from the SCOR model's performance attributes. The supporting process is selected as one of the SCOR's major processes though it is possible that a single capability might combine several of the major processes (e.g., deliver and plan processes are supporting one capability). The process categories are selected from the SCOR model to become process variants. The organization selects only the categories it considers necessary for attaining the capability (e.g., Engineer-to order is not selected if only make-to-order and make-to-stock policies are considered). The context dependency is indicated for the selected processes. The capability indicators are selected from the metrics defined in the SCOR model for the selected process categories.

The patterns in the pattern repository are searched by comparing the capability context with the pattern applicability definition also expressed as context. A suitability rank is computed for every pattern. The ranking is based on similarity between the capability context definition and the pattern context definition what can be evaluated using measures described in [16]. The final selection of the patterns is performed by a human decision-maker and the solution procedure defined in the pattern is incorporated in the overall process design.

5 Example

The capability design is illustrated using the demand fulfillment capability. The capability is developed on the basis of the Deliver process (Fig. 3). The overall

capability goals are defined according to the performance attributes. In this case, reliability, responsiveness and costs are selected as relevant. The organization also chooses to support only Deliver Stocked Product and Deliver Make-to-Order policies and the corresponding process variants are added to the capability design from the SCOR model. The indicators measuring capability delivery goals (not shown in the figure) are derived from metrics associated with the selected process variants. Sample indicators are Order entry and maintenance costs and Percentage of call back of total inquiries.

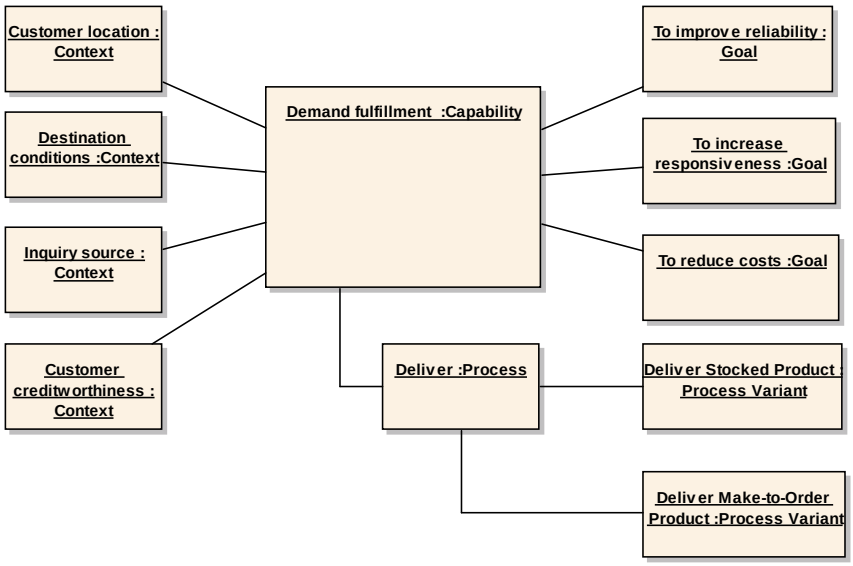


Fig. 3. The key elements of the demand fulfillment capability

The context elements identified for the demand fulfillment capability are customer location, destination conditions, inquiry source and customer creditworthiness. The Customer location context element captures region the customer is from (e.g., regions one to six typically used by many e-commerce companies). The particular capability is designed to support deliveries to regions one to five (i.e., the organization does not possess the capability of delivering to region six) and not all products are delivered to all regions. The destination conditions are classified as normal and hazardous what might be caused by extra-ordinary events taking place at the destination (this information can be aggregated from various web sources). The inquiry source defines the channel customer uses to make an inquiry (e.g., mobile, web, e-mail). The customer creditworthiness can be retrieved from internet-based credit agencies and the organization classifies customer as trusted, trusted with advance payment and non-trusted. The capability is purposely designed to support only trusted and advanced payment customers and support for these two types is provided in the processes while non-trusted customers are rejected or treated by some other capabilities.

It is assumed that the pattern repository contains three patterns described in Table 2. These patterns are derived from the best practices in the SCOR model with addition of contextual information and the process describing the solution as well as from other sources.

Table 2. Patterns defined in the repository

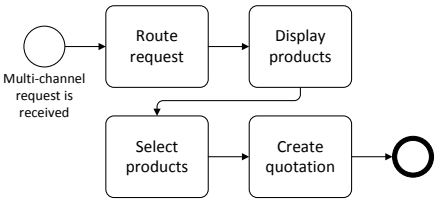
Name	Description	Process fragment
Single point of contact	Problem: Customer requests received through various channels get lost Context: Customer profile, Inquiry source Solution: All request are routed for processing in a single view	
Tailor offering	Problem: Customers should not be offered products, which cannot be delivered Context: Customer location, Destination, Product type Solution: Filter products according the context values	
Extra insurance	Problem: Some shipments are dangerous and require extra precautionary measures Context: Customer location={R1- R5}, Destination conditions={Normal, Hazardous} Solution: Insure high risk shipments	

Fig. 4 shows the Deliver stocked products process variant, and the context dependencies are indicated using data objects. In order to elaborate this process, the pattern repository is queried to find solutions for dealing with various context situations. In the case of D1.1 Process Inquiry & Quote, two patterns are found in the repository, and they are used to refine the process (inserted patterns are shown in the expanded sub-process in Fig. 4). D1.2 depends on customer creditworthiness, and there is no corresponding pattern available. Nevertheless, the activity needs to be refined to represent process variability to deal with different types of customers.

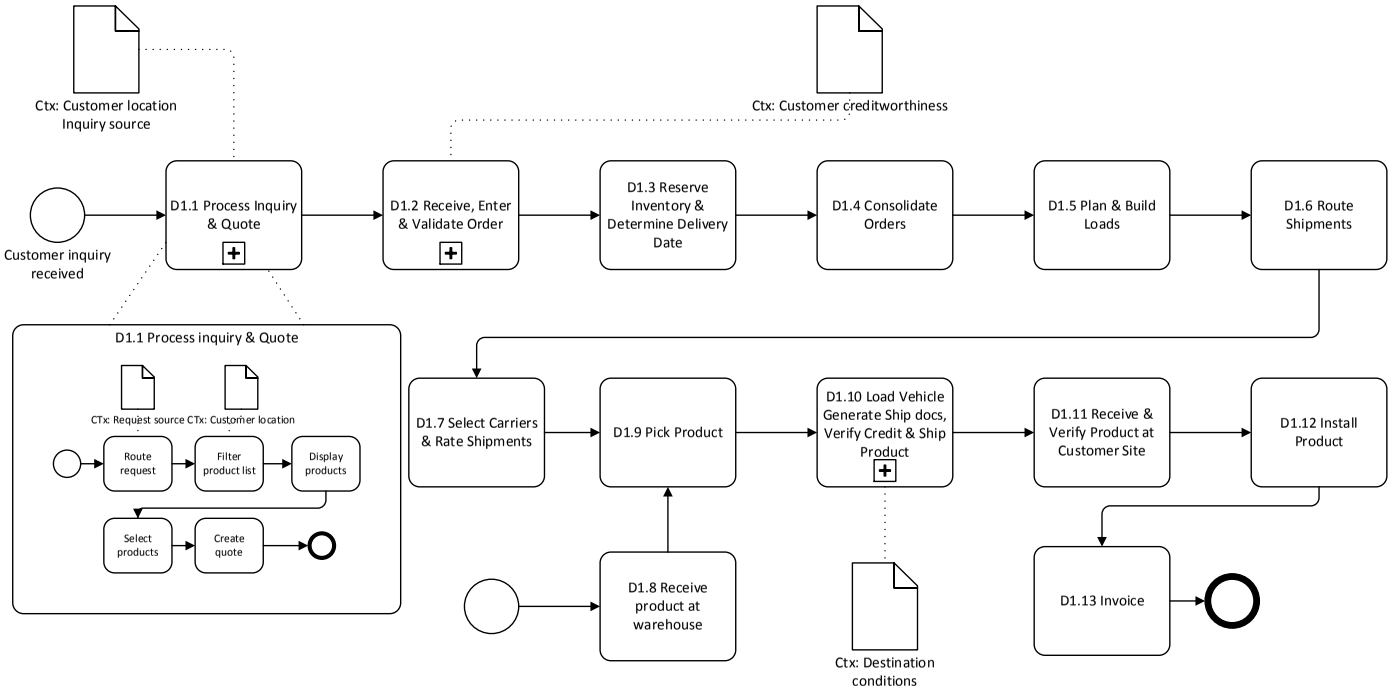


Fig. 4. Deliver stocked products process variant

The Extra insurance pattern is found to be suitable for D1.10 and is used to provide a solution for dealing with hazardous conditions at the delivery destination. In the example, there is a direct correspondence between the capability context and the pattern context what would not be a case in more realistic cases. That would require using appropriate similarity measurements.

6 Conclusion

The paper reports an initial proposal of the method for the SCOR model based capability design. It allows for quick development of new capabilities using established best practices. The SCOR model is used in two ways. Its process categories and metrics are used to specify process variants and indicators in the capability model. Its best practices are used to populate the repository of capability delivery patterns, which are used to refine processes supporting the capability delivery. The SCOR model also benefits from its combination with capability modeling by introducing the context dimension.

The feasibility and practicality of the proposed method depends upon availability of appropriate patterns and ability to contextualize supply chain management processes. Identification of relevant supply chain context factors and describing their impact on supply chain management processes in an important area for further investigations. The further elaboration of pattern selection and process refinement mechanisms is also required along with guidelines describing usage of the method.

References

1. TOGAF. TOGAF, Version 9.1, http://www.opengroup.org/togaf/ (2011)
2. OMG. Unified Profile for DoDAF and MODAF, Version 2.1, http://www.omg.org/spec/UPDM/2.1 (2013)
3. Stirna, J., Grabis, J., Henkel, M. & Zdravkovic, J.: Capability driven development - An approach to support evolving organizations. Proceedings of PoEM'2012, The Practice of Enterprise Modeling, ,LNBIP, pp. 117-131 (2012)
4. De Oliveira, L.B.R., Romero Felizardo, K., Feitosa, D. & Nakagawa, E.Y. 2010, Reference models and reference architectures based on service-oriented architecture: A systematic review. 4th European Conference on Software Architecture, ECSA 2010, LNCS 6285, pp. 360-367 (2010)
5. Supply Chain Council. Supply Chain Operations Reference Model, Version 10.0 (2011)
6. Huang, S.H., Sheoran, S.K. & Keskar, H.: Computer-assisted supply chain configuration based on supply chain operations reference (SCOR) model. Computers and Industrial Engineering 48, 2, 377-394 (2005)
7. Millet, P., Schmitt, P. & Botta-Genoulaz, V.: The SCOR model for the alignment of business processes and information systems. Enterprise Information Systems 3, 4, 393-407 (2009)
8. Medini, K., Bourey, J.P.: SCOR-based enterprise architecture methodology.International Journal of Computer Integrated Manufacturing 25, 594-607 (2012)
9. Zdravković, M., Panetto, H., Trajanović, M. & Aubry, A.:An approach for formalising the supply chain operations. Enterprise Information Systems 5, 4, 401-421 (2011)

10. Bjeladinović, S. & Marjanović, Z.: A Comparison and Integration of Ontologies Suitable for Interoperability Extension of SCOR Model. 6th Information and Communication Technologies Innovations 2014 conference, Advances in Intelligent Systems and Computing 311, pp. 75-84 (2015)
11. Bērziša, B., Bravos, G., Gonzalez, T.C., Czubayko, U., España, E., Grabis, J. et al.: Capability Driven Development: An Approach to Designing Digital Enterprises. Business & Information Systems Engineering 57, 1, 15-25 (2015)
12. Stephens, S.: Supply Chain Operations Reference Model Version 5.0: A New Tool to Improve Supply Chain Efficiency and Achieve Best Practice. Information Systems Frontiers 3, 4, 471-476 (2001)
13. Huan, S.H., Sheoran, S.K. & Wan, G.: A review and analysis of supply chain operations reference (SCOR) model. Supply Chain Management. 9, 1, 23-29 (2004)
14. Stirna, J. & Sandkuhl, K.: An outlook on patterns as an aid for business and IT alignment with capabilities. 26th International Conference on Advanced Information Systems Engineering, CAiSE 2014, LNBIP 179, pp. 148-158 (2014)
15. He, M., Ji, J., Wang, Q., Ren, C., Lougee, R: Big data fueled process management of supply risks: sensing, prediction, evaluation and mitigation. Proceedings of the 2014 Winter Simulation Conference, pp. 1005-1013 (2014)
16. Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal 10, 4, 334–350 (2001)