

Service Identification Methods: A Systematic Literature Review

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE
provided by DSpace at VU

Department of Computer Science, VU University Amsterdam, The Netherlands

Abstract. Many service identification methods (SIMs) have been proposed to support the determination of services that are appropriate for use in an SOA. However, these SIMs vary in terms of analysis objectives, identification procedures and service hierarchies. Due to the heterogeneity of the SIMs, practitioners often face the difficulty of choosing a SIM that copes with available resources and fits their needs. To gain a holistic view of existing SIMs and to support the selection of the right SIM, in this paper we present the results of a systematic literature review. A total number of 237 studies were examined, of which 30 studies were selected as primary studies. From these studies, we identified different types of inputs, outputs and processes used by the existing SIMs. Based on these results, we created a matrix which can be used in three different ways for practitioners to select among alternative SIMs.

1 Introduction and Research Questions

Service-oriented architecture (SOA) is an architectural enabler for quick response to business changes and effective reuse of software assets [1]. In an on-demand world, many enterprises have adopted SOA in order to gain competitive power [1]. A key factor that determines whether an enterprise really can benefit from adopting SOA is the design of services, including the scope of functionality a service exposes to meet business needs, and the boundaries between services to achieve maximum reusability and flexibility [2].

In the design of services, service identification (SI) is a significant task aiming at determining (based on available resources) services that are appropriate for use in an SOA. Many service identification methods (SIMs) have been proposed from both academia and industry. However, these SIMs differ significantly, ranging from source code extraction to business domain analysis, from bottom-up to top-down strategy and from ontology-based process to guideline-driven process. Accordingly, the inputs and outputs of these approaches vary as well. Some approaches start with business process whereas some others start with domain knowledge (e.g. goals and strategies); some approaches focus on business services whereas others focus on software services.

* The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement 215483 (S-Cube).

Given many SIMs, a common question that practitioners face is how to select the most appropriate SIM that copes with the available resources and fits their needs [2]. Some enterprises, for instance, not only have well defined business process models in place but also well documented goals, business partners and enterprise capabilities (e.g. IT resources) to support its business process. For them, a SIM that takes all of this information into account will be more suitable than those identifying services only based on e.g. business processes.

Despite the comparisons (e.g. [3,4]) presented so far, none systematically searches for all the existing SIMs. As a result, a holistic overview of extant SIMs is missing. Moreover, the criteria used in the existing comparison frameworks cover many aspects, including economic, business, and technical aspects. However, a comparison of the basic elements (such as the inputs, outputs and processes) of the SIMs is currently missing. When selecting SIMs one often starts questioning what is required for using certain SIMs, what can be expected from them and how to carry them out, before addressing other requirements. Without such an overview of the basic elements of SIMs, the selection of SIMs becomes more complicated. Accordingly, the following research questions arise:

- RQ1: *What are the existing SIMs?*
- RQ2: *How do the SIMs differ? This can be elaborated into: RQ2.a what types of inputs do the SIMs start from? RQ2.b what types of services do the SIMs produce? RQ2.c what types of strategies and techniques are used by the SIMs?*

To answer these research questions, we decided to conduct a systematic literature review, which are particularly powerful in collecting and analyzing existing work. It can maximize the chance to retrieve complete data sets and minimize the chance of bias and summarizing the existing evidence [5]. In the remainder of the paper, Sec. 2 reports on the review results; Sec. 3 presents an input-output matrix that aids the selection of SIMs; and Sec. 4 concludes the paper.

2 The Results of the Review

To conduct the review, we followed the guidelines suggested in [5]. In the first step, we obtained 237 articles whose titles or abstracts contain the keywords specified in our search query. After applying the selection criteria, 38 articles were identified as *primary studies* that are relevant for review. By further reviewing their related work (motivated by the fact that an article presenting a SIM most likely discusses other SIMs as related work), we identified 11 more primary studies. Among these two lists of primary studies, 19 articles are duplicates and hence resulting 30 articles as identical primary studies.

¹ Due to the space limitation, we do not present the review protocol that we followed. Interested readers are referred to www.few.vu.nl/~qgu/SIMReviewProtocol.pdf for details, including the specified search query, selected electronic libraries, and strategies for data extraction data synthesis.

2.1 RQ1 What Are the Existing SIMs?

Each primary study presents one SIM and we have identified 30 SIMs altogether. An overview of the SIMs is given in Table 7. As we can see, SIM S1 and SIM S2 (hereafter labeled as S1 and S2) are the pioneering SIMs presented in 2004. The increasing number of SIMs being proposed in the last three years (6 in 2007, 9 in 2008 and 9 in 2009) reveals its increasing importance in the service engineering field.

Most of the SIMs have certain form of validation as shown in Table 7. The best way to validate a method is to put it into practice. Two primary studies describe the experience in using their methods in real life *projects*. Another way of validation is to experiment a SIM in *case studies*, which was adopted by 13 studies. In order to improve their usability, 6 primary studies provide *examples* in explaining how to use the proposed SIMs. For judging their quality, 4 primary studies *evaluate* the method in terms of e.g. survey or comparison. Only five primary studies do not discuss any validation of the proposed SIMs.

2.2 RQ2.a What Are the Different Types of Inputs?

We examined each primary study and extracted information about the input of the SIMs. We found that many SIMs start from the same type of information. For instance, legacy source code and existing software application or components are both existing software assets but in different forms. In the same vein, a collection of business models, requirements, strategies and organizational structures are all about the domain knowledge of an enterprise but describing specific enterprise elements. By comparing the inputs and classifying the ones that share the same type of information, we identified seven types of inputs. These types of inputs and the SIMs that use them are summarized in Table 11.

The classification of the inputs shows that the resources used by SIMs often have a different scope. For instance, types *data*, *feature* and *use case* are more specific than *application domain* or *business process* in that the former can be

Table 1. Types of inputs used in the SIMs

Type of input	Description	SIM	Total
Business process	A collection of related tasks or activities to fulfill a specific business requirement	S3,S10,S12,S15,S16,S17,S23,S24,S25,S26,	10
Application domain	A collection of models or documents that describe the various aspects of an application domain, including enterprise goals or mission, business rules, business processes, business entities, organization structures, etc.	S1,S7,S8,S9,S14,S22	6
Legacy system	Existing software assets of an enterprise. It can be software systems, source code, or the architecture of the existing systems	S4,S5,S20,S27	4
Mix	A mix of type legacy system and other types	S2,S13,S21,S30	4
Data	The information that is processed, exchanged, or produced by business processes	S6,S29	2
Feature	A set of distinctive attributes or characteristics of software systems	S18,S19	2
Use case	A sequence of business functions that human actors benefit from	S11,S28	2

derived from the latter. The number of SIMs using each type of input shows that fewer methods start with more specific or technical information.

Most of the SIMs start from business processes and enterprise level information, taking both the business and its context into consideration. This is in line with the fact that service-oriented design intends to realize software re-use through large grained services especially meant to create business value.

Legacy system is another type of input and is used by four SIMs, as shown in Table 1. These SIMs take a bottom-up approach to examine the architecture (e.g. S5) or source code (e.g. S27) of the existing systems for identifying services.

As we can see, most of the SIMs are either based on domain knowledge at the enterprise level (*top*) or existing systems (*bottom*). When adopting SOA, enterprises rarely start from scratch; instead, very often they need to migrate their existing systems while creating additional services to address new business requirements. Only four SIMs take a meet-in-middle approach to identify services based on a combination of legacy systems and other information type *Mix*, such as domain analysis in S2 and business process models in S30. This low number is contradictory to the comparison of service analysis approaches reported in [3], which pointed out that most approaches postulate a meet-in-the middle strategy. The cause for this contradiction lies in the fact that in our review we selected *only* the SIMs that provide detailed description of the methods. In [3] such a criterion does not apply and many approaches that only conceptually discuss their SI strategies have been selected for comparison. It has been admitted by the authors in [3] that many approaches do fail to go into detail. Despite of the equal importance of existing software assets and business requirements, only few SIMs provide enough guidance on how to take into account both of them.

2.3 RQ 2.b What Are the Different Types of Services Being Produced?

In this section, we discuss the outputs of the SIMs, including the *types of services* produced and how these services are described (*output format*).

Types of services. The general goal of SIMs is to identify services that are valuable, either from business or IT perspective. Each individual SIM has a specific goal in identifying specific types of services. For instance, some SIMs target at services that represent business processes whereas others focus on identifying services that bridge the gap between business services and IT infrastructure. By studying, comparing and synthesizing the data about the objectives of the services produced by the SIMs, we identified 6 types of services that have been explicitly discussed, summarized in Table 2 (note that a SIM may identify multiple types of services).

From a business perspective, we can see that 21 SIMs result in *business services* representing businesses processes, and 12 result in *data services* for business entities. Both of these two types of services are business-related. Because of the nature of services (i.e exposing business functions), it is quite understandable that a large number of SIMs focus on business.

Table 2. Types of outputs produced by the SIMs

Type of output	Description	SIM	Total
Business process service (BS)	A service that has the business logic or represents a business process, including task services, process services.	S3,S7,S8,S9,S10,S11,S12,S13,S14,S15,S17,S18,S19,S21,S22,S23,S24,S25,S26,S28,S29	21
Data service (DS)	A service that represents business centric entities, including information services, entity services	S6,S7,S8,S9,S13,S15,S16,S17,S21,S22,S24,S25	12
Composite Service (CS)	A composition of multiple services.	S7,S8,S11,S12,S13,S14,S15,S17,S21,S25,S28,S29	12
IT service (IS)	A service that represents technology specific functionalities, including application services, software services, utility services and infrastructure services.	S3,S7,S8,S9,S10,S13,S17,S22,S24	9
Web service (WS)	A service that is implemented using the web service technology. This type is orthogonal to the other types	S1,S2,S4,S5,S20,S30	7
Partner service (PS)	A service that is offered to external partners.	S3,S16,S17,S26,S29	5

On the other hand, it is worth noticing that the numbers of SIMs for identifying *IT services* and *partner services* is relatively low and lower than we expected. Business-IT alignment is recognized as a research challenge [6] and the need of integrating technical architecture (e.g. IT infrastructure, data models) with business architecture (e.g. business models, organizational structure) has been widely agreed [12]. As shown in Table 2, all 9 SIMs that do consider IT services also identify business services and more importantly, they pay specific attention to the integration of business and IT. This alignment should, in our opinion, be supported by all the SIMs, which points out a gap in those SIMs lacking support for IT services.

As for *partner services*, we see that only 5 SIMs lead to services that explicitly consider their service providers (SPs) and consumers (SCs). Services are designed, developed, provided, maintained and owned by SPs. The task of SCs is to discover and compose services fulfilling their needs. SI for SPs entails how to identify services that are potentially useful to others. For SCs, it, instead, entails how to identify services for integration purposes. Because of this difference, a SIM should explicitly indicate which role it supports. Unfortunately, most of the SIMs fail to highlight this difference. Despite that the separation of SPs and SCs is considered as one of the main characteristics of SOA, these two roles are often not explicitly considered in service engineering in general, as found by a survey of SOA methodologies that we reported in [7].

Further, 7 SIMs explicitly aim at identifying *web services* without describing any of the types described above. That is why we regard web service as a special type, orthogonal to the others. Interestingly, nearly all of these SIMs (except for S1) rely on legacy systems.

Types of output format. Different SIMs also describe the identified services in multiple ways and in multiple levels of detail. Some SIMs describe their output in terms of a list of services with short descriptions; whereas some others illustrate their output in terms of a model describing the relation between services and

sometimes with other artifacts of the system. To understand better the outputs of the SIMs, we analyzed the ways that the SIMs describe their identified services. As a result, five different ways of describing services have been identified, as summarized in Table 3.

Table 3. Ways of describing outputs produced by the SIMs

Output format	Description	SIM	Total
Informal service specification	Specify the identified services with a list of terms, such as service description, input, output, etc	S22,S7,S21,S29,S6,S9,S11,S18,S19,S25,S28	11
Service model	Model the identified services in terms of diagrams, illustrating the service landscape	S3,S8,S10,S12,S13,S14,S16,S24	8
Formal service specification	Describe the identified services using standard language, such as WSDL	S2,S15,S20,S26	4
Service implementation	Implement the identified services	S4,S5,S27,S30	4
A list of services	List the identified services with several key elements, such as name, operation, etc)	S1,S17,S23	3

From Table 3 we can see that many identified services are described in *informal service specification*. However, different SIMs often use different terms for specifying the identified services. For instance, in S11 services are specified in detail using many terms including functional and non-function description, and technique-related aspects, such as operations and standards. In S6, however, services are specified only in terms of their operation and in/out messages. Some SIMs describe their output using only *a list of services*, without entering their details. While we do not enter the merit of one or the other approach, we observe that there is no unified way for describing services.

A *service model* is used by 8 SIMs, with the purpose of illustrating the relation between the identified services (e.g. S8) and the relation between services and their providers (e.g. S16). Thanks to its power of illustrating relations, a service model is extremely useful for describing composite services (CSs) and partner services (PSs). As shown in the overview of the SIMs (given in Table 7), only 5 (out of 12) SIMs identifying CSs, and 2 (out of 5) SIMs that identifying PSs use the form of a service model to describe the service landscape. In our opinion, a service model should be used more often as long as CSs and PSs are concerned.

Some other formats of output produced by the SIMs are implementation-driven. One of such formats is *formal service specification*, often used to describe services identified under more formal techniques (e.g. algorithms used by S15; another format is *service implementation*, often used when services as executable programs are created by wrapping source codes of legacy systems (e.g. S30).

2.4 R2.c What Types of Strategies and Techniques?

The previous two research questions (RQ2.a and b) mainly focus on *what* is involved in the SIMs. In this section we shall focus on *how* to carry out the SIMs. In this paper, we define *strategy* as the style, approach or plan for identifying services; we define *technique* as the technical procedures or actions taken to accomplish certain tasks defined in a SIM.

Strategies. Hubbers et. al [8] suggested ten common ways (or strategies) for identifying services. To find out if these ten ways are indeed used by the SIMs, we analyzed the data elicited from the primary studies and mapped all of the SIMs on at least one of these ways. We also identified one way (*W11*) that has not been discussed in [8]. All these ways and their use by SIMs are given in Table 4 (note that a SIM may use multiple strategies).

Table 4. Strategies used by the SIMs

Strategy	Description	SIM	Total
W1 (Business Process Decomposition)	Decompose business process models that depict how the work is done within an organization	S3,S7,S10,S12,,S15,S16,S17,S21,S23,S24,S25,S26,S30	13
W2 (Business Functions)	Decompose business function models that depict what an organization does	S1,S2,S8,S9,S11,S18,S19,S22,S28	9
W3 (Business Entity Objects)	Model services according to business object models	S6,S29,S21,S16	4
W4 (Ownership and Responsibility)	Take the ownership of processes into consideration when identifying services	S7,S11,S28	3
W5 (Goal driven)	Decompose a company's goals down to the level of services	S8,S14,S13,S22	4
W6 (Component-Based)	Identifies services based components	-	0
W7 (Existing Supply)	Identify services from the functionality provided by existing legacy applications	S2,S4,S5,S13,S20,S27,S30	7
W8 (Front-Office Application Usage Analysis)	Select a set of applications that support business processes and extracts comparable functions into a single service	-	0
W9 (Infrastructure)	keep the technical infrastructure into consideration when identifying services	-	0
W10 (NFRs)	Use non-functional requirements as the primary input to identify services	-	0
W11 (User interface)	Identify services based on the design of user interface	S15	1

Four strategies (*W6, 8, 9, 10*) are discussed in [8] but have not been used by any of the SIMs that we studied. As explained in [8], W6 has practical difficulties due to the different nature of component and services; W8 might be risky if existing application design is of low quality; and W9 results in services heavily coupled with infrastructure. Due to these issues, it is understandable that the SIMs avoid using these strategies. W10 points out the importance of non-functional requirements (NFRs) in SI as conflicting NRFs might cause redesign. As no SIM relies on W10, further research is required in supporting NFRs.

Interestingly, we also identified a new strategy, *W11 user interface* (not discussed in [8]). User interface (UI) design, an integral part of the software design process, is often considered out of scope of SOA design [9]. However, a UI design helps one to distinguish the purely presentation aspects from data and process aspects and hence aids the identification of services. The use of UI design in SI is regarded as an innovative approach.

Some SIMs use only one strategy (*W1, 2, 3 or 7*) and we call these strategies *primary strategies*; while the others are always used in combination with the primary strategies and we call them *complementary strategies*. The most popular primary strategies are *W1, 2 and 7*, used by 13, 9 and 7 SIMs respectively. The first two are top-down approaches by examining the business requirements while

the last one represents a bottom-up approach by extracting valuable and/or reusable functions from existing applications. In most of the cases, the strategies used by a SIM are directly related to its input. E.g., all the SIMs that use *business process* as their input use *W1* (decompose business process) as its strategy, only *S15* exceptionally uses a combination of *W4* and *W11*.

The complementary strategies are *W4*, *5* and *W11* which are specifically aided by the information about goals, stakeholders and user interfaces. However, this information alone is often not sufficient for identifying services. For instance, in *S15* user interfaces are first designed based on business process decomposition and then services are identified by extracting abstract WSDL specifications from user interface specifications. Obviously using *W11* (user interface) only is not sufficient in this example. The use of these complementary strategies often results in services more business-driven as explained in e.g. [10].

Techniques. After studying the data about the techniques used in the SIMs, we have identified six different types of techniques, summarized in Table 5.

Table 5. Techniques used by the SIMs

Technique	Description	SIM	Total
Algorithm	A formal approach to problem solving, such as heuristic or formalized rules	S1,S2,S3,S5,S8,S11,S15,S16,S17,S25,S26	11
Guidelines	A set of pre-defined regulations, such as criteria or policies; suggested but not codified	S7,S9,S10,S14,S28,S29,S30	7
Analysis	A process of studying, interpreting, reasoning and synthesizing data	S4,S12,S13,S18,S21,S22	6
Ontology	A technique to conceptually represent (domain) knowledge	S24,S23,S19,S27	4
Pattern	Defined as recurring solution to recurring problems	S6	1
Information manipulation	A text process techniques for identifying or eliciting useful information, such as information retrieval or textural similarity analysis	S20	1

Some of them are more formal, in the sense that they formally codify formulas or rules to specify the way that services are identified, such as *algorithm*, *ontology*, *pattern* and *information manipulation*. Nearly half of the SIMs use these formal techniques and thanks to the advantage of codification, some SIMs partially automate the SI process. For instance, in *S17* a tool called SQUID was developed to automate the process of SI and in *S2* executable algorithms are used to analyze the legacy code.

A less formal technique is *guidelines*, which is used by 7 SIMs as shown in Table 5. These SIMs provide advices like how to identify the right-grained services from goal-scenario models (*S14*) and how to map tasks in business process models to services (*S30*).

Different from these relatively formal techniques, *analysis* is a technique that is more abstract and requires its users to deeply understand the problem they face and make motivated decisions. Accordingly, the subjectivity of using the technique is relatively high and different actors may achieve different results by applying the same SIM.

3 An Input-Output Matrix for the Selection of SIMs

The variety in the types of inputs, outputs and processes discussed in Sec. 2 explains why practitioners often face difficulties to select a SIM that both fits their needs and copes with the available resources. To help compare and select among the SIMs, we use these results (summarized in Table 7) to create an input-output matrix. The matrix is presented in Table 6, where rows represent the types of outputs produced by the SIMs and columns represent the types of inputs being used. Each cell of the matrix describes a specific SIM (in terms of its output format, strategy and technique) if it uses the input and produces the output represented by the column and row respectively. For instance, a SIM that uses *Application domain* as its input (column 1) and produces *Business services as well as composite service (BS+CS)* (row 2) is S14, whose output is described in terms of a service model (*SM*), uses *goal driven (W5)* as its strategy and *guidelines* as technique. In the following, we shall explain how this matrix aids the selection of SIMs in three different ways.

Selection driven by the targeted output. Sometimes, before performing the task of SI, it is expected that certain types of services are identified. For instance, an enterprise that focuses on improving the efficiency and maintainability of its business processes may be interested in business services; while an enterprise that intends to expose its business functions to other partners for business collaboration might be also interested in partner services. In our input-output matrix, the SIMs are classified horizontally, according to the types of services they produce. When the target types of services to be identified are decided, one can use the matrix to eliminate those SIMs that are irrelevant to the needs. Suppose partner services (*PS*) are of great importance, one can see from our matrix that five SIMs (row 8 to 12) could be selected. By determining more types of services to be identified (e.g. *DS*), one can further narrow down the number of candidate SIMs (e.g. S17 and S16 at row 10 and 12). Further, the matrix also provides a straightforward view on what types of inputs are needed if certain SIMs are selected. For example, we can see from the matrix that to identify partner services, either business processes or business centric entities (*data*) should be known. This helps one to judge the feasibility of the SIMs.

Selection driven by the available resources. Knowing what kinds of resources are available for SI, one can also choose a SIM based on its starting point. In the input-output matrix, the SIMs are classified vertically, according to the types of inputs they start with. Using the matrix, one can have an overview of what types of services can be produced given the available resources and at the same time find out the SIMs that support these resources. For instance, if the only available resource is a set of *use cases* (see Table 6 column 7) describing some business functions, one can find from our matrix that two SIMs, S11 and S28 (column 7, row 2) start SI from this resource. Accordingly, one can also expect that business services and their compositions (*BS+CS*) can be identified using either of these two SIMs.

Table 6. Input-output matrix of the SIMs

In \ Out.	1.Application domain	2.Business process	3.Legacy system	4.Mix	5.Data	6.Feature	7.Use case
1.BS		S23(List, W1, Onto)				S18(ISP, W2, Anal) S19(ISP, W2, Onto)	
2.BS+CS	S14(SM, W5, Gline)	S12(SM, W1, Anal)					S11(ISP, W2-&4, Algo) S28(ISP, W2-&4, Gline)
3.BS+DS+CS		S15(FSP, W1&11, Algo) S25(ISP, W1, Algo)		S21(ISP, W1&3, Anal)			
4.BS+DS+IS	S9(ISP, W2, Gline) S22(ISP, W2&5, Anal)	S24(SM, W1, Onto)					
5.BS+DS+IS+CS	S7(ISP, W1&4, Gline) S8(SM, W2&5, Algo)			S13(SM, W5&7, Anal)			
6.BS+IS		S10(SM, W1, Gline)					
7.DS					S6(ISP, W3, Patt)		
8.PS+BS		S26(FSP, W1, Algo)					
9.PS+BS+CS					S29(ISP, W3, Gline)		
10.PS+BS+DS+IS+CS		S17(List, W1, Algo)					
11.PS+BS+IS		S3(SM, W1, Algo)					
12.PS+DS		S16(SM, W1&3, Algo)					
13.WS	S1(List, W2, Algo)		S4(SI, W7, Anal) S5(SI, W7, Algo) S20(FSP, W7, Infor) S27(SI, W7, Onto)	S30(SI, W1&7, Gline) S2(FSP, W2&7, Algo)			

Legenda: **Output format:** List - A list of services, ISP: Informal service specification, FSP - Formal service specification, SM - Service model, SI - Service implementation; **Technique:** Algo - Algorithm, Gline - Guideline, Onto - Ontology, Anal - Analysis, Patt - Pattern, Info - Information manipulation

Selection by comparison of alternative SIMs. Some SIMs can be seen as alternative methods when they share the same type of inputs and outputs. Despite of these commonalities, these methods often differ in the way that the outputs are described, and/or the strategy and technique they use. Using the input-output matrix, one can easily pinpoint and compare these alternative methods since they are grouped and located in the same cell of the matrix. For instance, given *legacy system* (column 3) as starting point and *web services* (row 13) as output, the matrix shows four SIMs: S4, S5, S20 and S27. Comparing these four SIMs in terms of their output formats, strategies and techniques, we can see that the main difference lies in the techniques they use. Therefore, one can choose among these four SIMs based on their preference of one technique over another, depending available technologies, competencies in place, etc. To give another

example, given *application domain* (column 1) as starting point and *BS+DS+IS* (row 4) as output, our matrix shows two SIMs: S9 and S22. By comparing these two SIMs, it is also easy to see that S22 uses *W5 (goal driven)* complementary to *W2 (business functions)* as its strategy; and thereby an enterprise that have clearly defined business goals might opt to select S22 over S9. However, if the enterprise prefers to follow guidelines to provide relatively objective results than to perform analysis to produce relatively subjective results, it might select S9 over S22. As such, our matrix provides a way to preliminarily select alternative SIMs before more comprehensive comparison (if needed).

4 Conclusion

In this paper, we report the classification and comparison of 30 SIMs identified from a systematic literature review. The many different types of inputs, outputs and processes of the SIMs show a significant heterogeneity. Our results provide a holistic overview of the SIMs and highlight their differences. To help practitioners compare and select from these SIMs, we created an input-output matrix that aids the selection of SIMs in three different but complementary ways.

Further, the findings of this review outlines future research directions to further improve the existing SIMs and to guide the design of new SIMs. Our main observations are 1) *IT services* that leverage business processes and underlying IT infrastructure require more attention (Sec. 2.3); 2) Services for internal use and external consumption should be differentiated due to their different characteristics (Sec. 2.3); and 3) NFRs should be explicitly considered due to their importance through the entire service life cycle (Sec. 2.4).

References

1. Bieberstein, N., Bose, S., Fiammante, M., Jones, K., Shah, R.: Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap. Prentice-Hall, Englewood Cliffs (2005)
2. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice-Hall, Englewood Cliffs (2005)
3. 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(1), 50–64 (2009)
4. Boerner, R., Goeken, M.: Service identification in SOA governance literature review and implications for a new method. In: *Int. Conference on Digital Ecosystems and Technologies*, pp. 588–593 (2009)
5. Kitchenham, B.: Guidelines for performing systematic literature reviews in software engineering (2007)
6. Gu, Q., Lago, P.: Exploring service-oriented system engineering challenges: A systematic literature review. *SOCA* 3(3), 171–188 (2009)
7. Gu, Q., Lago, P.: A service aspects driven evaluation framework for service-oriented development methodologies (under submission, 2010)
8. Hubbers, J., Ligthart, A., Terlouw, L.: Ten ways to identify services. *The SOA Magazine* (2007)

9. Mani, S., Sinha, V.S., Sukaviriya, N., Ramachandra, T.: Using user interface design to enhance service identification. In: *Int. Conference on Web Services*. IEEE, Los Alamitos (2008)
10. 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: *ECIS*, pp. 1802–1814. University of St. Gallen (2007)
11. Jain, H.K., Zhao, H., Chinta, N.R.: A spanning tree based approach to identifying web services. *Int. J. Web Service Res.* 1(1), 1–20 (2004)
12. Zhang, Z., Yang, H.: Incubating services in legacy systems for architectural migration. In: *Asia-Pacific Soft. Eng. Conf.*, pp. 196–203. IEEE CS, Los Alamitos (2004)
13. Wang, Z., Xu, X., Zhan, D.: Normal forms and normalized design method for business service. In: *Int. Conference on e-Business Eng*, pp. 79–86. IEEE CS, Los Alamitos (2005)
14. Chen, F., Li, S., Chu, W.C.C.: Feature analysis for service-oriented reengineering. In: *Asia-Pacific Soft*, pp. 201–208. IEEE Computer Society, Los Alamitos (2005)
15. Zhang, Z., Liu, R., Yang, H.: Service identification and packaging in service oriented reengineering. In: *Int. Conference on Software Engineering and Knowledge Engineering*, pp. 241–249 (2005)
16. Baghdadi, Y.: Reverse engineering relational databases to identify and specify basic web services with respect to service oriented computing. *Information Systems Frontiers* 8(5), 395–410 (2006)
17. Chaari, S., Biennier, F., Favrel, J., Benamar, C.: Towards a service-oriented enterprise based on business components identification. In: *Enterprise Interoperability II*, pp. 495–506 (2007)
18. Kohlmann, F., Alt, R.: Business-driven service modeling - a methodological approach from the finance industry. In: *Int. Working Conference on Business Process and Services Computing* (2007)
19. Inaganti, S., Behara, G.K.: Service identification: BPM and SOA handshake (2007)
20. Kim, Y., Doh, K.G.: The service modeling process based on use case refactoring. In: Abramowicz, W. (ed.) *BIS 2007*. LNCS, vol. 4439, pp. 108–120. Springer, Heidelberg (2007)
21. Amsden, J.: Modeling SOA: Part 1. service specification. *IBM Dev. Works* (2007)
22. Fareghzadeh, N.: Service identification approach to SOA development. *World Academy of Science Engineering and Technology* 35 (2008)
23. Kim, S., Kim, M., Park, S.: Service identification using goal and scenario in service oriented architecture. In: *APSEC*, pp. 419–426. IEEE, Los Alamitos (2008)
24. Jamshidi, P., Sharifi, M., Mansour, S.: To establish enterprise service model from enterprise business model. In: *Int. Conf on Services Computing*. IEEE, Los Alamitos (2008)
25. Dwivedi, V., Kulkarni, N.: A model driven service identification approach for process centric systems. In: *Congress on Services*, pp. 65–72. IEEE CS, Los Alamitos (2008)
26. Lee, J., Muthig, D., Naab, M.: An approach for developing service oriented product lines. In: *Int. Software Product Line Conf.*, pp. 275–284. IEEE, Los Alamitos (2008)
27. Kang, D., Song, C.Y., Baik, D.K.: A method of service identification for product line. In: *ICCIT*, pp. 1040–1045. IEEE, Los Alamitos (2008)
28. Aversano, L., Cerulo, L., Palumbo, C.: Mining candidate web services from legacy code. In: *Int. Symposium on Web Site Evolution*, pp. 37–40 (2008)

29. Cho, M.J., Choi, H.R., Kim, H.S., Hong, S.G., Keceli, Y., Park, J.Y.: Service identification and modeling for service oriented architecture applications. In: *Int. Conference on Software Engineering, Parallel and Distributed Systems*, pp. 193–199. World Scientific and Engineering Academy and Society (2008)
30. Bianchini, D., Cappiello, C., De Antonellis, V., Pernici, B.: P2S: A methodology to enable inter-organizational process design through web services. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) *CAiSE 2009*. LNCS, vol. 5565, pp. 334–348. Springer, Heidelberg (2009)
31. Yousef, R., Odeh, M., Coward, D., Sharieh, A.: BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures. In: *ICADIWT*, pp. 50–55 (2009)
32. Azevedo, L.G., Santoro, F., Baiao, F., Souza, J., Revoredo, K., Pereira, V., Herlain, I.: A Method for Service Identification from Business Process Models in a SOA Approach. In: *Enterprise, Business-Process and Information Systems Modeling*, pp. 99–112 (2009)
33. Kim, Y., Doh, K.G.: Formal identification of right-grained services for service-oriented modeling. In: Vossen, G., Long, D.D.E., Yu, J.X. (eds.) *WISE 2009*. LNCS, vol. 5802, pp. 261–273. Springer, Heidelberg (2009)
34. Chen, F., Zhang, Z., Li, J., Kang, J., Yang, H.: Service identification via ontology mapping. In: *Int. Computer Software and Applications Conference*. IEEE, Los Alamitos (2009)
35. Huayou, S., Yulin, N., Lian, Y., Zhong, C.: A service-oriented analysis and modeling using use case approach. In: *Int. Conference on Computational Intelligence and Software Engineering*, pp. 1–6 (2009)
36. Yun, Z., Huayou, S., Yulin, N., Hengnian, Q.: A service-oriented analysis and design approach based on data flow diagram. In: *Int. Conference on Computational Intelligence and Software Engineering*, pp. 1–5 (2009)
37. Ricca, F., Marchetto, A.: A “quick and dirty” meet-in-the-middle approach for migrating to SOA. In: *ACM IWPSE-Evol.* (2009)

Table 7. An overview of the existing SIMs (Appendix)

SIM	Year	Type of input	Strategy	Output format	Type of output	Technique	Validation
S1 [11]	2004	Application domain	W2	WS	A list of services	Algorithm	Evaluated
S2 [12]	2004	Mix	W2&7	WS	Formal service specification	Algorithm	Evaluated
S3 [13]	2005	Business process	W1	PS+BS+IS	Service model	Algorithm	No
S4 [14]	2005	Legacy system	W7	WS	Service implementation	Analysis	Case study
S5 [15]	2005	Legacy system	W7	WS	Service implementation	Algorithm	Case study
S6 [16]	2006	Data	W3	DS	Informal service specification	Pattern	No
S7 [10]	2007	Application domain	W1&4	BS+DS+IS+CS	Informal service specification	Guidelines	Case study
S8 [17]	2007	Application domain	W2&5	BS+DS+IS+CS	Service model	Algorithm	No
S9 [18]	2007	Application domain	W2	BS+DS+IS	Informal service specification	Guidelines	Project
S10 [19]	2007	Business process	W1	BS+IS	Service model	Guidelines	No
S11 [20]	2007	Use case	W2&4	BS+CS	Informal service specification	Algorithm	Example
S12 [21]	2007	Business process	W1	BS+CS	Service model	Analysis	Example
S13 [22]	2008	Mix	W5&7	BS+DS+IS+CS	Service model	Analysis	Case study
S14 [23]	2008	Application domain	W5	BS+CS	Service model	guidelines	Case study
S15 [9]	2008	Business process	W1&11	BS+DS+CS	Service implementation	Algorithm	Case study
S16 [24]	2008	Business process	W1&3	PS+DS	Service model	Algorithm	Evaluated
S17 [25]	2008	Business process	W1	PS+BS+DS+IS+CS	A list of services	Algorithm	Example
S18 [26]	2008	Feature	W2	BS	Informal service specification	Analysis	Case study
S19 [27]	2008	Feature	W2	BS	Informal service specification	Ontology	Case study
S20 [28]	2008	Legacy system	W7	WS	Service implementation	Information manipulation	Project
S21 [29]	2008	Mix	W1&3	BS+DS+CS	Informal service specification	Analysis	Example
S22 [3]	2009	Application domain	W2&5	BS+DS+IS	Informal service specification	Analysis	No
S23 [30]	2009	Business process	W1	BS	A list of services	Ontology	Case study
S24 [31]	2009	Business process	W1	BS+DS+IS	Service model	Ontology	Case study
S25 [32]	2009	Business process	W1	BS+DS+CS	Informal service specification	Algorithm	Case study
S26 [33]	2009	Business process	W1	PS+BS	Service implementation	Algorithm	Evaluated
S27 [34]	2009	Legacy system	W7	WS	Service implementation	Ontology	Case study
S28 [35]	2009	Use case	W2&4	BS+CS	Informal service specification	Guidelines	Example
S29 [36]	2009	Data	W3	PS+BS+CS	Informal service specification	Guidelines	Example
S30 [37]	2009	Mix	W1&7	WS	Service implementation	Guidelines	Case study