Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

Applying Situational Method Engineering to the Development of Service Identification Methods

Rene Börner
ProcessLab, Frankfurt School of Finance & Management, Sonnemannstr. 9-11, 60314 Frankfurt/Main, Germany,
r.boerner@frankfurt-school.de

Applying Situational Method Engineering to the Development of Service Identification Methods

Rene Börner
ProcessLab, Frankfurt School of Finance & Management,
Sonnemannstr. 9-11, 60314 Frankfurt/Main, Germany
r.boerner@frankfurt-school.de

ABSTRACT

Recently, situational method engineering (SME) has become more and more popular. Due to the fact that different situations require different methods, SME offers a flexible adaptation of methods. Furthermore, various approaches for the identification of services within service-oriented architectures have been proposed in the last few years. However, none of them provide a significant degree of situation-specific adaptability. Goal of this paper is to develop a meta model that can be used to construct service identification methods specific to certain situations. Therefore, ideas of SME are transferred into the realm of service identification. The meta model is further applied to a fictitious case.

Keywords

Service-oriented Architectures, Service Identification, Situational Method Engineering, Method Composition.

INTRODUCTION

Service-orientation is a highly recognized paradigm in enterprise architecture. There are lots of expected benefits related to service-oriented architectures (SOA) in a technical and in a business-oriented sense. Although the business-oriented benefits, like flexibility, reusability and standardization are of high importance, up to now, development of SOAs is mainly technically driven and most approaches consider technical aspects in the first place (Nadhan, 2004).

For the last couple of years many authors have been looking at the identification of services, which is one of the first and most important steps for an SOA implementation (for an overview see (Börner and Goeken, 2009b)). Interestingly, most existing methods to identify services are based on a one-fits-all approach. Only few consider a configuration of methods depending on different circumstances such as the goal of an SOA implementation. Even if the latter are considered, the scope of configurations is usually very limited.

The field of method engineering (ME) that has emerged since the early 1990s has been advanced by ideas of situational method engineering (SME) in the last decade. The central aspect behind it is that a fixed method is not suitable for all situations that occur in reality. Thus, methods have to be adaptable to different kinds of situations. The objective of this paper is to apply SME to the configuration of methods for service identification. Therefore, a meta model that transfers ideas of SME to the challenges of developing service identification methods is presented.

The paper is structured as follows: Section 2 discusses related work of both the field of SME and service identification. In section 3 a meta model for the development of service identification methods based on SME is suggested. One instantiation of the meta model at a concrete example in described in section 4. Finally, section 5 gives a summary, reflects on limitations and proposes future research on the topic.

RELATED WORK

Method engineering is a discipline in information systems research meant to "design, construct and adapt methods (…) for systems development."(Ralyté, Brinkkemper and Henderson-Sellers, 2007) Based on the fact that one method constructed at the time (t1) cannot fit to all conceivable conditions and circumstances when it is used at the time (t2), the concept of situational method engineering was created. Indeed, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification (Aydin, 2007). A method should thus be configurable. To provide for the adaptability of a method so called fragments are constructed and afterwards composed depending on the situation (Ralyté and Rolland, 2001).

ME literature offers many method elements that constitute method fragments as discussed in this section. The four most commonly used elements shown in Figure 1 are activities, techniques, results and roles (Gutzwiller, 1994; Heym, 1993;

Brinkkemper, 1996; Brinkkemper, Saeki and Harmsen, 1999; Karlsson, 2002). Subsequently, according to (Cossentino, Gaglio, Henderson-Sellers and Seidita, 2006) a method fragment consists of precisely these elements.

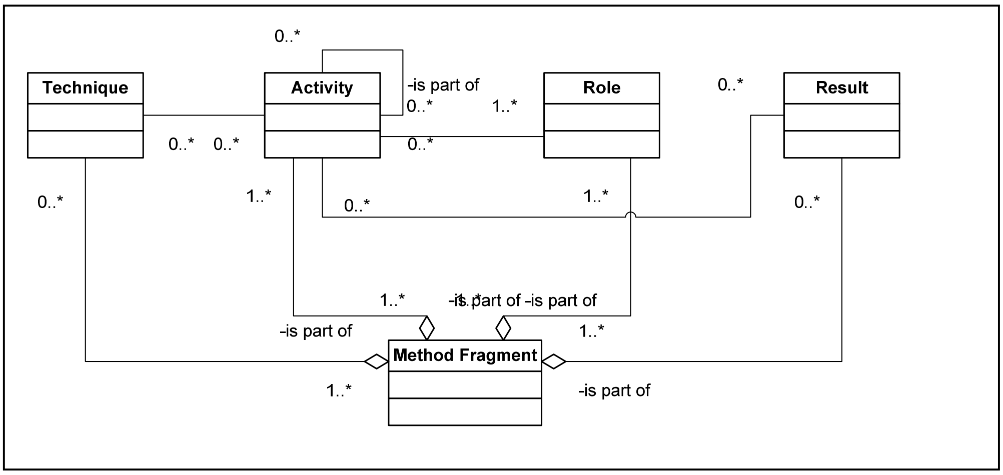

Figure 1. Concept of a method fragment

The term method fragment is used inconsistently in literature (Sunyaev, Hansen and Krcmar, 2008). (Agerfalk, Brinkkemper, Gonzalez-Perez, Henderson-Sellers, Karlsson, Kelly and Ralyté, 2007) define method fragments as “standard building blocks based on a coherent part of a method. A situational method can be constructed by combining a number of method fragments.” Some authors use the term synonymously with method chunk (Ralyté and Rolland, 2001). For the purpose of this paper, the notion of method fragments will be defined in the next section and used throughout the paper.

A number of approaches for service identification can be found in related literature. Herein, the analysis is limited to criteria such as configurability and facilitated method elements (i.e. activities, roles, techniques and results) that will be subject to further considerations in the following sections (see Table 1). A detailed analysis of existing methods can be found in (Börner and Goeken, 2009a).

	Klose et al. (2007)	**Böhmann & Krcmar (2005)**	**Winkler (2007)**	**Arsanjani et al. (2008)**	**Kohlmann & Alt (2007)**	**Kohlborn et al. (2009)**
Activities	++	++	++	++	+	++
Roles	--	--	--	o	--	o
Techniques	+	o	+	++	-	+
Results	+	++	+	++	+	++
Configurability	-	--	--	o	-	--
-- not existent - only implicitly o mentioned + defined/used ++ special focus						

Table 1. Method engineering elements and configurability in existing approaches

Usually, activities and results are described fairly well in all compared methods for service identification. The stakeholder-based approach by (Klose, Knackstedt and Beverungen, 2007) does not deal with configuration issues explicitly. Activities and techniques are described in detail whereas roles are not mentioned. Although a strict sequence of activities leaves no room for flexible adaptation, they acknowledge that certain preconditions can influence the process of service identification. The method presented by (Böhmann and Krcmar, 2005) neither discusses different roles nor describes techniques in depth. Although one of their first steps is the documentation of goals and scope (which should lead to different application contexts), this has no apparent influence on the method design. In Winkler’s approach (2007) any hints to roles are missing, too. She presents a strictly sequential proceeding. (Arsanjani, Ghosh, Allam, Abdollah, Ganapathy and Holley, 2008) note the importance of roles but omit a detailed description from their approach. They describe a fractal model and allow for an iterative sequence of activities. Their method can be adapted to the respective circumstances of a project but the authors fail to give guidelines how to configure it. Again, roles are not discussed in (Kohlmann and Alt, 2007) and even techniques are rarely mentioned. Their activities are in sequential order allowing iteration at only one point. Finally, (Kohlborn, Korthaus, Chan and Rosemann, 2009) consider roles in so far that their clear distinction between business services and software services necessitates that employees from functional departments as well as from the IT department are involved in the service identification process. They provide techniques but their method is not configurable for different application contexts. Table 1 shows the result of the conducted analysis.

A META MODEL FOR SITUATIONAL METHOD ENGINEERING OF SERVICE IDENTIFICATION METHODS

Configuration vs. Composition of Method Fragments

For the purpose of this paper any reasonable combination of method elements representing a coherent part of a method shall be referred to as method fragment. Figure 1 illustrates the concept of a method fragment as used herein. A method fragment does not have to include either all elements or one special element. It is meant to support the composition of a method to identify services. It may include even multiple instances of elements (e.g. more than one activity or technique).

According to (Bucher, Klesse, Kurpjuweit and Winter, 2007), there are two ways to build a configurable method, namely situational method configuration and situational method composition.

Situational method configuration follows the so called adaptive principle. This means that at design time (t1) of the method changes depending on the situation are explicitly allowed. There are precise instructions how an existing method has to be configured in certain contexts at time (t2). If situational method configuration is used, situational changes to a base method have to be foreseen and planned when a situational method is developed at time (t1). *Situational method composition* provides for a spontaneous combination of method fragments (orchestration) that does not have to be foreseen at (t1). There is no pre-defined base method that is adapted. Instead, method fragments are combined and aggregated as required at (t2).

Since all method fragments will be coherent parts of a method for service identification, situational method composition will be used in the following. After a situation is identified, a method for service identification will be composed at (t2). Method fragments are the building blocks of this method. Composition will be based on well-defined composition rules. These rules will be attached to the fragments (see Fig. 2).

What makes a situation?

Many authors agree that characteristics of a project have to be defined in order to describe a situation (Brinkkemper, 1996; Karlsson and Agerfalk, 2004). Still, according to (Bucher et al., 2007) they do not define sufficiently what constitutes a situation. In order to identify different situations, (Bucher et al., 2007) define a situation as combination of context and project types. The context is represented by environmental contingency factors. Contingency factors remain constant while the method is applied to solve a certain problem. A project type is inherent to a project and is transformed from its initial state to a target state by the method's activities.

In practice there are numerous context variables and each of them can have multiple parameter values. Examples can be found in the next section. In order to identify a situation, all reasonable context parameter value combinations (CPVC) have to be considered. All of these combinations make up one dimension necessary to identify a situation. For the purpose of this paper the above model is altered to better support the needs inherent to service identification in SOA. Thus, the second dimension is not a project type as proposed by (Bucher et al., 2007) but an SOA implementation goal (SOAIG) (see Fig. 2).

A matrix used for identifying relevant situations will therefore consist of the dimensions CPVC and SOAIG. Every intersection of this Situation Identification Matrix combining a CPVC with an SOAIG results in one specific situation (see Figure 3). Thus, the number of situations is determined by the number of CPVCs times the number of SOAIGs.

$$(\text{\# of situations}) = (\text{\# of CPVCs}) \times (\text{\# of SOAIGs})$$

Keeping in mind that already the number of context parameter value combinations can be quite large the number of situations can be significant. An early limitation of context variables, parameter values, CPVCs and implementation goals will help to save time and reduce complexity. The earlier dispensable variables and parameters are eliminated the more effective an identification of situations will proceed. Figure 4 shows five steps (a) to (e) that are necessary to define situations within the Situation Identification Matrix:

(a) Context variables: Identification of influential environmental contingency factors is the first important step. There are a number of potential factors that could probably influence a situation and it is important to find the decisive ones. These factors can be derived from literature or expert interviews.

(b) Parameter values: All context variables identified in (a) have certain parameter values. The latter have to be defined in this second step. Although the parameter values small, middle and large are commonly used to describe a company's size (Brooksbank, 1991), a differentiation between small and large might be sufficient for certain purposes. A finer-grained differentiation increases complexity and should only be used if the choice of the method is influenced by that.

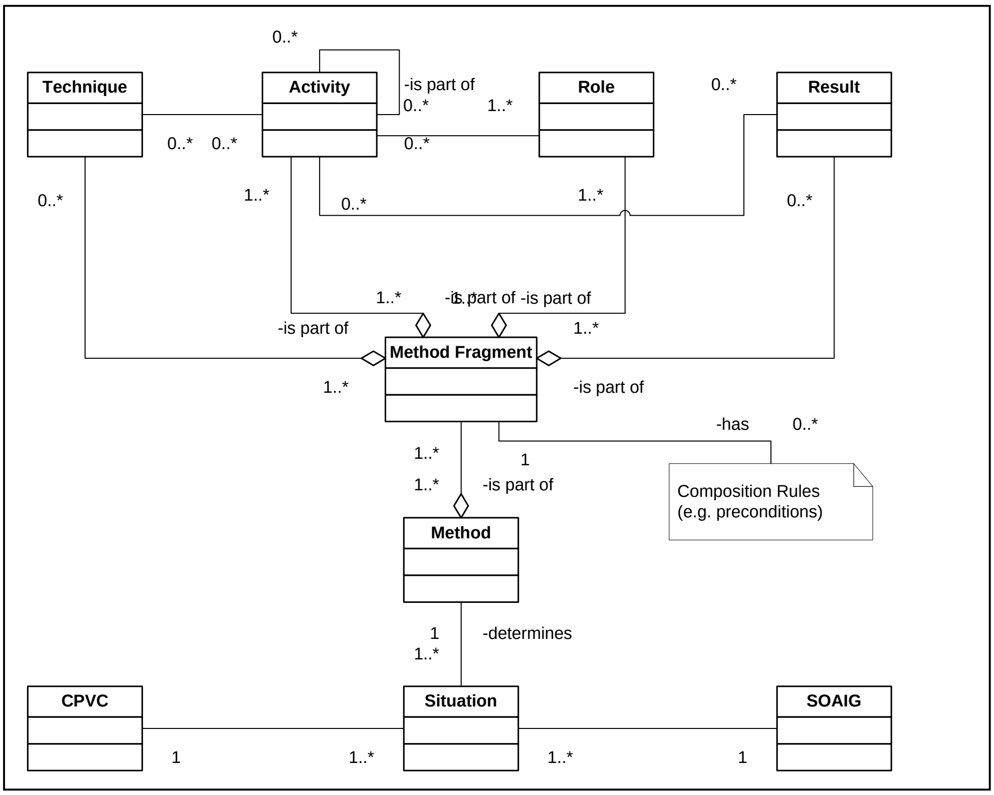

Figure 2. Concept of a situational method

(c) Context parameter value combinations: All parameter values are combined to constitute one dimension of the Situation Identification Matrix. At this point some combinations can be discarded. E.g. a small company combined with a project budget bigger than five million euros is not reasonable. Therefore, this parameter value combination can be removed from the matrix. From case to case combinations that can be excluded have to be determined. Certainly, steps (a) and (b) have a much stronger influence on the total number of combinations than the exclusion of single combinations described here.

(d) Similar to the choice of context variables described in (a), relevant SOA implementation goals (such as legacy system integration) have to be chosen. These goals form the second dimension of the matrix.

(e) Finally, the Situation Identification Matrix is used to illustrate all possible situations. Generally, all combinations of CPVCs and SOAIGs should lead to a situation. Still, there are some combinations that can be discarded. Combining a small start-up company with the goal to integrate legacy systems is not a reasonable presumption for instance.

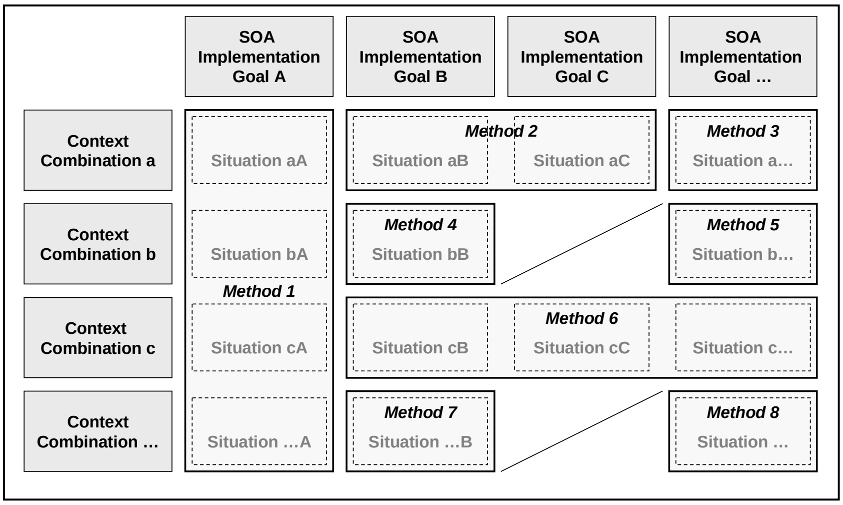

Figure 3. Situation Identification Matrix (following (Bucher et al., 2007))

Assigning Methods to Situations

Subsequently, every situation is mapped to a method that is tailored to the needs of the respective situation. The mapping of situations to methods is an n:1 relation, i.e. that every situation is assigned to exactly one method. However, methods may be used for several situations in case the preconditions of the latter are similar. As depicted in Figure 3 there are basically three types of methods. First, methods like number 3 or 4 apply to exactly one situation, i.e. they support one given set of context variables (in the form of a CPVC) and an implementation goal. Second, a so called goal-specific method like method 1 covers all situations with SOA implementation goal A irrespective of CPVCs. In this case the implementation goal is so dominant that any given CPVC does not influence the choice of a method. Third, methods like 2 or 6 are context combination-specific but span multiple implementation goals. In these cases only environmental factors determine the suitability of a method. The goal of an SOA implementation does not have a considerable effect on this selection.

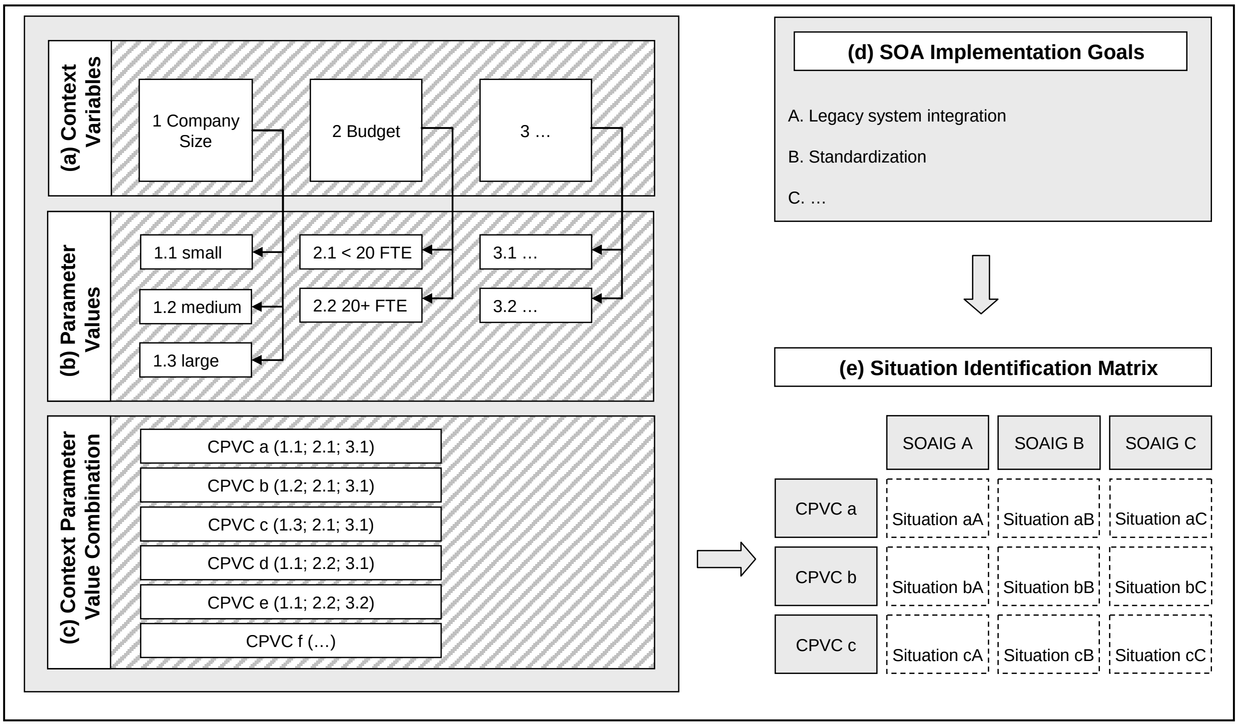

Figure 4. Five steps to define situations

INSTANTIATION OF A SERVICE IDENTIFICATION METHOD

The following section shows how a concrete method for service identification can be derived from the meta model described in the previous section. Firstly, some instances of method elements are shown and exemplarily combined to method fragments. Secondly, situations for service identification based on relevant contingency factors and SOA implementation goals are identified. Thirdly, a concrete example of a company is used to show how a method is composed out of method fragments. Hence, this method is tailored for the previously identified situation.

Building Method Fragments

In order to identify services, many activities, techniques, roles and results exist. Examples are shown in Table 2. Method fragments consist of these elements (see above). Although the latter are independent from one another in first place, they can be closely linked. For instance, the activity “create activity diagram” usually comes with the result “activity diagram”.

Method fragments should have meaningful names to improve composition. The broader process of service identification encompasses many steps that are represented by method fragments and finally part of an overall method. Examples for such fragments in this context are “Business Process Model”, “Breakdown of Business Processes”, “Strategic Alignment”, and “Positioning in the Value Chain”. It is important to design fragments as coherent and autonomous as possible to provide for an easier composition. However, there will be dependencies in using certain fragments that have to be reflected in composition rules.

Method Element	Instances
Activity	Identify sub processes Derive elementary activities Create activity diagrams Identify service candidates Normalize activity diagrams Analyze request frequency of services Explore technical feasibility Assess strategic potential
Technique	Business process modeling Asset analysis Goal service modeling Strategy questionnaire Governance questionnaire Technical feasibility checklist
Role	Business process owner (business department) Service owner (IT department) Service Design Unit (SDU) Service Excellence Center
Result	Activity diagram Business process documentation/landscape Service map

Table 2. Examples for method elements in service identification

The fragment “Breakdown of Business Processes” e.g. encompasses the activities “Identify Sub Processes”, “Derive Elementary Activities” and “Create Activity Diagrams”. The business process owner (role) is responsible for the activities and finally delivers activity diagrams (results). We assume that there is no distinct technique involved. However, existing business process models are a precondition for using this fragment (composition rule). Method fragments such as “Strategic Alignment” can require new organizational roles such as a service design unit (SDU) as proposed by (Börner, Looso and

Goeken, 2009). Based on a questionnaire this SDU provides for an alignment of (IT) strategy and services that support business (Alter, Börner and Goeken, 2009).

Identifying Contingency Factors and Implementation Goals

While a huge number of context variables could be used to describe situations in which a method is ought to be applied, only relevant factors for the identification of services shall be described in the following. On the one hand the list should be comprehensive in order not to miss a crucial variable. On the other hand – due to the complexity problem described previously – the number should be as small as possible. Table 3 shows the herein discussed context variables and SOA implementation goals.

According to (Sedera, 2008) the *company size* is important in this context. The *geographic scope* of a company's operation has to be considered in the process of service identification. It influences the cooperation of employees as far as activities are concerned and can lead to different roles and artifacts in the process of service identification. The *skills and qualification of employees* can vary significantly among projects and has to be taken into account (Becker, Knackstedt, Pfeiffer and Janiesch, 2007). There might be a necessity for external support if required know-how is not sufficiently available. The existence of a designated *IT department* is often bound to a company's size but should be considered explicitly as the existence of certain roles will depend thereon (Anderson, Howell-Barber, Hill, Javed, Lawler and Li, 2005). The *available budget* determines the number of full time equivalents (FTE) available in the project and is another important factor (Becker et al., 2007). The influence of other contingency factors, e.g. the legal form of the company, has been considered negligible in the context of service identification.

Context variables (parameter values)	SOA implementation goals
Company size (small, medium, large)	Integration of legacy systems
Geographic scope of operations (national, international)	Identification of outsourcing candidates
Employee qualification / skills (BPM skills, SOA skills, both, none)	Agility and flexibility of business processes
IT department (yes, no)	Standardization
Available budget (<20 FTE, 20+ FTE)	Provision of services for third parties

Table 3. Context variables, parameter values and SOA implementation goals

The goal of an SOA implementation and hence the identification of services is crucial for configuring a method. *Integration of legacy systems* is frequently a goal of SOA implementations (Erl, 2004; Heutschi, 2007; Arsanjani et al., 2008) especially in medium and large enterprises (Offermann, 2009). The identification of *outsourcing candidates* is another goal for SOA implementations (Beverungen, Knackstedt and Müller, 2008). In this case, costs, performance and strategic relevance of services must be analyzed. The *agility and flexibility of business processes* is a competitive advantage and strongly tied to the concept of SOA (Papazoglou, 2003; Heutschi, 2007). An alignment of business and IT is a necessary precondition to achieve this flexibility (Becker, Buxmann and Widjaja, 2009). In contrast to an enhanced flexibility on process services level, the *standardization* of basic services is meant to reduce redundancies in development and maintenance of IT and thus reduce costs significantly (Bieberstein, Bose, Walker and Lynch, 2005; Legner and Heutschi, 2007). A completely different perspective is taken by companies that aim at the *provision of services* for third parties. The former specialize on a small part of a value chain concentrating on their core competencies. They are able to generate economies of scale by providing services for many other companies typically – but not necessarily – belonging to the same industry sector. These cross company value networks become increasingly important e.g. in the banking industry (Kohlmann, 2007).

Composing a Method for a Concrete Situation

Exemplarily, one concrete situation, i.e. a combination of one SOA implementation goal and one context parameter value combination, shall be elaborated in the following. We assume that a small company (< 100 employees) with operations in only one country wants to provide services for third parties. This young company plans to use its core competencies in the production of industrial printing machines to establish itself as a layer player in a value network (Heuskel, 1999). Due to its size, there is no designated IT department. Instead, every business division takes care of its own IT infrastructure. Thus, no employee possesses any SOA know-how but some of the company's employees have considerable BPM skills. The budget is rather small so there is little scope to include external help.

Since the situation at hand is comprehensively described, a suitable method can now be selected. This method includes e.g. the fragment "Business Process Modeling" and uses the role "business process owner" to achieve a complete overview over the company's business processes. Fragments including roles such as "service owner" or "SDU" are not applicable due to the non-existence of an IT department. Thus, the method fragment "Strategic Alignment" as described above cannot be used in its current form. Since strategy plays an important role in this case, a similar fragment like "Service Provider Strategy" using other roles but pursuing the same goals has to be identified to complement the method. Furthermore, "Positioning in the Value Chain" is crucial to define the company's position within its value network. This fragment certainly contains results like service maps that support inter-organizational service integration (Kohlmann and Alt, 2009). The specifications given above determine a situation through their CPVC and SOAIG. Looking at the Situation Identification Matrix (Fig. 4) the most suitable method assigned to this situation can be chosen.

CONCLUSION AND FURTHER RESEARCH

So far, methods for service identification offer only limited support for a situation-specific adaptation. Situational method engineering taken from the realm of information systems development offers many approaches to provide precisely this kind of adaptability. This paper combines existing service identification methods with concepts from SME. In a first step, a meta model for method fragments and the identification of situations was presented. Afterwards, the applicability of this meta model was shown. All steps to develop a service identification method are demonstrated at a running example of a fictitious company. For this purpose, concrete value parameters for context variables and the SOA goal "Provision of services for third parties" are used.

However, there are some limitations to the findings presented in this paper. Context variables, their parameter values and the SOA implementation goals are taken from literature or based on experience. On the one hand the list might not be complete, i.e. important factors might not be included. Here, expert interviews could lead to further evidence about crucial contingency factors and relevant implementation goals. On the other hand, proof of the actual impact is weak for at least some variables and has to be improved through more empirical evidence. Currently, two case studies are evaluated in order to empirically underpin the relevance of contingency factors. A third case study is planned.

Finding a way to reduce the number of possible situations will be subject to further research. Using only five context variables, their respective parameter values and five implementation goals as described previously leads to 96 context parameter value combinations and thus to 480 situations in the Situation Identification Matrix. These combinations have to be limited to reasonable ones to produce fewer situations. Alternatively, methods have to encompass more situations in order to keep the number of methods manageable. Finding the right method for certain variables is not the main problem. Given a repository that contains all methods and a user interface that enables a search with given variables, retrieving a suitable method should be easy. However, the maintenance of such a large number of methods would pose a major problem.

Several examples for method fragments and their respective elements are mentioned in this paper. Method fragments are the building blocks for the composition of situational methods. Before this composition takes place, a repository containing method fragments has to be installed. Thus, reasonable combinations of method elements have to be combined to develop coherent fragments. Finding rules and giving advice on building such fragments is subject to further research. Particularly, an assignment of composition rules to each fragment needs to be elaborated in more detail. Designing method fragments for each case (i.e. every single company) is not feasible due to budget restrictions. Much more likely is the development of best practices for method fragments within certain industries that serve as a blueprint. Moreover, the applicability of method fragments may not be binary but could be marginally, fairly or highly relevant in a situation at hand. This again increases complexity.

Evaluation of the meta model so far rests on the case of one fictitious company. This can only indicate that the ideas of situational method engineering taken from information systems development are transferable to the area of service identification in service-oriented architectures. In order to properly evaluate the applicability and usefulness of the meta model, more evaluation methods should be applied.

REFERENCES

1. Agerfalk, P. J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S. and Ralyté, J. (2007), Modularization Constructs in Method Engineering: Towards Common Ground?, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP*, Springer, Boston.
2. Alter, S., Börner, R. and Goeken, M. (2009) Operationalisierung der IT-Governance-Kernbereiche für die Identifizierung und Gestaltung von Services, *Informatik 2009*, Lübeck, Germany, 3660-3674.

3. Anderson, D., Howell-Barber, H., Hill, J., Javed, N., Lawler, J. and Li, Z. (2005) A Study of Web Services Projects in the Financial Services Industry, *Information Systems Management*, 22, 1, 66-76.
4. Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S. and Holley, K. (2008) SOMA: A method for developing service-oriented solutions, *IBM Systems Journal*, 47, 3, 377-396.
5. Aydin, M. N. (2007) Examining Key Notions for Method Adaptation, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP Working Conference*, Springer, Geneva, Switzerland, 49-63.
6. Becker, A., Buxmann, P. and Widjaja, T. (2009) Value Potential and Challenges of Service-Oriented Architectures, *17th ECIS*, Verona, Italy.
7. Becker, J., Knackstedt, R., Pfeiffer, D. and Janiesch, C. (2007) Configurative Method Engineering, *13th AMCIS*, Keystone, Colorado, Paper 56.
8. Beverungen, D., Knackstedt, R. and Müller, O. (2008) Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung, *Wirtschaftsinformatik*, 50, 3, 220-234.
9. Bieberstein, N., Bose, S., Walker, L. and Lynch, A. (2005) Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals, *IBM Systems Journal*, 44, 4, 691-708.
10. Böhmann, T. and Krcmar, H. (2005), Modularisierung: Grundlagen und Anwendung bei IT-Dienstleistungen, in Thomas Herrmann, Uwe Kleinbeck and Helmut Krcmar (Eds.) *Konzepte für das Service Engineering. Modularisierung, Prozessgestaltung und Produktivitätsmanagement*, Physica, Heidelberg.
11. Börner, R. and Goeken, M. (2009a) Identification of Business Services, *15th AMCIS*, San Francisco, California, Paper 162.
12. Börner, R. and Goeken, M. (2009b) Methods for Service Identification, in Daniel Moldt, Juan Augusto and U. Ultes-Nitsche (Eds.) *7th International Workshop on MSVVEIS*, Milan, Italy, 76-84.
13. Börner, R., Looso, S. and Goeken, M. (2009) Towards an Operationalisation of Governance and Strategy for Service Identification and Design, *13th IEEE International EDOC Conference*, Auckland, New Zealand, 180-188.
14. Brinkkemper, S. (1996) Method engineering: engineering of information systems development methods and tools, *Information & Software Technology*, 38, 4, 275-280.
15. Brinkkemper, S., Saeki, M. and Harmsen, F. (1999) Meta-Modelling Based Assembly Techniques for Situational Method Engineering, *Information Systems*, 24, 3, 209-228.
16. Brooksbank, R. (1991) Defining the small business: a new classification of company size, *Entrepreneurship and Regional Development*, 3, 1, 17-32.
17. Bucher, T., Klesse, M., Kurpjuweit, S. and Winter, R. (2007), Situational Method Engineering - On the Differentiation of "Context" and "Project Type", in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP Working Conference*, Springer, Boston.
18. Cossentino, M., Gaglio, S., Henderson-Sellers, B. and Seidita, V. (2006) A metamodelling-based approach for method fragment comparison, *11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design*, Luxembourg.
19. Erl, T. (2004) Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services, Prentice Hall, Upper Saddle River, NJ.
20. Gutzwiller, T. A. (1994) Das CC-RIM-Referenzmodell für den Entwurf von betrieblichen, transaktionsorientierten Informationssystemen, Physica, Heidelberg.
21. Heuskel, D. (1999) Wettbewerb jenseits von Industriegrenzen: Aufbruch zu neuen Wachstumsstrategien, Campus, Frankfurt/Main.
22. Heutschi, R. (2007) Serviceorientierte Architektur, Springer, Berlin Heidelberg.
23. Heym, M. (1993) Methoden-Engineering - Spezifikation und Integration von Entwicklungsmethoden für Informationssysteme, Institut für Wirtschaftsinformatik, St. Gallen.
24. Karlsson, F. (2002) Meta-Method for Method Configuration - A Rational Unified Process Case, Linköping.
25. Karlsson, F. and Agerfalk, P. J. (2004) Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets, *Information & Software Technology*, 46, 9, 619-633.
26. Klose, K., Knackstedt, R. and Beverungen, D. (2007) Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning, *15th ECIS*, St. Gallen, Switzerland, 1802-1814.
27. Kohlborn, T., Korthaus, A., Chan, T. and Rosemann, M. (2009) Identification and Analysis of Business and Software Services - A Consolidated Approach, *IEEE Transactions on Services Computing*, 2, 1, 50-64.
28. Kohlmann, F. (2007) Service Modelling: A Hybrid Approach In Decomposed Financial Value Chains, *2nd International Workshop on EMISA*, St. Goar, Germany, 205-218.

29. Kohlmann, F. and Alt, R. (2007) Business-Driven Service Modeling, in Witold Abramowicz and Leszek Maciaszek (Eds.) *Business Process & Service Computing: First International Working Conference on BPSC,* Gesellschaft für Informatik e.V., Leipzig, Germany, 180-193.
30. Kohlmann, F. and Alt, R. (2009) Aligning Service Maps, *42nd HICSS*, Hawaii, 1-10.
31. Legner, C. and Heutschi, R. (2007) SOA Adoption in Practice, in Hubert Österle, Joachim Schelp and Robert Winter (Eds.) *15th ECIS*, St. Gallen, Switzerland, 1643-1654.
32. Nadhan, E. G. (2004) Seven Steps to a Service-Oriented Evolution, *Business Integration Journal*, 1, 41-44.
33. Offermann, P. (2009) Eine Methode zur Konzeption betrieblicher Software mit einer Serviceorientierten Architektur, Gito, Berlin.
34. Papazoglou, M. P. (2003) Service-Oriented Computing: Concepts, Characteristics and Directions, *4th International Conference on WISE,* IEEE Computer Society, Rome, Italy, 3-12.
35. Ralyté, J., Brinkkemper, S. and Henderson-Sellers, B. (2007), Situational Method Engineering: Fundamentals and Experiences, in Jolita Ralyté, Sjaak Brinkkemper and Brian Henderson-Sellers (Eds.) *IFIP,* Springer, Boston.
36. Ralyté, J. and Rolland, C. (2001) An Approach for Method Engineering, in H.S. Kunii, S. Jajodia and A. Solvberg (Eds.) *20th International Conference on Conceptual Modelling*, Springer, Yokohama, Japan, 471-484.
37. Sedera, D. (2008) Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations, *2nd Workshop on 3rd Generation Enterprise Resource Planning Systems*, Copenhagen, Denmark.
38. Sunyaev, A., Hansen, M. and Krcmar, H. (2008) Method Engineering: A Formal Approach, *17th International Conference on Information Systems Development*, Paphos, Cyprus.
39. Winkler, V. (2007) Identifikation und Gestaltung von Services, *Wirtschaftsinformatik,* 49, 4, 257-266.