

Towards Construction of Situational Methods for Service Identification

René Börner

ProcessLab, Frankfurt School of Finance & Management,
Sonnemannstraße 9-11, 60314 Frankfurt am Main, Germany
r.boerner@fs.de

Abstract. The service-oriented paradigm plays an increasingly significant role in designing and governing IT architectures in organizations. The identification of services belongs to the most important parts of the service management life-cycle and is essential for the successful implementation of service-oriented architectures (SOA). However, existing methods for service identification mostly ignore situation-specific factors for such projects. Situational method engineering can be used to design a meta method to support the development of situation-specific methods for service identification. Based on a literature review and two case studies, this paper elaborates on context factors and SOA implementation goals being constituting elements of situations. These situations are tailored to the service identification domain. Applying this meta method in concrete project situations will help to engineer appropriate methods for service identification.

Keywords: Situational method engineering, service-oriented architectures, service identification, context factors, meta method, SOA implementation goals.

1 Introduction

Service orientation is currently a dominating paradigm for enterprise and IT architectures. Service-oriented architectures promise a greater flexibility of IT and a faster adoption to changing business needs. The identification of services is one of the most important steps for successful SOA implementations and many authors have developed methods for this purpose (for an overview see [1]). Interestingly, most of these methods are based on a one-fits-all approach and do not consider a configuration of methods depending on different circumstances.

The field of situational method engineering (SME) offers opportunities to overcome these shortcomings. A 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. This paper elaborates on context factors and SOA implementation goals to identify situations in the field of service identification. These are necessary to support a reasonable configuration of fragments that are developed as part of the meta method for the configuration of methods for service identification. Identifying situations is pivotal for this meta method and the focus of this paper.

Designing suitable fragments complements the meta method and will be discussed briefly herein.

The paper is structured as follows: Section 2 discusses related work in the fields of SME and service identification, outlines the research design and describes the scope of this paper. In section 3, context factors and SOA implementation goals – both being defining parts of a situation – are elaborated in detail. The design of method fragments for service identification is presented in section 4. Section 5 reflects on limitations, proposes avenues for future research and provides a conclusion.

2 Foundations of a New Meta Method

This section discusses related literature in the field of SME and service identification. The research process used to derive the context factors is outlined. Furthermore, scope and goals of the meta method are defined.

2.1 Situational Method Engineering and Configurability in Existing Approaches

It is commonly accepted that no universal method constructed at time (t_1) can fit every conceivable situation in which it is applied in time (t_2) [2]. Actually, it is quite improbable that a rigid method developed from theory is applicable in a concrete setting without modification [3] and therefore, the concept of SME emerged (see e.g. [4]). 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. To support this adaptability, smaller parts of a method – so called method fragments – are created and can be composed depending on the situation at hand [5]. Method fragments consist of the four elements activity, technique, role, and result [6].

Unfortunately, the term method fragment is inconsistently used in literature [7]. Ågerfalk et al. [8] define method fragments as “standardized building blocks based on a coherent part of a method” (p. 360). A situational method can be constructed by combining a number 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 [8].

According to Bucher et al. [9], there are two adaptation mechanisms to engineer a situational method, namely *situational method configuration* and *situational method composition*. Situational method configuration follows the so called adaptive principle. This means that a base method is created at design time (t_1) and configured in certain contexts at time (t_2). For situational method configuration, situational changes to a base method have to be foreseen and planned when a situational method is developed at time (t_1). In contrast, situational method composition provides for a spontaneous combination of method fragments (orchestration) that does not have to be foreseen at (t_1). There is no pre-defined base method that is adapted. Instead, method fragments are combined and aggregated as required at (t_2). Börner [10] suggests a third possibility, in which pre-composed methods are assigned to situations in (t_2). Although methods are defined in (t_1) already, there is no pre-configured base method in this approach.

Many authors agree that characteristics of a project have to be defined in order to describe a situation [11-13]. Still, according to [9] they do not explicitly say what constitutes a situation. For the purpose of this paper, Börner's [10] concept of a situation is used. It defines *context factors* and *SOA implementation goals* as two determining factors that constitute a situation.

The analysis and comparison of existing service identification literature is the basis for the development of a meta method for situational methods conducted in this paper. Additionally to the literature analyzed by [1], two currently published approaches [14, 15] were included in this paper. All approaches were examined considering their description of activities, roles, techniques and results as well as the configurability of the presented methods. The first four criteria were chosen because they are commonly used elements for methods and method fragments [6, 16]. Configurability is frequently regarded important in SME literature. However, in the field of SOA and especially in service identification a lack of configurability of methods can be stated.

Since *activities* are the focus of all compared approaches, they are described precisely in most cases. Many *techniques* and intermediate *results* are usually provided as well. The most striking feature is the almost complete absence of *roles*. Although many authors discuss the importance of business and IT alignment, all compared service identification approaches tend to underestimate the significance of properly assigning roles to respective activities and techniques. Roles will play an important role when method fragments are assigned to situations.

The *configurability* describes the possibility to choose adequate fragments that suit the situation at hand and arrange them in an adequate sequence. The latter means that activities used in an approach do not have to be executed in a linear order but can be used iteratively and hence allow for loops or iterations. Although hints on possible configurations can be found in some places, none of the authors explicitly incorporates the former into their approach. Since the goal of this paper is to guide the engineering of situational methods, configurability of service identification methods will be considered a crucial feature.

2.2 Research Design

The development of a meta method for service identification methods presented herein is based on a hybrid research approach combining several research methods to gain a richer understanding of the topic [17]. Construction of the meta method supported by SME belongs to the realm of *design science*. The derivation of relevant context factors builds on both *desk research*, i.e. literature reviews, and *case study research*. Since the identification of relevant context factors is an explorative goal, a case study approach was deliberately chosen to give the results an empirical grounding. Case studies are appropriate in this respect because they "provide descriptions of phenomena" [18]. Furthermore, case studies are particularly relevant for research in its "early, formative stages" [19] which applies to the field of SOA [20].

The case studies were conducted in two SOA implementation projects in Australian companies providing completely different environments. One of the companies is a small data provider; the other is one of Australia's biggest insurance companies. The significantly diverse settings of both cases opened up a continuum [21] of instantiations for identified context factors, i.e. their parameter values. At the insurer,

researchers conducted an action research study. They actively participated in the project and helped test and apply a service analysis & design methodology developed by them previously. In the second case study, the most important sources of evidence have been interviews that were conducted shortly after the project had been completed. The data provider's employees and researchers were the interview partners. The researchers had helped the data provider to implement an SOA in order to enable the retrieval and analysis of heterogeneous data from different sources (grid environment) spontaneously in an unforeseeable fashion (ad-hoc).

The interviews have been transcribed afterwards and analyzed along with all other documentation and reports. In an iterative approach, relevant context factors were determined based on this data and compared with related literature. The identification of such factors (concepts) was conducted by employing techniques from grounded theory, for example, open and axial coding [22], and interpretative techniques [23]. Even though these coding techniques were not used to their fullest extent, the general approach and respective tools supported the assignment of statements from the interviews and documents to concepts. The goal was to detect relevant particulars within the case data and to identify relevant concepts [24], i.e. context factors.

2.3 Goals and Scope of the Meta Method

Due to the lack of configurability in existing approaches, this paper argues that situational method engineering can support methods for service identification that suit certain project situations and are thus situation-specific. Particularly, context factors (including their respective parameter values) and SOA implementation goals that jointly determine a situation are the focus. They provide the basis for the intended meta method that is subject to ongoing research.

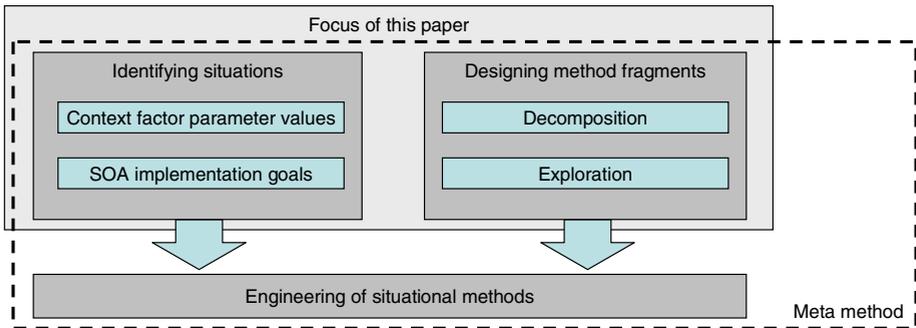


Fig. 1. Scope of the Meta Method and Focus of This Paper

This meta method encompasses the identification and description of context factors, their value parameters and SOA implementation goals. Together, combinations of these parts constitute a situation. Every instantiation of a method will rely on the situations that will be developed in section 3 since context factors, value parameters and SOA implementation goals belong to the service identification domain. Thus, the latter are design elements of this meta method. Apart from situations, the meta

method also includes descriptions of method fragments. Principles of fragment design will be outlined and shown exemplarily in section 4.

Figure 1 illustrates that the meta method encompasses the identification of situations and the design of method fragments. Both are necessary for the configuration of situational methods. Focus of this paper is the identification of situations in the domain of service identification based on relevant context factors and SOA implementation goals. Moreover, the design of method fragments as part of a meta method will be shown exemplarily herein. Therefore, two ways of method re-engineering, namely decomposition and exploration, are presented. A generally valid description of how to engineer situational methods in any conceivable situation is left to further research.

3 Identifying Situations

Following [10], the combination of context parameters and SOA implementation goals determines situations. Figure 2 illustrates the five necessary steps (a) to (e) to identify situations. These steps are introduced briefly herein (for a more detailed explanation see [10]):

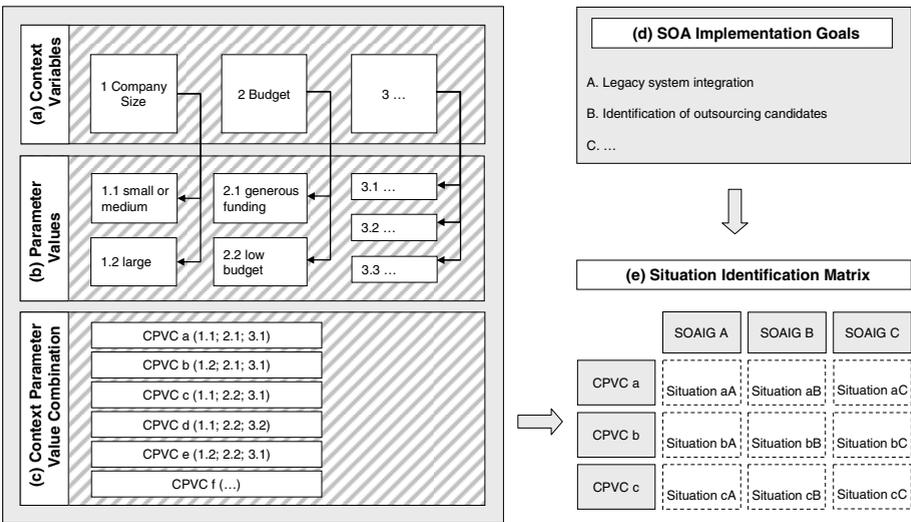


Fig. 2. Five Steps to Identify Situations (following [10], p. 5)

- (a) The context variables (contingency factors) that can influence the SOA implementation project at hand are defined.
- (b) The context variables identified in (a) have certain parameter values (instantiations) that are defined in this second step.
- (c) All context parameter values are combined with one another. These context parameter value combinations (CPVC) serve as one input for the situation identification matrix.

- (d) Possible SOA implementation goals are identified in this step. They are the second input for the situation identification matrix.
- (e) The situation identification matrix illustrates all possible combinations of CPVCs and SOAIGs retrieved in steps (c) and (d).

Context factors, respective parameter values and SOA implementation goals that are relevant to the service identification domain, and therefore an important part of the meta method presented herein, will be introduced in the following subsections. A combination of concrete parameter values (e.g. a small company with low budget, no industry-specific legal restrictions producing only one product) and an SOA implementation goal (provision of services for third parties) constitutes a specific situation.

3.1 Context Factors

Based on related literature and two recently conducted case studies described in section 2.2, the importance of the most influential context factors is discussed in the following. Although these context factors might not be unique to the field of SOA, their consideration in service identification methods will improve applicability of the latter.

It is a generally held belief that the *company size* is a considerable contextual factor in many kinds of software projects [25] such as service identification in SOA implementation projects. Whereas Sedera [26] proposes three classes of company size, this paper follows Welsh and White [27] who suggest only two. Hence, companies are differentiated into *small and medium-sized enterprises* on the one hand and *large companies* such as multi-national enterprises on the other hand. The former shall comprise organizations with up to 250 employees [28].

The *budget* of a project plays an important role when it comes to choosing necessary steps and general proceeding of a method for service identification. Generally, a generous budget that allows for an extensive time frame provides the opportunity for a thorough and systematic application of identification methods. One would expect the utilization of many techniques in order to ensure a high quality of implemented services. A detailed analysis of available strategic and technical documents would be typical in such circumstances. Literature confirms that the budget has implications on the number of available staff, the time pressure and the possibility to incorporate external help from consultants [29].

Some activities have to be carried out regardless of the budget. Still, there might be obligatory as well as optional techniques which support the activity. Consequently, an exhaustive use of techniques would only be chosen if financial resources are easily available. The parameter values *generous funding* and *low budget* will thus be used for this context variable.

Depending on a company's strategy, services can be provided for different *service consumers*, for example other divisions (internally), third parties (externally) or both. Strategically important services that lead to a competitive advantage in the market should only be available to end-consumers but not to competitors. A clear distinction between services offered to internal customers, other companies in a value chain or end-consumers is indispensable. If it is known in advance that a service will be offered internally only, a number of activities and results such as the creation of an inter-organizational service map are not applicable in this situation. Moreover, there

might be legal constraints that only apply if services are offered to third parties and thus demand an examination of these regulations. An analysis of consumer interaction is always essential whereas the “line of visibility” is much more important if services are exposed to external customers [30].

For the purpose of this paper, service consumers will be divided into *internal* and *external consumers*. Additionally, the case of a service being offered to *both* is considered.

Skills and experience with both service-oriented architectures and business process management significantly influence the proceeding of service identification. Limited SOA experience on the side of employees often leads to a technical SOA understanding. Project teams that are more familiar with the service-oriented paradigm are more likely to succeed in combining technical aspects with a Business Process Management perspective in mind. Hence, software services that support business processes or at least sub-processes can be the goal of their analysis. Their identification usually includes activities related to the analysis of process models and involves not only IT-related, but also business-related staff roles. If the service identification is limited to a rather technical point of view, the set of method fragments to be considered will therefore be a different one.

The configuration of situational methods and choice of fragments is affected, if for example certain roles cannot be occupied by available employees. Limited employee skills can necessitate external support by consultants. Although this enables the application of certain fragments, this option might be limited by the project budget and might therefore not be feasible. Parameter values for this context factor are *SOA skills available*, *BPM skills available*, *both skills available* and *none available*.

Furthermore, the **SOA maturity level** a company has achieved is seen as a further influential factor on the delivery strategy of SOA [31]. Thus, it plays an important role in the configuration of methods for service identification.

SOA maturity models are used to classify the status of SOA implementations within a company. This paper will use the Service Integration Maturity Model (SIMM) [32] to distinguish advanced organizations with *level 4 to 7* from less mature organizations (*level 1 to 3*). The former are likely to use more sophisticated and strategy-oriented fragments. The latter usually use more technically-oriented techniques and thus other fragments.

On the one hand, **compliance** issues can arise from legal obligations and regulatory restrictions. These differ among countries and especially companies that operate in more than one country have to consider legal demands arising from that. In many countries, all companies have to obey certain rules as far as the confidentiality of customer data is concerned. Additionally, some industries such as banking or pharmaceuticals have to adhere to special regulations. Finally, regulations can arise from the fact that a company is listed on a stock exchange, i.e. it also depends on its legal form. On the other hand, internal policies may require corresponding method fragments that address issues like service ownership. Three parameter values will be used for this context factor, namely *standard legal compliance*, *special regulations due to industry*, *legal form or international operations* and *internal policies*.

Another important context variable is the *existence* of a designated **IT department** and thus the degree of centralization of the IT infrastructure. In a small company that lacks an IT department, methods have to be adapted to accommodate for this

circumstance. Larger organizations usually have such an IT division or are structured along the lines of business. On the one hand a high degree of centralization or the existence of a central division supervising and governing IT implementation throughout a company usually leads to more transparency. Frequently, at least some information on applications and data is readily available. This can be used as input for service identification method fragments. On the other hand, some fragments demand certain roles such IT administrators or newly composed units consisting of business and IT employees (see also [33]). In a small company that lacks an IT department, these method fragments are frequently not applicable.

Table 1. Context Variables and Respective Parameter Values

Context Variable	Parameter Value
1 Company size	1.1 Small or medium-sized enterprise
	1.2 Large company
2 Service consumers	2.1 Internal consumer
	2.2 External consumer
	2.3 Internal and external consumers
3 Budget	3.1 Generous funding
	3.2 Low budget
4 Skills and experience	4.1 SOA skills available
	4.2 BPM skills available
	4.3 Both skills available
	4.4 None available
5 SOA maturity level	5.1 SIMM level 1-3
	5.2 SIMM level 4-7
6 Compliance	6.1 Standard legal compliance
	6.2 Special regulations
	6.3 Internal policies
7 IT department	7.1 Existent
	7.2 Not existent
8 Interaction	8.1 Customer interaction
	8.2 Employee interaction
	8.3 Customer and employee interaction
9 Organizational structure	9.1 One product company
	9.2 Multiple product company

Varying degrees and forms of *interaction* with both customers and employees necessitate the use of different method fragments. In some cases employees are not directly involved in service delivery because the services are very fine-grained and fully automated. The coarser-grained services are, the greater is the possibility that they are only semi-automated or manual and subsequently interact with employees. Customer interaction can be of high importance when the composition of services by the end user is a primary goal. In general, a customer interaction can be obligatory in some places or can happen “on demand” if required [34]. If customer interaction is a major issue for the identification of services in a situation at hand, respective method fragments (e.g. swim lane diagrams that show interfaces to customers) are crucial for a successful implementation. Thus, *customer interaction*, *employee interaction* and a *combination of both* are differentiated for the purpose of this paper.

In a company specialized on one product only, an analysis of a service's reusability is trivial in most cases. The same analysis is much more complex when looking at companies with a wide range of products. An organization can, e.g., be structured by products (business lines), regions, functions or customer groups [35, 36]. Even a multidimensional structure combining two or more dimensions of the above is not uncommon. Thus, the *organizational structure* can be an important factor when it comes to service identification. Herein, *one product companies* and *multi product companies* are differentiated.

Table 1 gives an overview of the context variables used and their respective parameter values. After identifying the context factors that are one part of a situation, the next section will elaborate possible goals for the implementation of service-oriented architectures that are the second constituting part of a situation.

3.2 SOA Implementation Goals

The second constituting element of a situation are SOA implementation goals. Depending on the purpose of an SOA implementation, the identification of services can necessitate the application of different method fragments. Many such goals can be found in related literature and the case studies also confirmed some of them. In the following, these goals and their influence on a situational method configuration will be discussed.

The *integration of legacy systems* is a frequently mentioned goal of SOA implementations [37, 38]. Especially in medium-sized and large enterprises, IT architectures have developed over years or even decades. In the absence of a central governing body, manifold isolated applications were developed and implemented which led to a plethora of problems. New functionalities and updates have to be made separately for each system which causes high maintenance costs. In some cases, it is difficult to find specialists who are able to administer for instance cobol code. Due to their restricted function-oriented view, employees do not know about IT systems of other divisions. This redundancy causes high costs because of unnecessary licensing fees.

Hence, integrating existing applications plays a major role in enterprise IT architectures and is one of the reasons for SOA implementation projects. In this case, techniques such as asset analysis and results that illustrate dependencies of the existing IT infrastructure are crucial parts of the service identification. The knowledge of IT experts about technical interfaces is indispensable.

The *identification of outsourcing candidates* is another goal for SOA implementations [39]. In this case, costs, performance and strategic relevance of services must be analyzed. On the one hand, based on a business process analyses the exact scope of the outsourcing activity has to be defined. A strategic make-or-buy decision determines which parts of the process are performed within the organization and which parts shall be outsourced to service providers. On the other hand, an outsourcing candidate needs clearly defined technical interfaces. Inputs and outputs of automated services provided by a third party have to be explicated in service level agreements. These outsourcing considerations demand fragments that produce for instance inter-organizational service maps and incorporate strategic aspects as well as detailed technical descriptions.

The *agility and flexibility of business processes* is a competitive advantage and strongly tied to the concept of SOA [40]. An alignment of business and IT is a necessary precondition to achieve this flexibility. Therefore, a company's strategy, i.e. a business process perspective, has to be considered. An enterprise-wide governance of the IT infrastructure is indispensable to provide for this agility. Hence, fragment results such as service ownership models [41] have to be used.

In contrast to an enhanced flexibility on process services level, the *standardization* of basic services is meant to avoid redundancies in development and maintenance of IT and thus to reduce costs significantly [42]. The goal is to improve efficiency by reusing a service in as many processes as possible. However, a customer should not be limited in his choice of varieties. A faster processing through increased efficiency should lead to a higher customer satisfaction. Therefore, services that directly interact with customers should not be standardized. This makes fragments for the analysis of the line of interaction and the line of visibility indispensable.

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. These companies are able to generate economies of scale by providing services for many other companies typically – not necessarily – belonging to the same industry sector. Hence, the focus here is again on inter-organizational and strategic instruments. Services must be easily exposable to third parties, i.e. interfaces have to be well-defined and performance has to be readily measurable. Method fragments should thus concentrate on interaction, interface analysis and the strategic value of providing a service to third parties.

4 Designing Method Fragments for Service Identification

In order to design situation-specific methods for service identification, method fragments that support this identification have to be provided. There are basically two possibilities to design these method fragments [43]. On the one hand, fragments can be re-engineered from existing methods. On the other hand, they can be designed from scratch in case no experience exists, i.e. no fragments or elements can be retrieved from existing approaches.

As shown in section 2.1, literature provides a number of methods for service identification. Although they include many effective method elements and fragments, the lack of configurability is a major shortcoming. Thus, the following design of method fragments will concentrate on re-engineering of existing methods rather than on ad-hoc construction. Ralyté [44] identifies two ways to design method fragments from existing methods, namely *decomposing models from existing methods* and *exploring different possibilities to apply a model* (p. 5). In the following, both will be introduced. Two examples of method fragments will show the applicability to the service identification domain.

4.1 Decomposition

Identifying fragments through decomposition is supposed to be easier than creating new ones through exploration and should thus be the first step. Most of the fragments

that can be found in existing service identification approaches concentrate strongly on the result of activities and are thus strongly product-driven. In these cases the process part – including roles and techniques – has to be conceptualized in order to obtain fragments. In cases where a fragment is identified by process model decomposition, the product part has to be elaborated since the processes are already available [44]. The following is one example of a fragment that was decomposed from an existing method [30] and enhanced as far as the process part is concerned.

Fragment 1: Overview of Existing Process Models	
Description:	If there are documented business processes, these should be used for the further analysis to save time and money if possible.
Input:	Meaningful documents of existing business processes from formerly conducted Business Process Management (BPM) projects
Preconditions:	If no BPM projects had been conducted before, it is necessary to identify business processes before using this fragment.
Taken from:	Klose, Knackstedt, Beverungen (2007)
Design:	In the first phase "preparation" of their approach, Klose et al. [31] include the task "prepare existing process models" into their procedure model (p.1804). Since the authors describe activity, techniques and results, the fragment is identified by product decomposition. Only the role to perform the activity has to be added in order to complete the fragment.
Activity:	Preparation
Role:	Employee of the business department
Technique:	Prepare existing process models
Result:	Consolidated and complete set of hierarchical process models, modeling conventions

Fig. 3. Method Fragment 1

4.2 Exploration

After identifying as many fragments as possible in this first step, exploration is used to find additional fragments on the basis of the elements used in existing approaches. Thus, concrete activities, roles, techniques and results found in different sources are extracted and subsequently used to design new fragments. A comprehensive overview of elements cannot be provided herein, but examples for these constituting elements of methods will be given in the following. The most important sources are the literature on service identification and the two case studies described in section 2.

Activities: Many approaches use activities such as *service analysis* and *service categorization*. *Preparation* is also a common activity to be found in literature. However, activities like *goal definition* or *develop SOA strategy* can be found in only one approach, respectively. Some authors use the word activity for very detailed descriptions of how something has to be done. In the definition used herein this would rather be a technique. Furthermore, one and the same activity might have different names in different approaches. This makes consolidation a difficult task.

Roles: Despite their importance, roles only occur in four of the seven compared approaches. Sometimes they are hard to identify as such because the notion of *consumer view* might be used where *employee of the business department* would be a better description. Besides the *employee of the IT department*, roles like *project manager* were important in the case studies. Related literature additionally suggest new roles such as a *service design unit* for certain activities [33].

Techniques: Consolidating all techniques utilized in literature is difficult since most approaches offer plenty of techniques with sometimes overlapping components and scopes. However, there seem to be some typical and wide-spread techniques that are common to many approaches such as *decomposition of business processes* and *asset analysis*. A couple of other techniques that were frequently encountered are *goal service modeling* and *use case modeling*. It is noteworthy that the scope of the listed techniques can differ considerably. Using a *governance questionnaire* is a straight forward and unambiguous procedure with limited scope and little room for interpretation. The *decomposition of business processes* is much more complex and likely to yield different outcomes depending on who actually conducts the task.

Results: Similar to the significant number of techniques, there are many results presented as part of the methods. These results are outputs of respective activities and techniques and can be an input for the next activity. Thus, they are a crucial link between method fragments. The results themselves are quite different in nature and reach from technical *interface descriptions* to comprehensive *SOA strategy documents* or *network models* on an inter-organizational level. *Use cases*, *reference processes* and *activity diagrams* are examples for other results.

Fragment 2: IT Governance Analysis	
Description:	An organization can have manifold demands when it comes to implementing new IT infrastructures. Using agile methods for example could be one imperative. Technical restrictions, programming language, interfaces or naming conventions can impose restrictions on IT projects. Hence, these IT governance issues can be covered by a questionnaire and are incorporated in this fragment. Employees of the business and the IT department jointly forming a so-called service design unit (SDU) form an important role to successfully design services based on this questionnaire.
Input:	Information on IT governance, IT strategy, SOA strategy
Preconditions:	Derived from an organization's strategy, the IT strategy must be defined and documented before an analysis for service identification can be performed. Often, conventions and principles regarding the implementation and development of IT are not explicated in readily available documents. Therefore, it might be necessary to interview (IT) managers in order to retrieve necessary information.
Taken from:	Klose, Knackstedt and Beverungen (2007), Kohlborn, Korthaus, Chan and Rosemann (2009), Arsanjani, Ghosh, Allam, Abdollah, Ganapathy and Holley (2008), Kohlmann and Alt (2007), Alter, Bömer and Goeken (2009)
Design:	Since the scope of this fragment quite wide, elements have been selected from different approaches. Domain decomposition for instance could be found in three existing methods. Special roles and techniques such as an SDU or a governance questionnaire, respectively, have been taken from related literature that does not present a comprehensive method for service identification but deals with governance aspects in general.
Activity:	Service design
Role:	SDU, employee of the IT department, (IT) manager
Technique:	Governance questionnaire, domain decomposition, naming
Result:	Naming conventions, service ownership list, modeling conventions, design principles

Fig. 4. Method Fragment 2

Based on the elements identified previously, more fragments can be designed. Elements that are used in fragments created through exploration are taken from more than one existing approach because if they were to be found in one single approach, the fragment could have been derived by decomposition as shown in section 4.1. Fragment 2 is one example for a fragment designed by exploration.

5 Conclusion and Further Research

This paper outlined the necessity of designing situation-specific methods for service identification since a literature review attested a missing configurability of existing approaches. Hence, a meta method should guide the engineering of such situational methods. An important part of this meta method is the definition of situations in the domain of service identification. Thus, the identification and discussion of context factors and SOA implementation goals was the centerpiece of this work. The idea of how to design method fragments was explained briefly and shown at two examples.

On the way to creating a meta method for the construction of situation-specific methods for service identification there are a number of limitations that should be considered. The identified context factors are based on an extensive literature research. Moreover, their significance was supported by two case studies where qualitative research methods were used. An investigation of relationships and interdependencies of these context factors and the SOAIG is subject to ongoing research.

The number of situations has to be restricted to make the approach feasible. Following [10], the context factors, their parameter values and SOA implementation goals presented herein lead to 17,280 situations. Therefore, the relevance of context factors should be scrutinized through further research. The same is true for parameter values and SOA implementation goals. The aforementioned analysis of interdependencies is also likely to reduce the number of context factor parameter values and thus the number of situations that have to be considered.

Decomposed method fragments extracted from existing methods usually have been applied to real-life projects before and are thus quite reliable. Those fragments created through exploration could be criticized for being an arbitrary combination of elements without proper foundation. Both the exemplary fragments presented in section 4 and further ones that are currently developed are based on literature and the experience of two case studies. Proving quality is difficult and indeed, completeness of a method fragment base that is proposed by many authors [45] cannot be guaranteed. Thus, it is important to feed back information from projects that will use the meta method in future. This will improve fragments and give them a stronger empirical grounding.

As demonstrated, ideas from SME can contribute significantly to the field of service identification by supporting the design of situational methods. In order to build a comprehensive meta method, experience and expert knowledge from the service-oriented domain have to be incorporated in this meta method. Further case studies or action research could support an empirical validation of this meta method.

References

1. Börner, R., Goeken, M.: Identification of Business Services - Literature Review and Lessons Learned. In: 15th AMCIS, Paper 162, San Francisco, California (2009)
2. Fitzgerald, B., Russo, N.L., O’Kane, T.: Software Development: Method Tailoring at Motorola. *Communications of the ACM* 46, 65–70 (2003)
3. Aydin, M.N.: Examining Key Notions for Method Adaptation. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) *Situational Method Engineering: Fundamentals and Experiences*, pp. 49–63. Springer, Boston (2007)

4. Harmsen, F., Brinkkemper, S., Oei, H.: Situational Method Engineering for Information System Project Approaches. In: Verrijn-Stuart, A.A., Olle, T.W. (eds.) *Methods and Associated Tools for the Information Systems Life Cycle*, pp. 169–194. Elsevier Science B.V., Amsterdam (1994)
5. Ralyté, J., Rolland, C.: An Approach for Method Engineering. In: *20th International Conference on Conceptual Modelling*, pp. 471–484. Springer, Yokohama (2001)
6. Cossentino, M., Gaglio, S., Henderson-Sellers, B., Seidita, V.: A metamodelling-based approach for method fragment comparison. In: *11th International Workshop on Exploring Modeling Methods in Systems Analysis and Design at CAiSE, Luxembourg* (2006)
7. Sunyaev, A., Hansen, M., Krmar, H.: Method Engineering: A Formal Approach. In: *17th International Conference on Information Systems Development, Paphos, Cyprus* (2008)
8. Agerfalk, P.J., Brinkkemper, S., Gonzalez-Perez, C., Henderson-Sellers, B., Karlsson, F., Kelly, S., Ralyté, J.: Modularization Constructs in Method Engineering: Towards Common Ground? In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) *Situational Method Engineering: Fundamentals and Experiences*, pp. 359–368. Springer, Boston (2007)
9. Bucher, T., Klesse, M., Kurpjuweit, S., Winter, R.: Situational Method Engineering - On the Differentiation of “Context” and “Project Type”. In: Ralyté, J., Brinkkemper, S., Henderson-Sellers, B. (eds.) *Situational Method Engineering: Fundamentals and Experiences*, pp. 33–48. Springer, Boston (2007)
10. Börner, R.: Applying Situational Method Engineering to the Development of Service Identification Methods. In: *16th AMCIS, Paper 18, Lima, Peru* (2010)
11. Brinkkemper, S.: Method engineering: engineering of information systems development methods and tools. *Information & Software Technology* 38, 275–280 (1996)
12. Punter, T., Lemmen, K.: The MEMA-model: towards a new approach for Method Engineering. *Information & Software Technology* 38, 295–305 (1996)
13. Karlsson, F., Agerfalk, P.J.: Method Configuration - Adapting to Situational Characteristics while Creating Reusable Assets. *Information & Software Technology* 46, 619–633 (2004)
14. Kohlborn, T., Korthaus, A., Chan, T., Rosemann, M.: Identification and Analysis of Business and Software Services - A Consolidated Approach. *IEEE Transactions on Services Computing* 2, 50–64 (2009)
15. Inaganti, S., Behara, G.K.: Service Identification - BPM and SOA Handshake. *BPTrends* 3, 1–12 (2007)
16. Goeken, M.: *Entwicklung von Data-Warehouse-Systemen. Anforderungsmanagement, Modellierung, Implementierung*. Deutscher Universitäts-Verlag, Wiesbaden (2006)
17. Mingers, J.: Combining IS Research Methods: Towards a Pluralist Methodology. *Information Systems Research* 12, 240–259 (2001)
18. Darke, P., Shanks, G., Broadbent, M.: Successfully completing case study research: combining rigour, relevance and pragmatism. *Information Systems J.* 8, 273–289 (1998)
19. Benbasat, I., Goldstein, D.K., Mead, M.: The case research strategy in studies of information systems. *MIS Quarterly*, 269–386 (1987)
20. Luthria, H., Rabhi, F.A.: Building the business case for SOA: A study of the business drivers for technology infrastructure supporting financial service institutions. In: Kundisch, D., Veit, D.J., Weitzel, T., Weinhardt, C. (eds.) *FinanceCom 2008*. LNBIP, vol. 23, pp. 94–107. Springer, Heidelberg (2009)
21. Yin, R.K.: *Case Study Research - Design and Methods*, 3rd edn., vol. 5. SAGE Publications, Thousand Oaks (2003)
22. Strauss, A., Corbin, J.M.: *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, Thousand Oaks (1990)
23. Walsham, G.: Interpretive case studies in IS research. *Nature and Method. European Journal of Information Systems*, 74–81 (1995)

24. Eisenhardt, K.M.: Building Theory from Case Study Research. *The Academy of Management Review* 14, 532–550 (1989)
25. DeLone, W.H.: Determinants Of Success For Computer Usage In Small Business. *MIS Quarterly* 12, 50–61 (1988)
26. Sedera, D.: Does Size Matter? Enterprise System Performance in Small, Medium and Large Organizations. In: 2nd Workshop on 3rd Generation Enterprise Resource Planning Systems, Copenhagen, Denmark (2008)
27. Welsh, J.A., White, J.F.: A small business is not a little big business. *Harvard Business Review* 59, 18–32 (1981)
28. Laukkanen, S., Sarpola, S., Hallikainen, P.: Enterprise size matters: objectives and constraints of ERP adoption. *J. of Enterprise Information Management* 20, 319–334 (2007)
29. Becker, J., Knackstedt, R., Pfeiffer, D., Janiesch, C.: Configurative Method Engineering. In: 13th AMCIS, Paper 56, Keystone, Colorado (2007)
30. Klose, K., Knackstedt, R., Beverungen, D.: Identification of Services - A Stakeholder-Based Approach to SOA Development and Its Application in the Area of Production Planning. In: 15th ECIS, St. Gallen, Switzerland, pp. 1802–1814 (2007)
31. Terlouw, J., Terlouw, L., Jansen, S.: An Assessment Method for Selecting an SOA Delivery Strategy: Determining Influencing Factors and Their Value Weights. In: 4th International Workshop on BUSITAL, Amsterdam, The Netherlands (2009)
32. Arsanjani, A., Holley, K.: The Service Integration Maturity Model: Achieving Flexibility in the Transformation to SOA. In: IEEE International Conference on Services Computing (SCC 2006), Chicago, IL, p. 515 (2006)
33. Börner, R., Looso, S., Goeken, M.: Towards an Operationalisation of Governance and Strategy for Service Identification and Design. In: 13th IEEE International EDOC Conference, Auckland, New Zealand, pp. 180–188 (2009)
34. Leyer, M., Moormann, J.: Facilitating operational control of business services: A method for analysing and structuring customer integration. In: 21st ACIS, Paper 42, Brisbane, Australia (2010)
35. Pitts, R.A., Lei, D.: *Strategic Management*, 4th edn. South-Western, Mason (2005)
36. Schermerhorn, J.R.: *Management for Productivity*. Wiley, New York (1993)
37. Erl, T.: *Service-Oriented Architecture - A Field Guide to Integrating XML and Web Services*. Prentice Hall, Upper Saddle River (2004)
38. Heutschi, R.: *Serviceorientierte Architektur*. Springer, Heidelberg (2007)
39. Beverungen, D., Knackstedt, R., Müller, O.: Entwicklung Serviceorientierter Architekturen zur Integration von Produktion und Dienstleistung. *WI* 50, 220–234 (2008)
40. Papazoglou, M.P.: Service-Oriented Computing: Concepts, Characteristics and Directions. In: 4th International Conference on Web Information Systems Engineering, pp. 3–12. IEEE Computer Society, Rome (2003)
41. Kohlmann, F., Alt, R.: Deducing Service Ownerships in Financial Networks. In: 15th AMCIS, Paper 518, San Francisco, CA (2009)
42. Bieberstein, N., Bose, S., Walker, L., Lynch, A.: Impact of service-oriented architecture on enterprise systems, organizational structures, and individuals. *IBM Systems Journal* 44, 691–708 (2005)
43. Henderson-Sellers, B., Ralyté, J.: Situational Method Engineering: State-of-the-Art Review. *Journal of Universal Computer Science* 16, 424–478 (2010)
44. Ralyté, J.: Towards Situational Methods for Information Systems Development: Engineering Reusable Method Chunks. In: International Conference on Information Systems Development (ISD 2004), Vilnius Technika, Vilnius, Lithuania, pp. 271–282 (2004)
45. Deneckère, R., Iacovelli, A., Kornysheva, E., Souveyet, C.: From Method Fragments to Method Services (2009)