SOCA (2014) 8:199–219
DOI 10.1007/s11761-014-0161-y

ORIGINAL RESEARCH PAPER

A systematic survey of service identification methods

Rosane S. Huergo · Paulo F. Pires · Flavia C. Delicato · Bruno Costa · Everton Cavalcante · Thais Batista

Received: 18 October 2013 / Revised: 27 May 2014 / Accepted: 10 June 2014 / Published online: 1 July 2014

Abstract One of the major challenges for the adoption of the service-oriented architecture (SOA) is the service identification phase that aims to determine which services are appropriate to be implemented. In the last decade, several *service identification methods* (SIMs) were proposed. However, the service identification phase still remains a challenge to organizations due to the lack of systematic methods and comprehensive approaches that support the examination of the businesses from multiple perspectives and consider service quality attributes. This work aims to provide an overview of existing SIMs by detailing which service's perspectives, stated as relevant by the industry, are addressed by the SIMs and also by synthesizing the identification techniques used by them. We have performed a systematic survey over publications about SIMs from 2002 to June 2013, and 105 studies were selected. A detailed investigation on the analyzed SIMs revealed that the identification techniques applied by them have a correlation on how they address many of the service's perspectives. In addition, they are supporting the SOA adoption by handling many perspectives of the OASIS' reference architecture for SOA. However, most of them do not explicitly address service quality attributes and few studies support the evaluation of both. Therefore, future research should follow the direction toward hybrid methods with mechanisms to elicit business and service's quality attributes.

Keywords Service-oriented architecture · SOA · Service identification method · SIM · Systematic survey

1 Introduction

Organizations are in a competitive environment. In today's dynamic markets, the pressure to improve quality and productivity and to maintain competitive advantages has made the adaptability to new business requirements a critical issue to the survival of organizations. The Information Technology, as an important tool for organizations, has also followed this trend. Therefore, the construction of interoperable services, which can be organized in a flexible way to quickly meet business needs as described in the service-oriented architecture (SOA), became a promising alternative to be considered.

The process of service-oriented modeling and architectural design consists of three general steps, namely *identification*; *specification*; and *realization* of services, components and workflows [5]. The *identification* step (the main subject of this study) aims to determine which services are appropriate to be implemented in a SOA. Erl [16] defines three possible strategies for service identification. One is the *top–down* strategy, which advocates the completion of an inven-

R. S. Huergo · P. F. Pires · F. C. Delicato · B. Costa
Programa de Pós-Graduação em Informática,
Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
e-mail: rosanesfair@gmail.com

P. F. Pires (✉)
Departamento de Ciência da Computação, Universidade Federal
do Rio de Janeiro, Rio de Janeiro, RJ
PO Box: 68.530, 21941-590, Brasil
e-mail: paulo.pires@dcc.ufrj.br; paulo.f.pires@gmail.com

F. C. Delicato
e-mail: fdelicato@gmail.com

B. Costa
e-mail: brunocosta.dsn@gmail.com

E. Cavalcante · T. Batista
Programa de Pós-Graduação em Sistemas e Computação,
Departamento de Informática e Matemática Aplicada,
Universidade Federal do Rio Grande do Norte, Natal, Brazil
e-mail: evertonrsc@ppgsc.ufrn.br

T. Batista
e-mail: thais@ufrnet.br

tory analysis (definition of enterprise business model, technology architecture, and service inventory blueprint) prior to the physical design and development of services. The second is the *bottom–up* strategy that is tactically focused and makes the fulfillment of immediate business requirements a priority and the prime objective of the SOA project. The last strategy is *meet in the middle*, which is a combination of both. The top–down strategy is used to promote alignment with business goals or processes, whereas the bottom–up strategy is used to evaluate the existing assets, such as information systems, services repositories, databases, and legacy documentation.

Service identification is one of the most practical phases, and it is a real challenge when designing and implementing a SOA [8]. Besides predicting which services an enterprise will eventually need and defining which functions should be part of each service, service should also take into account different levels of service granularity in order to promote reuse and, at the same time, to provide enough flexibility to enable service composition and orchestration without significant performance loss. Moreover, service identification should also produce a catalog of services that is meaningful to the businesses. In order to address these challenges, it is essential to have a methodology to support examination of the businesses from multiple perspectives.

By considering the importance and the complexity of the identification phase in a SOA process, many *Service identification methods* (SIMs) have been proposed in the recent literature. These SIMs offer different techniques to identify SOA services, such as process decomposition, model-driven approaches, value analysis, source code extraction, and ontology mapping [13]. The ultimate goal of a SIM is to deal with challenges in the service identification phase in order to identify services that have correct functionality, granularity, reuse, and flexibility for service composition and orchestration. In this context, some surveys [9–13,115,116] have been published aiming at providing an overview of the existing SIMs. However, meet in the middle approaches are generally not addressed in these surveys. Moreover, none of them evaluates several perspectives pointed out as relevant to the industry, including standards such as reference architectures [14]. Furthermore, up to date only one survey on SIMs [12] was conducted by adopting the guidelines of a *systematic literature review* (SLR) [1], which provides a methodological, fair analysis of a given subject in a comprehensive and non-biased way. Therefore, all the aforementioned points create a gap that we seek to fulfill with this survey.

This survey intends to (i) take into account the different service perspectives stated by the SOA reference architectures presented in [14] and suggest which of them are relevant to the service identification phase; (ii) provide a comprehensive overview of existing SIMs by detailing the techniques used by them to identify candidate services; and (iii) to shed light on further opportunities for improvements in this field.

The remainder of this paper is structured as follows. Section 2 describes related work. Section 3 defines the systematic steps adopted in the performed survey. Section 4 presents the comparison criteria and reports the obtained results. Section 5 describes the threats to validity. Finally the conclusion is provided in Sect. 6.

2 Related work

Table 1 contains a summary about seven surveys focused on SIMs found in the current literature.

Boerner and Goeken [9] propose six groups of SIM characteristics (*basic characteristics*, *business aspects*, *technical aspects*, *economic aspects*, *components of method engineering*, and *principles of design science research*) that were used to compare five existing SIMs. In turn, Birkmeier et al. [10] define 13 SIM characteristics classified into *foundations* (conceptual design), *model and supporting measures*, and *procedure* (technique). Kohlborn et al. [11] list eight SIM characteristics (*type of services*, *strategies*, *lifecycle covered*, *degree of prescription of the methods*, *validity*, *adoption of existing notations/processes*, *adoption of consumers and providers perspectives*, and *use of service classification*) and compare 30 existing identification methods by using these characteristics.

Gu and Lago [12] select 30 primary studies from a set of 237 examined studies and compare them regarding types of inputs, outputs, and strategies used in service identification. Cai et al. [13] assess 41 studies and propose a classification of high-value activities shared by different identification methods. Zadeh et al. [116] propose a criterion to evaluate SIM inputs regarding their machine readability, the level of interaction details among process, stakeholders, service choreography that they elicit, their level of abstraction, goals coverage, and possibility of being decomposed. Finally, Vale et al. [115] select the most significant criteria in previous surveys and compare 30 SIMs by considering the service granularity and type, strategy, inputs, outputs, activities, research method, validation formalism, economic aspects, and the industry sector in which the method is applied.

These surveys provide a good overview of SIMs and differ in terms of the adopted criteria for selection and comparison. However, it is possible to notice that there is a lack of systematic methods for service identification. Moreover, they propose that the method to be developed has to be configurable depending on the utilization constraints within the organizations (e.g., unavailability of an input or the need of applying the method to small domains), and some of them also suggest that economic aspects and non-functional requirements must be considered.

Table 1 Comparison among existing SIM surveys

Survey	Year of publication	Purpose	Data sources	Years covered	Analyzed papers
Boerner and Goeken [9]	2009	Compare existing SIMs and discuss their shortcomings, in particular economic and governance aspects	Not metioned	2005, 2007, 2008	5
Birkmeier et al. [10]	2009	Analyze the current state of the art about service identification and highlight differences between the investigated approaches	Not mentioned	1984, 2001–2008	15
Kohlborn et al. [11]	2009	Provide a general overview of the SIMs and make a comparison covering business and technical view-points	Not mentioned	2004–2008	30
Gu and Lago [12]	2010	Aid practitioners on selecting the most appropriate SIM by providing an overview of their basic elements	ACM IEEEXplore Web of Knowledge ScienceDirect SpringerLink	2004–2009	30
Cai et al. [13]	2011	Present a comprehensive understanding of service identification and a classification based on the common high-value activities	Not mentioned	2004–2011	41
Vale et al. [115]	2012	Compare SIMs to aid stakeholders to choose the most suitable method in Service-Oriented Product Line Engineering context	Not mentioned	2004–2009	32
Zadeh et al. [116]	2012	Evaluate the inputs that could be selected in SIMs process and determine their level of applicability for small and medium enterprises	Not mentioned	2002–2010	48
This survey	2014	Provide a classification scheme based on a SOA reference architecture and an comprehensive overview of existing SIMs	ACM IEEEXplore DBLP Scopus	2002–June 2013	105

The main differences among these surveys and the survey presented in this paper regard to (i) the research method that was employed; (ii) the inclusion of a significant number of new studies on SIMs, published between 2010 and 2013; and (iii) the adopted criteria for comparison and analysis of existing SIMs.

First, as we have already mentioned, only one of the existing SIM surveys [12] has followed the guidelines of a *SLR* [6], which provides a methodological, fair analysis of a given subject in a comprehensive and non-biased way. Our survey was undertaken with the SLR guidelines suggested by Kitchenham et al. [1,3].

Second, our survey represents an update of the state of the art regarding SIMs. As also presented in Sect. 3.2, 36 studies were published from 2010 to June 2013, thus representing an expressive amount of recent studies that were not considered in the existing surveys. We have also observed that some of these new studies (for instance, [61]) are cited by more than 40 authors, thus indicating significant contributions published in the last years that were addressed in our survey.

Third, this survey compares and evaluates SIMs according to perspectives based on the OASIS' reference architecture for SOA [14]. Such a reference architecture is well known for both academia and industry as it describes several characteristics of a reference SOA environment and assists the SOA understanding and adoption. Many of these characteristics are related to service definition and their analysis may reveal opportunities of improvements to SIMs.

Finally, this survey also presents a greater number of findings when compared to the other surveys as it incorporates researches based on the *meet in the middle* strategy, which was less analyzed in previous surveys. This is an important aspect of the present survey since approaches based on such a strategy are more comprehensive than those based on top–down or bottom–up strategies. Meet in the middle strategies evaluate both business and technical perspectives, and they are more aligned with the enterprise reality when considering existing assets and quickly delivering recognizable benefits without neglecting the fact in which services are designed for reuse and must comply with the business context.

3 Systematic literature review

SLRs are means of evaluating and interpreting all available relevant research to particular research question(s), topic area, or phenomenon of interest, thus aiming to present a fair evaluation of a research topic by using a rigorous methodology. Such a rigorous methodology can be viewed as the main point that differentiates a SLR from a simple, traditional literature review (as it was performed in the related surveys about SIMs in Sect. 2) as it seeks to avoid the maximum of bias throughout the process, thus providing scientific value for the obtained findings. SLRs have been recently viewed as an useful way for dealing with research evidences, thus making it possible to systematically identify, select, analyze, and aggregate them for providing knowledge about a given research topic. They have been commonly used for synthesizing existing work from the literature in a comprehensive and non-biased way and for identifying research challenges and opportunities in the state of the art regarding a research subject.

As proposed by Kitchenham et al. [2], a SLR is structured over a systematic process that is typically divided in three main basic steps (Fig. 1):

1. *Planning*, which defines the research questions, search strategy, selection criteria, data extraction, and synthesis methods to be used, and yields a protocol that will guide the conduction of the whole SLR process;
2. *Conduction*, in which the primary studies are identified, selected, and evaluated according to the established protocol, and;
3. *Reporting* (or *Analysis*), which aggregates extracted information from the relevant primary studies considering the research questions and outlines conclusions from them.

Section 3.1 and 3.2 detail the application of the *Planning* and *Conduction* steps to the systematic survey presented in this paper, whereas Sect. 4 presents the results of the performed SLR (*Reporting* step).

3.1 Planning

In this phase, the goals and protocol of the SLR were defined. Such a protocol consists of a predetermined plan that describes the research questions, how the SLR process itself will be conducted (i.e., the search strategy to be adopted), and establishes the selection criteria and the data extraction and synthesis methods. The research questions must have a clear and well-defined focus as they drive the whole SLR, so that the search procedure must identify the studies that help to answer the research questions, and the data extraction and analysis processes must produce data and knowledge to answer them.

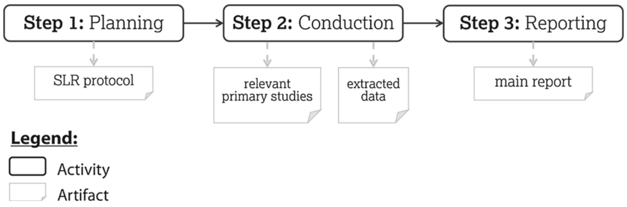

Fig. 1 SLR process

3.1.1 Research questions

Despite many published SIMs, there are gaps toward a comprehensive and systematic method for service identification. Identification methods can vary depending on the availability of inputs (business models, documentation, etc.) and the scope of identification (comprehensive and proactive identification or direct answer to a development project). Therefore, aiming at analyzing previous studies and then summarizing evidences about how existing SIMs work, the following research questions (RQs) were proposed based on these challenges:

RQ1: How do current SIMs address the different service perspectives stated by the SOA reference architecture presented in [14]?

RQ2: Which techniques are used by existing SIMs to identify candidate services?

3.1.2 Search strategy

In order to establish the search strategy based on the defined research questions, three main terms were initially identified, namely *SOA*, *identification*, and *services*. In addition, in order to ensure a greater coverage in the results, we have included the *design* and *analysis* terms, thus resulting in the following search string:

```
service-oriented architecture AND
(service identification OR identify
services OR service design OR service
analysis)
```

Table 2 Electronic databases selected as sources for the search process in the conducted SLR

Database	URL
ACM Digital Library	http://dl.acm.org
IEEEXplore	http://ieeexplore.ieee.org
Scopus	http://www.scopus.com
DBLP Computer Science Bibliography	http://www.dblp.org/search/

in which the main terms were connected by using the AND logical operator and the possible variations by using the OR logical operator.

In order to select the most proper databases for the search process, the following criteria discussed by Dieste et al. [7] were considered: (i) *content update*, i.e., if the publications are regularly updated; (ii) *availability*, i.e., if the full text of the primary study is available, and; (iii) *quality of results*, which is related to the accuracy of the results obtained by the search. As shown in Table 2, four electronic databases were selected based on these criteria and because they are the most commonly used databases in systematic reviews in the Software Engineering domain, as pointed out by Kitchenham and Charters [3] and Dybå et al. [4].

3.1.3 Inclusion and exclusion criteria

Some of the found studies might contain the keywords used in the search string, but are irrelevant to the research questions. Therefore, selection criteria are used to evaluate each primary study obtained by the search procedure according to the defined research questions, thus making it possible to include studies that are potentially relevant to answer them and to exclude studies that do not contribute to answer them.

The considered inclusion criteria (IC) were:

IC1: The study focuses on service identification.
IC2: The study should address different SIMs. If two different studies address improvements on the same SIM, then the most recent is considered.

The established exclusion criteria (EC) were:

EC1: The study is not publicly available in its complete form (full-access).
EC2: The study is not inserted into the SOA context.
EC3: The study is not written in English, which is the most common language in scientific papers.

3.1.4 Data extraction and synthesis methods

The bibliographic details of each selected primary study were recorded by using Jabref [117]. We have also recorded the names of the authors, title of the study, venue (journal, proceedings of conferences, etc.), and year of publication in a spreadsheet for quantitative analysis purposes. Furthermore, in order to extract data from these studies, data extraction spreadsheets related to each research question were built in order to synthesize the results and foster drawing of conclusions.

3.2 Conduction

In this phase, the primary studies were searched, selected, and evaluated according to the established protocol, thus resulting in a set of possibly relevant studies for the SLR. During the search process, the generic search string defined in the *Planning* phase has undergone minor changes in order to make it compatible with the specificities of each electronic database engine. After that, the automated search of primary studies was performed over the selected electronic databases (see Table 2) by searching for primary studies that matched the adapted search string. The performed search procedure was initially limited to title, abstract, and keyword fields and has covered publications from 2002 to June 2013; this start year was chosen by considering the oldest SIM reported by the surveys presented in Table 1. In addition, a cross-reference checking (snowballing) was also conducted by reviewing the related works section of the surveys presented in Table 1 aiming to identify additional studies that are potentially relevant and were not retrieved by the search procedures. These relevant studies found in the cross-reference checking were included in the result set of selected studies.

As summarized in Table 3, 871 studies were retrieved from the electronic databases. From this initial set, 93 studies were removed as duplicate entries between the databases and 692 studies were removed based on the selection (inclusion/exclusion) criteria. All of the retrieved studies had their title, abstract, and keywords read for the evaluation against the selection criteria. In case of doubt, the full text was ana-

Table 3 Number of retrieved studies after the search procedure over the electronic databases

Search mechanism	Results
ACM Digital Library	291
IEEEXplore	152
Scopus	199
DBLP Computer Science Bibliography	229
Total	871

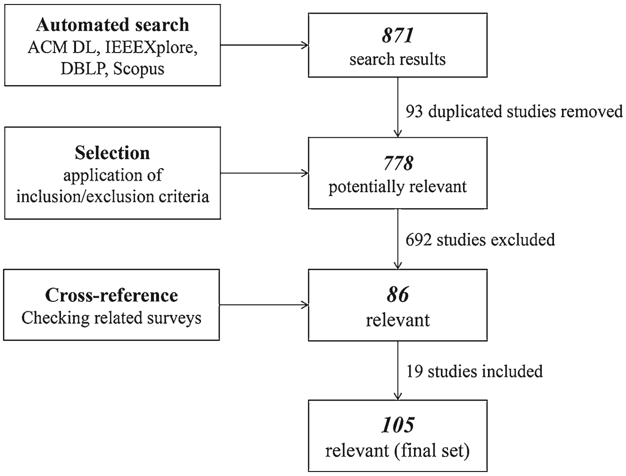

Fig. 2 Selection of relevant studies

lyzed. Many studies were excluded because they used terms such as *service identification* to refer to the process of discovering services that had already been deployed and could be reused, whereas other studies used *service design* to address implementation issues for composing services. Moreover, 19 studies were included after the cross-references checking, thus resulting in a set of 105 primary studies (7 surveys and 98 methods) that were finally considered relevant to this SLR and then selected for data extraction. In this SLR, a given primary study is considered relevant if it does not meet any exclusion criterion and meets at least one inclusion criterion. Figure 2 depicts these steps for selecting the relevant studies.

Figure 3 shows the number of selected studies classified per year of publication. From 2010 to June 2013 (a range that is not covered by the unique systematic survey [12] about SIMs published in 2010), 36 studies were published in these last four years. Therefore, this expressive number of recent studies (34.28 % of the total amount) reveals one of the main contributions of this systematic survey, in terms of updating the state of the art about SIMs.

4 Analysis of the selected studies

4.1 Classification scheme

In order to compare the approaches for identification of candidate services, we have developed a classification scheme based on the OASIS' reference architecture framework for SOA [14]. The proposed classification scheme intends to provide subsides to analyze how perspectives pointed out as relevant by industry are addressed by existing SIMs. The OASIS' reference architecture is an abstract realization of SOA that focuses on the required elements and their relationships to enable SOA-based systems to be used, realized, and governed. It also provides a common language for understanding the important features of SOA, it is independent from any technology, and it has been adopted by industry, thus being an important guide for issues that should be considered during the service identification phase.

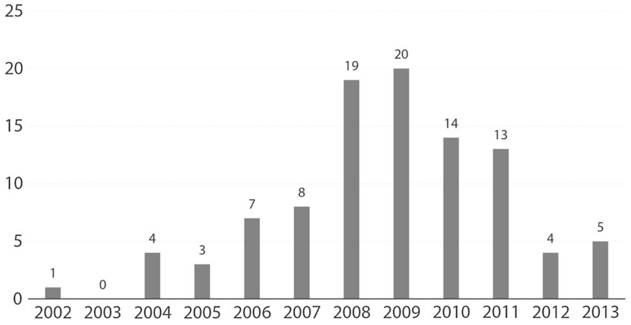

Fig. 3 Number of selected studies per year

The OASIS' reference architecture framework is structured upon *views*. Views are representations of the whole system from the perspective of a related set of concerns. Each view is comprised of *models*, which represent an abstraction or representation of some architectural aspect. Models are mainly described by class diagrams in which each class is an element or a concept involved in a *SOA ecosystem*. A SOA ecosystem is a network of processes and machines that, along with a community of people, creates, uses, and governs specific services [14].

According to the OASIS' framework, three views are used to describe SOA concerns: (i) *Participation in a SOA ecosystem*; (ii) *Realization of a SOA ecosystem*; and (iii) *Ownership in a SOA ecosystem*. The *Participation in a SOA ecosystem* view focuses on the constraints and context in which people conduct businesses by using a SOA-based system. The *Realization of a SOA ecosystem* view focuses on elements that are needed to support the discovery and interaction with services. Finally, the *Ownership in a SOA ecosystem* view focuses on the governance and management of SOA-based systems. Since service identification can crosscut within the software development activities of elicitation, analysis, and project of services, our classification schema only considers the first two views, which are directly related to such activities. Therefore, the third view is out of the scope of this work.

The first two views and their models were analyzed and the main elements and concepts of each model were identified. Figure 4 presents the classification scheme derived from the OASIS' views and models. Table 4 shows each element/concept selected from the reference architecture and the classification perspective that was derived from them. For each element/concept, a classification perspective was derived according to service identification concerns in order

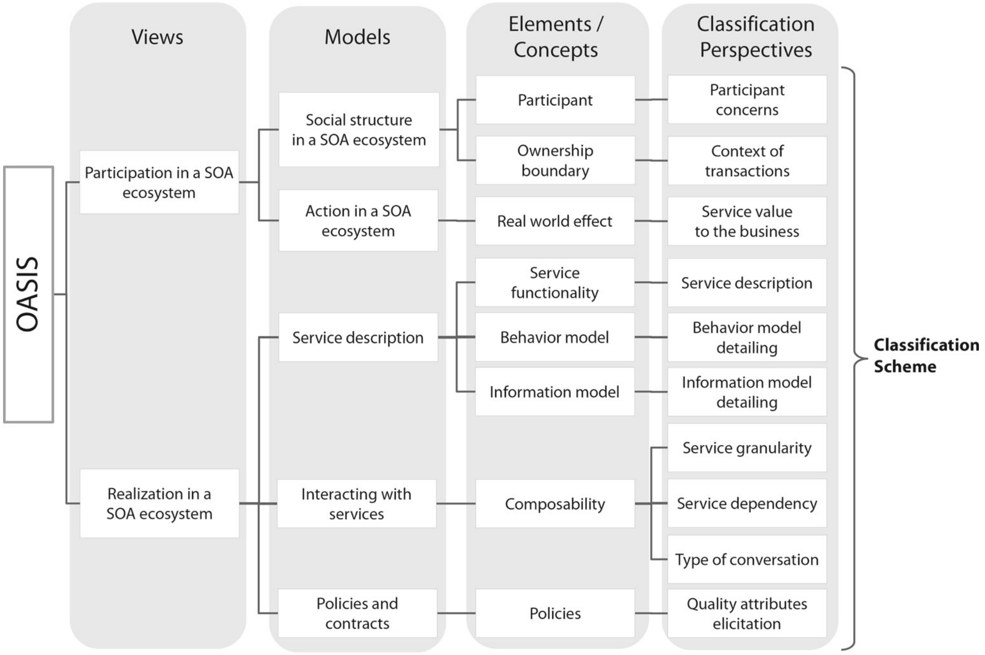

Fig. 4 Classification scheme based on the OASIS' reference architecture for SOA

to compose our classification scheme. For the *Participation in a SOA ecosystem* view, we have chosen the *Participant* and *Ownership boundary* elements, which are part of the *Social structures in a SOA ecosystem* model, and the *Real-world effect* element, which is part of the *Actions in a SOA ecosystem* model. For the *Realization of a SOA ecosystem* view, we have chosen the *Service functionality*, *Behavior*, and *Information* model elements, which are part of the *Service description* model. In addition, we have also chosen the *Composability* element, which is part of the *Interacting with services* model, and the *Policy* element, which is part of the *Policies and contracts* model.

It is important to mention that the *Realization of a SOA ecosystem* view has a fourth model named *Service visibility*. Achieving visibility is one of the key requirements to enable participants to interact with each other in the context of SOA in terms of awareness, willingness, and reachability. Since these concepts are more related to processes aiming at maintaining services descriptions available and to service deployment issues than being related to the identification phase, such a model is out of the scope of this work.

Although one of the purposes of this survey is to analyze perspectives pointed out as relevant by industry, the classification schema proposed illustrated in Fig. 4 is not disconnected from the criteria proposed by the surveys mentioned in Sect. 2. Several perspectives in Table 4 encompass the criteria used by the existing surveys as follows:

– *Participant Concerns* encompasses *Regard to Stakeholders* in [11];
– *Context of Transactions* encompasses *Types of categorization* in [9];
– *Service Value to the business* encompasses *Consideration of Strategic Perspectives* in [9] and *Inputs* in [12,115,116];
– *Behavior Model Detailing* and *Information Model Detailing* encompass *Model Views* in [10] and *Output Format* in [12,115];
– *Service Granularity* encompasses *Granularity* in [9,115], *Service Hierarchy* in [10,11], and *Service Types* in [12, 115];
– *Service Dependency* encompasses *Supported Objects* in [9] and *Dependencies* in [10];
– *Type of Conversation* encompasses *Orchestration vs. Choreography* in [9];
– *Quality Attributes Elicitation* encompasses *Legal Compliance*, *Internal Policies*, and *Service Level Agreements* in [9].

Some perspectives such as *Economic Perspectives* [9, 115], *Method Degree of Detail* [9–11], *Tool Support* [10], *SOA Lifecycle Coverture* [9,11], and *Industry Sector* [9,115] do not have correspondence to our classification criteria, and then, they were not analyzed in this survey. Finally, *Techniques* [9,10,12,13,115] and *Identification Strategy* [9–12,115] perspectives employed in the service identification process cannot be directly correlated to any element of the reference architecture. In particular, the *Techniques* perspective implements the identification strategy and describes the method used to identify service candidates. Nevertheless, due their relevance to the service identification phase and their

Table 4 Classification scheme

Model element or concept (from [14])	Derived classification perspective
A *participant* is an actor or stakeholder that expresses needs and sees these needs fulfilled. A *provider* role is played by a participant that offers a service, and a *consumer* role is played by a participant that interacts with a service in order to fulfill a need.	The *Participant Concerns* perspective analyzes whether the SIM addresses concerns regarding either the service provider or the consumer perspectives. The most common concerns are (i) the identification of business functions with high potential of reutilization and (ii) implementation issues, such as service internal details or guidelines to service realization.
The *Ownership Boundary* is the extent of ownership asserted by a stakeholder over a set of resources. The interaction between the provider and the consumer of a service crosses an ownership boundary when a service is owned by a provider that is different from the consumer.	The *Context of Transactions* perspective describes which interactions are analyzed by the SIMs (across or inside enterprise boundaries).
A *Real-World Effect* describes the result of interacting with a service. The real-world effect is the same regardless the consumer.	The *Service Value to the business* perspective describes the effect of a service to the businesses. Such an effect can be (i) *direct*, when it is related to the business strategy and then directly perceived by the consumer, or (ii) *indirect*, when it is related to technical aspects, and it is commonly used to fulfill a need of a service that produces a direct value to the businesses.
The *Service Functionality* is a clear expression of service functions and the real-world effects of invoking them. Functions represent business activities in some domain that produce the desired real-world effects.	The *Service Description* perspective analyzes which types of descriptions are elicited by the SIMs. Such descriptions can be (i) *syntax-based*, thus hiding what happens inside of the service and exposing input and output values of service interfaces or (ii) *semantic-based*, which describes the service based on semantic languages such as ontology annotation and context information-based methods [15].
A *Behavior Model* comprises an action model and a process model. An *action model* characterizes services actions and a *process model* characterizes the temporal sequence of actions and their dependencies.	The *Behavior Model Detailing* perspective analyzes whether the SIM addresses concerns regarding the internal or external behavior of a service. External behavior focuses on service inputs and outputs, whereas internal behavior focuses on internal actions, events, and conditions.
An *Information Model* describes the syntax and semantics of the messages and data payloads, exception conditions, and error handling in the event of faults.	The *Information Model Detailing* perspective describes whether the SIM elicits the syntax and semantics of the messages and data (detailed business perspective).
Composability is the ability of combining individual services that provide specific business functionality, so that their composition can provide more complex business solutions.	The *Service Granularity* perspective identifies which types of services are elicited by the SIMs. A service can be (i) *atomic*, when it is visible to a consumer through a single interface and is described through a single service description that does not use or interact with other services or (ii) *composite*, when it is visible to a consumer through a single interface and is described through a single service description comprised by the aggregation or composition of one or more other services.
	The *Service Dependency* perspective describes whether the SIM elicits relationships between services and/or resources required or provided by them.
	The *Type of Conversation* perspective identifies types of service collaboration techniques addressed by the SIMs. Collaboration is achieved by (i) *choreography*, which characterizes and composes business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal or (ii) *orchestration*, which composes hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent.
Policies are constraints or conditions of use regarding a service. Policies are potentially applied to many aspects of SOA, e.g., security, privacy, manageability, quality of service, etc.	The *Quality Attributes Elicitation* perspective analyzes whether the SIMs elicits quality attributes that influence design, quality, policies, or execution contexts of the services.

recurrence in the existing SIM surveys, this study analyzes the *Identification Strategy* perspective within the scope of almost all perspectives described in Table 4.

4.2 Results

The following sections detail the achieved findings of the conducted survey by evaluating the perspectives derived from the reference architecture (Sect. 4.2.1) and then detailing the techniques employed in the surveyed SIMs according to the *Technique* perspective (Sect. 4.2.2). Furthermore, we discuss some existing gaps in the field of SIMs (Sect. 4.2.3).

4.2.1 Analysis of the classification perspectives derived from the reference architecture

After defining our classification schema presented in Table 4, the findings were analyzed accordingly to it. Such an analysis intends to identify which of these classification perspectives are addressed by existing SIMs, thus answering RQ1. A summary of how each perspective was addressed is presented in the following subsections.

Participant concerns The *Participant Concerns* perspective evaluates providers' or consumers' concerns addressed by the SIMs. The most common concerns are in this perspective *reutilization* and *implementation* issues. Reutilization concern is addressed in SIMs by the identification of functions or tasks with high potential of reuse, i.e., functions required by several stakeholders inside or outside the organization.

Implementation concern is addressed in SIMs by (i) providing steps and guidelines to service realization and implementation [32,67,85,86,90,114]; (ii) extracting legacy code for service packaging [17,29,33,37,48,52,53,62,64,72,77, 84,97,101–103]; and (iii) identifying of points of variability. *Variability* refers to assumptions about how members of a family of products may differ from each other [40]. The most common types of variability addressed are variability of activities (in which activities can be optional or alternative to accomplish an action) [25,36,40,44,45,47,65,95], variability in the interfaces [36,44,66,95], and product dynamic reconfiguration based on context information [20].

Most of the top–down approaches aim at just identifying reusable services with no implementation concerns [19,21–24,26,27,30,31,34,38,39,41–43,46,49–51,54–56,58–61,63, 67,69,70,74,75,78,80,81,87–89,92,94,98–100,104,105, [106,107,109,111,113] Top–down SIMs that have implementation concerns [20,25,32,40,44,45,47,65–67,90,95] are those that address variability as aforementioned. On the other hand, bottom–up SIMs usually address implementation concerns [17,29,33,37,48,52,53,62,64,72,77,84,97, 101,103,114] since they deal with legacy code extraction and reorganization to define the service packaging.

It is noteworthy that despite the legacy systems evaluation performed by meet in the middle approaches, the majority of them only uses it to detail the business domain [57,91] or to identify if any existing function corresponds to a requirement that a service candidate must fulfill [18,57, 73,76,79,83,108,110] without addressing implementation concerns.

Regarding the techniques detailed in Sect. 4.2.2, only product line, source code analysis, and wrapping tackle implementation concerns. The other techniques only tackle reutilization issues.

Context of transactions The *Context of Transactions* perspective classifies SIMs by taking into account issues related to the distribution of resources and interactions of people and systems inside or outside the enterprise. The explicitness of these boundaries is important to identify the implications of crossing them, especially for analyzing their impact on aspects related to governance and security.

An example of how the *Context of Transactions* perspective is addressed inside of enterprise boundaries is the approach presented in [61], which decomposes business processes and identifies services goals, services interactions with applications or people, and the resources (inputs and outputs) of a service. In turn, the *Context of Transactions* perspective outside enterprise boundaries can be exemplified by the SIM proposed in [19], which uses the service responsibility analysis technique. Such a technique identifies services by using information sharing relationships between organizations as inputs, thus eliciting their responsibilities and rules that govern the exchange of information.

The majority of the studies does not detail interactions of actors (people or systems) with a service [[17,18,20,22–30,32,33,35–37,40,42–48,52–59,62,64–68,71–93,96,97], 98–103,105–108,110–114]. The studies that make such a detailing are top–down approaches that apply *ontologies* to describe services [21,31,61,63,69,70,95] or employ techniques as value analysis [38,39,41,49–51,56,60,94,109] and service responsibility analysis [19]. Ontologies provide means to describe services capabilities, resources, and actors, whereas value analysis and service responsibility analysis techniques identify services from the explicitness of participant's interaction.

SIMs that deal with identification outside the enterprise boundaries [19,34,38,39,41,49–51,56,60,94,104,109] can be adapted to identify services inside enterprise boundaries. This can be accomplished by replacing external economic entities by internal entities, such as departments. Nevertheless, the opposite is harder to be achieved because SIMs that deal with identification inside enterprise boundaries usually use intra-enterprise business process or legacy application as inputs to identify services.

Service value to the business The SOA approach aims to align IT and business perspectives by building services that are business-focused and can be reused and deployed across multiple software applications. The *Service Value to the business* perspective then evaluates whether the effect expected when a consumer interacts with a service is related to business strategy, thus being directly perceived by the consumer or not. Services that are indirectly related to business are usually related to technical aspects or are fine-grained services used to compose other services that offer direct value to the business.

This classification perspective is directly related to the *input* used by the SIM. The majority of the top–down approaches identify services that are both directly and indirectly related to business. The other top–down approaches only identify services directly related to business. Meet in the middle approaches always identify services that are both directly and indirectly related to business, and the bottom–up ones identify services that are indirectly related to business.

Business goals [22,38,49,58,79,81] are part of the business strategy. A SIM can identify services to achieve these goals, thus having direct traceability to the business needs. The same principle applies to business models [19,21,39,41,50,51,56,63,75,80,89,94,104,109,111]. As business models describe the enterprise mission, business requirements, and organizational architecture, they can be used as inputs to identify services that support information exchanges among organizations (e.g., [39]) or inside them (e.g., [63]). On the other hand, legacy code (existing software assets of an enterprise) [17,28,33,35–37,48,52,53,62,64,73,76,77,79,82–85,91,96,97,101–103,110] or documentation [18,29,57,68,71,86,93,108,112] inputs derive services that usually have indirect relation to business. These services correspond to technical aspects or they are fine-grained services used to compose other services (e.g., [29]).

Inputs as business processes [18,23,24,26,27,32,36,40,42–44,54,55,57,59–61,65,67,69,73,74,76,78,83,85–87,90,91,98–100,105,107,108,110,113], features [20,25,45,47,95], requirements [30,31,34,46,66,70,88,92,106,114] or database [72] assets can originate services with both direct and indirect effects. The service effect depends on the process decomposition level. High-level process originates services directly related to the businesses, whereas subsequent decompositions originate more fine-grained services that tend to be indirectly related to the businesses [100]. When using features as inputs, the effect depends on the relevance to the businesses of the feature's product.

Service description A good *service description* is essential to enable service reutilization by matching user requirements against service capabilities. Service descriptions can be started within identification or specification phases. The advantage of describing services in the identification phase is to have a detailed perspective of service capabilities by identifying whether the service really delivers the expected value, and also providing a better input to the specification phase. Descriptions can be *semantic* or *syntax-based*. Semantic descriptions describe a service based on semantically enriched formats, such as ontology annotation [118] and context information-based [20] methods. In turn, syntax-based descriptions hide service implementation details and expose the externally observable service behavior as input and output values of service interfaces [15]. Semantic descriptions convey real-world meaning to the services, so that this type of description is important to provide a clear understanding of the effects resulted from the invocation of a service and a consistent interpretation of the handled data, in particular when the interaction occurs across ownership boundaries. It can also enable automatic service discovery and composition [119].

We have noticed that only 16 studies present semantic descriptions of services [20,21,31,43,55,58,61,63,69–71,75,82,87,94,95]. Among them, only two are bottom–up approaches [71,82] and the other 14 are top–down approaches. This result might reflect an influence of the traditional process modeling and software engineering methodologies that are used for modeling and developing systems, which do not have a strategy to document semantic aspects of information. The studies that semantically describe services usually adopt ontology-based techniques to identify services since ontologies are able to provide semantic classes to organize relevant domain aspects to service description, such as participants, resources, and operations.

Only one study [20] explicitly identifies and describes context information. This is accomplished by defining attributes (as data types and validity conditions) of each identified context parameter and specifying each situation as a logical expression of contextual parameters to enable service dynamic reconfiguration.

Behavior model A *behavior model* is part of the service description, and it is essential to understand and foster the interaction with the service. A well-defined behavior model characterizes knowledge of the (i) actions invoked against the service, (ii) events, and (iii) temporal relationships associated in a service interaction. It should also describe activities involved in a workflow that represents a work unit [14]. Since the service external behavior is dependent of service internal actions, sequence, and events, SIMs should elicit these perspectives to support the behavior model description.

A behavior model description can be done after the service implementation, i.e., outside SIMs scope, but such a description helps to elicit and understand the service scope and to enable assessment of service responsibilities (cohesion) and dependency from other services. Existing SIMs deal with the behavior model in three ways:

- *External behavior only*: The main purpose is to identify the external behavior of a service, but not its internal operation. The focus is on inputs and outputs or on the service purpose (real-world effect resulted from service execution) [38];
- *Action description*: SIMs identify and describe internal service actions, but do not detail their internal behavior [78];
- *Actions and behavior description*: SIMs identify and describe service internal actions and their behavior. Internal action behavior can be described as events, pre- and post-conditions, and actions sequence [87].

All SIMs that have elicited only external behavior were top–down approaches [34,38,39,41,50,51,56,80,94, 104,111]. Among the SIMs that elicit action descriptions, the one in [68] is bottom–up, [91] is meet in the middle, and the other SIMs are top–down [22,25,27,30,40,42–44,47,59,70, 78,102]. In regards to events and conditions elicitation, most of SIMs are top–down approaches [19,21,58,63,69,75,87], except [57] that is a meet in the middle one, and [71] that is bottom–up. Except for these studies, all bottom–up and meet in the middle approaches elicit the sequence of actions for a service [17,18,20,23,24,26,28,29,31–37,45,46,48,49,52–55,60–62,64–67,72–74,76,77,79,81–86,88–90,92,93,95, 96–101,103,105–110,112–114].

Some of the techniques detailed in Sect. 4.2.2 are related to behavior model description. Among the studies that only detail external behavior, the majority uses the value analysis technique [38,39,41,50,51,56,94,104,111]. In the studies that provide internal actions and internal behavior description, techniques such as model-driven ones, ontology mapping, and service responsibility analysis tend to foster the description events, pre- and-post conditions in the behavior model. Platform-independent models (PIMs) are used to identify events and conditions in model-driven approaches [57,58,87,90], whereas classes and relationship types are used in ontology mapping [21,63,69,71,75], and governance rules are used in service responsibility analysis [19]. On the other hand, techniques such as decomposition, requirement analysis, source code analysis, and wrapping tend to focus in the description of internal actions sequence. In these techniques, the sequence of actions is identified by process activities sequence [18,23,24,26,32,54,60,61,67, 73,74,76,79,81,83,85,86,89,98–100,105,107–110,113] in decomposition approaches, by process diagrams [36,46] or use case description [31,66,88,92,106,114] in requirement analysis, and by implementation sequence in source code analysis [28,33,35,48,52,53,62,64,72,82,84,96,97, 101,103,112], as well as in wrapping [17,37,77,102].

Information model As the behavior model, the *information model* is part of the service description. A well-defined service information model describes the syntax and semantics of the messages and data payloads, exception conditions, and error handling in the event of faults [14], thus enabling meaningful exchange of information by matching the model semantics with the semantics of the service consumers.

SIMs deal with the information model in a variety of ways:

- *Not detailed*: Information handled by the service is not identified or it is identified, but is not detailed. The focus is on the service functionalities or on identifying the service purpose and not its internal operation [38].
- *Information structure description*: Information model is detailed with focus on the description of the information structure (classes and attributes) [18].
- *Messages and parameters*: Information model is detailed with focus on identification and description of service messages and parameters structure [37].
- *Semantic description*: Information model is detailed with focus on semantic description of the handled information or service messages [69].

The majority of the studies that do not detail information model [22,23,25–30,34,36,38–44,47,50,51,54,56,59, 65,73,78–81,83,92–96,98–100,104–107,111–113] regards to top–down approaches. Bottom–up approaches detail information structure or messages and parameters [17–21,24,31–33,35,37,37,45,46,46,48,49,52,53,55,55,57,58,60,62,62–64,64,66,66–68,68,69,69,70,70–72,74,76,77,77,84,84–86, 88,88,89,89–91,97,97,101–103,108,109,114]. Finally, meet in the middle approaches do not detail neither information model nor information structure.

In regards to the techniques, we have found some correlation with the information model description perspective. The techniques that traditionally do not focus on describing an information model are (i) pattern matching, which focuses on identifying patterns in processes [42,59] or in legacy code [29]; (ii) product line, which focuses on functionalities performed by services [25,40,44,47,65,95]; and (iii) value analysis [38,39,41,50,51,56,94,104,111], which focuses on identifying what a service should be and not on its internal description.

Another correlation was found with the SIMs that detail the behavior model. SIMs that use syntax-based service descriptions tend to focus on detailing information or message structures, whereas the ones that use semantic-based tend to focus on the semantic description of information and messages or on message structure [21,61,63,69,70,75,82, 87].

Service granularity Services can be *atomic*, i.e., visible to a consumer via a single interface and described via a single service description that does not use or interact with other services, or *composite*, i.e., visible to a consumer via a single interface and described via a single service description comprised by the aggregation or composition of one or more other services [14]. Service composition can be either performed by composing atomic or composite services. When composing services, the business logic is implemented by several services, thus allowing the definition of increasingly complex solutions by progressively aggregating components at higher levels of abstraction. In this perspective, 57 of the considered SIMs deal with both atomic and composite services [17–20,24,26–28,30,31,33,36,37,39,40,45,47–49, 55,56,58,60,61,65,67–72,74–77,79,81,83,85–87,90,92,93, 94–98,100,106–108,110], whereas 41 SIMs specifically deal with atomic services [21,22,25,32,34,35,38,41–43,46, 50–54,57,59,62–64,66,73,78,80,82,84,88,89,91,99,101, 102–105,109,111–114], and only three SIMs specifically deal with composite services [23,29,44].

The study in [28] is an example of how a SIM can address the identification of both atomic and composite services. Such a SIM defines a service layers model, which is a natural composition hierarchy. The definition of which functions are part of a service is done accordingly to the layer responsibility. Therefore, the organization in layers contributes to the identification of services with "right" granularity and cohesion.

It was not possible to identify any correlation between the techniques or approaches with the identification of atomic or composite services as all techniques were able to identify both types of services. The decision of grouping functions in a coarse-grained service or creating only fine-grained services or even creating fine-grained services and composing them seems to be related only to the scope chosen by each SIM.

Service dependency The *Service Dependency* perspective aims to describe whether SIMs identify relations between services or resources required by them. Resources and relationship elicitation is important to provide a broader knowledge of service utilization and operation, thus making service design and implementation easier. SIMs are classified in four categories according to the dependencies that they elicit:

- *None*: SIMs do not identify any relationship between services or resources required by them [38,41,42,50,51,54, 56,59,80,99,104–106,109,111–113];
- *Resources:* SIMs identify resources required by the candidate service as informational entities, databases that store them and legacy code or classes [22,32,35,46,52,53,62–64,66,72,73,82,84,101,102,114];
- *Relationships*: SIMs identify any type of relationships between services as compositions or collaborations to accomplish a purpose [[19,20,23–25,27,29–31,34,39, 40,44,47,49,55,58,65,67,68,70,74,78,81,87,88,90,92], 95,98,107];
- *Both*: SIMs identify both resources required and relationships between services [17,18,21,26,28,33,36,37,43, 45,48,57,60,61,69,71,75–77,79,83,85,86,89,91,93,94, 96,97,100,103,108,110].

All SIMs that do not identify any relationship are top–down approaches. Among the SIMs that use a bottom–up approach, almost all (except two) identify resources required by candidate services or resources and relationships. The majority of the top–down approaches identifies service relationships, whereas the meet in the middle ones identifies both resources required and relationships between services.

The majority of the SIMs elicits relationships between services. This happens because the majority of the techniques used by the SIMs elicits relationships (dependencies) within the several inputs and these dependencies can derive services relationships. Process-oriented approaches as decomposition [18,23,24,26,60,61,67,74,76,79,81,83,85,86,89, 91,98–100,105,107–110,113] and model-driven [27,57,58, 87,90] ones usually identify control flow between activities. These activities (or group of activities) are performed by service compositions and collaborations. Relationships between activities also reflect relationships between services that implement them. Product line identifies services relationships from similarities and relationships between functions [20,25,40,44,45,47,65,95], which can be achieved by service collaborations and compositions. Requirements analysis [30,31,34,36,70,88,92,106,114] and Service-oriented design aspect (SODA) [55,78] identify service collaborations in order to fulfill a requirement or an aspect, and service responsibility identifies dependencies between task and data services [19]. Techniques such as asset identification [18, 57,68,86,93,108,112] and ontology mapping [21,43,63,69, 71,75] focus on the identification of both relationships and resources, whereas source code analysis [28,33,35,36,48, 52,53,62,64,72,73,76,79,82–85,96,97,101,103] and wrapping [17,37,77,102] tend to identify legacy resources (source code, data and existing services) that might be used for implementing a service. Finally, pattern matching [42,59] and value analysis [38,41,50,51,56,104,111] usually do not identify service resources or relationships as the focus is on defining which operations should be grouped within a service or the service purpose, respectively.

Type of conversation Services can be composed in a variety of ways, including direct consumer-to-service interaction by using programming techniques, or they can be aggregated by means of an aggregation approach such as *choreography* and *orchestration*. Choreography is used to characterize and to compose business collaborations based on ordered message exchanges between peer entities in order to achieve a common business goal. In turn, orchestration is used to compose hierarchical and self-contained service-oriented business processes that are executed and coordinated by a single agent [14].

SIMs can detail the type of conversation by means of orchestration and/or choreography or none of them. Most studies do not detail the type of conversation between services [22,24–26,28–35,37,38,40–43,46–48,50–54, 57,59,60,62–64,66–69,71–73,75,78,80–85,88,89,91–93, 95–114]. Among the studies that detail this perspective, the majority regards to top–down approaches. Studies that mention both types of conversation do not detail how service collaborations are identified or implemented [27,45,86]. SIMs that mention choreography identify service collaborations by interactions flows [21,44,79] or define service adapters to enable dynamic composition [36]. Service orchestrations are also identified by interactions flows [17,20,23,39,56,58,61,65,70,74,76,87,90,94] or by defining composition patterns [49,55]. We have not found any correlation between conversation type detailing and the technique or approach employed by the SIM.

Quality attributes elicitation Finally, the *Quality Attributes elicitation* perspective is related to the elicitation of the quality attributes that influence the design, policies, or execution contexts of the services. Quality depends on the stakeholders' requirements, but some general service quality attributes can be identified in a SOA context. Erl [16] emphasizes that the basic software quality design principles of low coupling and high cohesion should be observed during all service creation cycle. Service granularity is also pointed out as a quality attribute because the granularity level of a service can affect its capabilities, performance, reusability, and coupling.

Only ten studies identify quality attributes. SIMs deal with service candidate quality by using metrics of coupling [22,45,100,109], cohesion [22,45,100,101,109], granularity [22,105], modularity [26,33], reusability using the semantic distance between features [25], and QoS [23] by estimating a weight to execute activities. Due to this small number of reported SIMs that address service quality attributes, we can infer that such a perspective still is immature in the context of service identification. Almost all SIMs intend to identify service candidates, but they usually do not assess their quality neither any effort to improve identified candidates. Quality attribute elicitation does not seem to be related to techniques or approaches, thus being a consequence of the scope chosen by the SIM.

4.2.2 Analysis of the techniques employed in the service identification process

In order to answer the RQ2, the studies were categorized in light of software engineering techniques employed in the service identification process and the proposed classification scheme (Sect. 4.1). Aiming to categorize each technique, we have used the *high-value activities* criteria proposed in [13] with some improvements: (i) the addition of *Product line approach* and *Artifact-centric approach* techniques and (ii) the adjustments of the delivery strategy for *Ontology mapping* and *Service classification* techniques in order to include the bottom–up strategy since these techniques can be used with both top–down and bottom–up strategies. Table 5 presents all selected techniques. For each one, we show their description, artifacts generated by them, and the delivery strategy adopted (top–down or bottom–up).

Decomposition is the most used technique with 38 occurrences among the analyzed SIMs, followed by *Source code analysis* (27), and *Value analysis* (13). More than a half of the studies combine two or more techniques. The most used combination is *Decomposition* and *Source code analysis*, which is usually applied in meet in the middle approaches. *Service classification*, *SODA*, *Asset identification*, and *Wrapping* are always used together with other techniques, but only *Service classification* acts as a complementary technique often used to ensure services with right granularity and high cohesion.

Service classification and *Ontology mapping* techniques can be applied in both top–down and bottom–up strategies. Considering all techniques, the top–down is the most used approach, being employed by 56 SIMs, while the bottom–up one is employed by 30 studies. The meet in the middle approach is less used, being employed only by 12 studies.

Table 6 correlates the multiple perspectives of the OASIS' reference architecture for SOA described in Sect. 4.1 with the software engineering techniques employed by the existing SIMs. Table 6 aims to assist practitioners to reason about how the techniques can be applied to the service identification process in the light of the OASIS' reference architecture classification scheme. The techniques presented in Table 5 (except *Service Classification* as it is a complementary technique) were analyzed in order to find out influences or correlations to each perspective. Since we have not found any correlation among *Service Granularity*, *Type of Conversation*, and *Quality Attributes elicitation* perspectives and the software engineering techniques employed by existent SIMs, they are not listed in Table 6. Despite the absence of direct correlations, some relevant observations can be drawn from the analysis of these perspectives. The identification of services with different granularities (*Service Granularity* perspective) can be accomplished by segregating services in layers as suggested in [18]. A precondition to address the *Type of Conversation* perspective is to elicit service relationship. Therefore, a technique that enables this type of identification must be chosen (for instance, a model-driven approach or the value analysis technique). The *Quality Attributes elicitation* perspective can be addressed by defining metrics to assess services quality accordingly to the elicited requirements, independently on the applied technique. The other perspectives are addressed according to the

Table 5 Techniques employed in the service identification process

Technique	Delivery strategy	Description	Artifacts
Model-driven approach	Top–down	Based on generating several business models, requirements, and business rules. Transform the business models to business services.	CIM/PIM, business models, service model
Decomposition	Top–down	Based on decomposing high-level, coarse-grained processes/goals/features, etc. into low-level, fine-grained sub-processes/goals/features, etc.	Sub-processes/ goals/features etc.
Value analysis	Top–down	Based on the analysis of business value and value change with value chain, value network, etc. to decide which business services are of high value.	Value model, high-value services
Pattern matching	Top–down	Identify services from business models by matching existing business patterns.	Business services
Service responsibility analysis	Top–down	Make use of tools, such as service responsibility table to identify service owner and responsibilities.	Service responsibility specification
Product line approaches	Top–down	Identify commonalities or refine variation points from business level or from a portfolio of existing services by specifying the action semantics of the variation and its specific applicability.	Feature model, variability model
Artifact-centric approach	Top–down	Identify services from the information model and the lifecycle of business entities.	Information model, information lifecycle model
Requirement analysis approaches	Top–down	Other activities in requirement engineering for requirement elicitation, such as scenario and gap analysis.	Refined requirements, gap report
SODA (Service-oriented design aspect)	Top–down	Based on the decomposition of interactions, concerns, and features into aspects and composing them according to requirements.	Aspect specification
Assets identification	Bottom–up	Based on the analysis of informative entities (diagrams, schemas, models, documents, etc.) of legacy systems to identify reusable assets.	Reusable assets
Source code analysis	Bottom–up	Based on the analysis of structure, data flow, and dependency in source code.	Software services
Wrapping	Bottom–up	Wrapping existing applications directly as candidate services.	Wrapped services
Ontology mapping	Top–down/Bottom–up	Based on the definition of a set of ontologies to identify business services from business requirements or legacy documentation.	Business services
Service classification/categorization	Top–down/Bottom–up	Based on the classification services to different types such as process service, composite service, and molecular service. to decide the function and granularity of services.	Classified business services

correlations between techniques and classification perspectives presented in Table 6.

As an example of usage of Table 6 to aid practitioners in the service identification phase, consider that a practitioner wants to know which *Participant concerns* can be used from the *Decomposition* technique. Since such perspective presents two concerns, namely *reutilization issues only* and *implementation and realization issues* (see Sect. 4.2.1), Table 6 shows that the answer is *Reutilization only*. Another example is when a practitioner wants to identify services' implementation concerns and the resources required or provided by them, according to the *Service Dependency* perspective (see Sect. 4.2.1). In this case, Table 6 also indicates that (s)he can use *Product line* or *Source code analysis* or *Wrapping* techniques combined with the *Decomposition* or *Ontology Mapping* or *Assets Identification* techniques.

Table 6 can also assist practitioners in the selection of software engineering techniques to be used depending on the type of project they are developing. For example, if a practitioner is in charge of a SOA project focused on delivering fast results, he/she can choose the *Source code analysis* technique because it fosters the elicitation of implementa-

Table 6 Techniques and classification perspectives

Technique/ classification perspective	Participant concern	Context of transactions	Service description	Service value	Behavior model	Information model	Service dependency
Decomposition	Reutilization only	Inside	Syntax-based	Both	Internal actions sequence	No tendency	Both (resources and service relationship)
Model-driven approach	Reutilization only	Inside	Syntax-based	Both	Internal events, pre- and-post conditions	Information messages and parameters	Relationship
Value analysis	Reutilization only	Outside	Syntax-based	Direct	External behavior only	Not detailed	Relationship
Ontology mapping	Reutilization only	Inside	Semantic-based	Both	Internal events, pre- and-post conditions	Semantic description	Both (resources and service relationship)
Pattern matching	Reutilization only	Inside	Syntax-based	Both	No tendency	Not detailed	None
Product line	Implementation	Inside	Syntax-based	Both	No tendency	Not detailed	Relationship
Requirements analysis	Reutilization only	Inside	Syntax-based	Both	Internal actions sequence	No tendency	Relationship
Service responsibility analysis	Reutilization only	Outside	Syntax-based	Direct	Internal events, pre- and-post conditions	Information structure description	Relationship
SOAD	Reutilization only	Inside	Syntax-based	Both	No tendency	No tendency	Relationship
Source code analysis	Implementation	Inside	Syntax-based	Indirect	Internal actions sequence	Information structure description	Resources
Wrapping	Implementation	Inside	Syntax-based	Indirect	Internal actions sequence	Information, messages and parameters structures	Both (resources and services relationships)
Assets identification	Reutilization only	Inside	Syntax-based	Indirect	No tendency	Information structure description	Resources

tion concerns (from the *Participant Concerns* perspective), thus contributing to speed up the construction process. In this same example, according to Table 6, the identified services will be indirectly related to businesses (*Service Value* perspective). Therefore, such services tend to deliver value to IT department rather than to the core enterprise business. As another example, if the practitioner is involved in a SOA project to integrate inter-organizational processes, Table 6 tells him/her that the *Value Analysis* or *Service Responsibility Analysis* techniques must be used because they are the only ones capable of eliciting cross-enterprise interactions (*Context of Transactions* perspective). These examples show how this type of correlation can be a useful tool to shed light on the consequences of practitioner decisions of using different software engineering techniques in the service identification phase of SOA projects.

Moreover, Table 6 can suggest which techniques are the most complete ones for service identification purposes. Service identification must support SOA goals of increasing organizational agility, increasing the return of investment, and promoting the alignment of business and IT domains [16]. In order to promote business and IT alignment, a software engineering technique must be able to *identify services that deliver direct value to business and also IT services that support business services* (C1). The business agility goal needs a clear understanding of the service candidate capability in order to provide services that deliver value to businesses and that are also reusable. Service capability comprehension can be enhanced by: *eliciting the service actions and behavior* (behavior model) (C2), *eliciting service information model* (messages, parameters, and semantic) (C3), *using semantic descriptions* (C4), and *eliciting services interactions with applications or people* (C5). The return of investment can be increased by *analyzing implementation issues* (C6) during the identification in order to avoid the identification of candidates that are too complex to be implemented, and also by *identifying services and resources that can be reused to realize a service capability* (C7). The analysis of implementation issues in the identification step and the clear understanding of the service candidate capability also promote the delivery of a better input for the specification and realization steps, thus enhancing the overall quality of the service-oriented modeling and architectural design process.

Table 7 Service identification characteristics addressed by software engineering techniques

Technique/characteristic	C1	C2	C3	C4	C5	C6	C7
Decomposition	×	×					×
Model-driven approach	×	×	×				
Value analysis					×		
Ontology mapping	×	×	×	×	×		×
Pattern matching	×						
Product line	×					×	
Requirement analysis	×	×					
Service responsibility analysis		×			×		
SOAD	×						
Source code analysis		×				×	
Wrapping		×	×			×	×
Assets identification							

The analysis of the aforementioned characteristics in light of the Table 6 reveals that *Ontology mapping* would be the most complete software engineering technique for service identification. This technique addresses six of the aforementioned characteristics: (i) identification of services that deliver both direct and indirect value to businesses (C1); (ii) elicitation of the internal events, pre- and-post conditions (C2); elicitation of the information model (C3); utilization of semantic-based service description (C4); elicitation of service interactions with applications or people (C5); and identification of services and resources dependencies (C7). Table 7 summarizes the characteristics addressed by the software engineering techniques analyzed in Sect. 4.2.2.

4.2.3 Existing gaps in the SIM field

Besides analyzing existing SIM in the literature in the context of the reference architecture perspectives, the survey findings allowed us to identify gaps in current work that could be further exploited in future researches. The most relevant gaps identified in the surveyed SIMs are (i) *identification of both business and software services*; (ii) *analysis of non-functional requirements*; and (iii) *quality assessment of candidate services*. In the following, we briefly discuss each one in the context of our classification scheme and/or of the analyzed techniques adopted by SIMs.

Identification of both business and software services. Service identification should be wide-ranged and consider multiple perspectives comprising business and technological issues [120]. Business perspective is related to business goals and requirements, and it is generally structured into processes or business models that express rules, constraints, and dependencies. IT perspective is the automation of the business perspective organized into several technology solutions. The analysis of the business perspective is important to identify services that deliver direct value to business and promote business agility, which is one of the SOA major goals [16]. On the other hand, the analysis of the IT perspective promotes an alignment with the existing IT assets and helps to identify resources (data, application functions, and existing services) needed to realize service capabilities, thus producing a better input to the specification phase. These two perspectives are complementary, so that the analysis of only one perspective can compromise the achievement of SOA goals or lead to services that are not suitable to the organizations reality. For example, the isolated analysis of the business perspective can lead to the identification of services that fulfill business requirements, but are very expensive to be developed and integrated with the existing software assets or IT architecture. Another example is the identification of services by functional analysis of existing software assets, which may lead to services with a limited range of reuse, thus compromising the SOA goals of promoting business agility and increasing the return of investment. This gap is directly related to the *Service Value to the business perspective* analyzed in Sect. 4.2.1. As reported in Table 7, few SIMs address both perspectives, and the majority of the SIMs addresses only the business perspective.

Analysis of non-functional requirements. Business requirements are not the only requirements that originate service candidates or affect the capabilities of service candidates. Non-functional, technical requirements might also reveal constraints, conditions of use of a service, or even additional service candidates that support the accomplishment of non-functional requirements. For instance, non-functional requirements regarding security can originate services to authenticate users, control access to specific functionalities, or limit access to some services depending on the user's profile. Moreover, conflicting non-functional requirements might cause service redesign [12], thus impacting all process of service-oriented modeling and design. Despite the relevance of non-functional requirements, few SIMs consider this aspect. It is noteworthy that the software engineering technique employed by the SIM directly affects the capability of eliciting non-functional requirements. Some SIMs elicit non-functional requirements by using SODA technique [52,55,78], which identifies services based on the decomposition of interactions, concerns and features into aspects, and compose them according to requirements. Dinh and Nguyen-Ngoc [19] elicit constraints and legal issues of the information exchange between organizations. Samavi et al. [41] identify non-functional requirements as services softgoals. Finally, the researches in [20,22,25,34,36,38,45,47,49,51,54,65,85,87] have an activity to elicit non-functional requirements or receive the requirements as inputs. Regarding the classification scheme adopted in our work, the gap concerning non-functional requirements is a crosscutting aspect affecting several of the considered perspectives, such as the *Participation Concern*, *Service Description*, *Behavior Model*, *Quality of Attribute*, and *Service Dependency*.

Quality assessment of candidate services. Most researchers agree on the importance of metrics to improve the quality of the identified services and of the SIM itself. Quality is dependent of stakeholder requirements, but some general service quality attributes can be identified in a SOA context. Erl [16] emphasizes that the basic software quality design principles of low coupling and high cohesion should be observed during all service lifecycle. Service granularity is also pointed out as a quality attribute as the granularity level of a service can affect its capabilities, performance, reusability, and coupling. SIMs deal with service candidate quality by using metrics of coupling [22,45,100,109], cohesion [22,45,100,101,109], granularity [22,88], modularity [26,33], reusability using the semantic distance between features [25], and quality of service by estimating a weight to execute activities [23]. SIMs usually do not assess service quality neither do any effort to improve quality attributes of identified candidates. Regardless of the adopted quality attribute, SIMs should provide means to assess service candidate quality. Services with low quality can affect the reuse, thus compromising the achievement of the SOA goals of promoting business agility and improving the return of investment.

5 Threats to validity

The main threats to the validity of this survey are related to:

- *Its completeness*. The electronic databases used in this systematic survey (see Table 2) are considered the most relevant available sources [3,4], but some studies may have been missed due to technical limitations of the search engines themselves, which are out of the control of the researchers. Furthermore, these databases do not represent an exhaustive list of publication sources, so that other databases might also be included.
- *Reviewers' reliability.* Although the conclusions might have been influenced by the researchers' opinions, it was adopted a dual-revision strategy in order to minimize the effect of any bias or misinterpretation. Therefore, the studies were evaluated more than once, each time by a different researcher.
- *Data extraction.* Since not all information was obvious to answer the established research questions, some data had to be interpreted. Nevertheless, discussions were conducted whenever a disagreement between the researchers occurred in order to ensure the validity of this systematic survey.

6 Conclusion

This paper reported the results of a systematic survey that quantitatively characterizes SIMs reported in 105 studies published from 2002 to June 2013 retrieved from four electronic libraries. A classification scheme based on a reference architecture adopted by industry was proposed as a way to suggest which issues should be considered during services identification phase and how the existing approaches address them. SIMs differ in the way they address the proposed classification perspectives. Nevertheless, the analysis presented in Sect. 4.2.1 demonstrates that existing SIMs address many perspectives of the adopted SOA reference architecture, thus suggesting that these SIMs are aligned with the concerns related to SOA adoption in the context of service identification phase.

More than a half of the proposed methods use more than one software engineering technique, but few are meet in the middle approaches. Meet in the middle approaches are more complete as they evaluate models from the highest level to the most detailed one, thus allowing the reuse of existing assets (services and applications) as well as the generation of fine-grained (more reusable) and coarse-grained services that generate immediate value to the businesses.

The technique chosen by each SIM can influence on how each classification perspective is addressed. Perspectives as *Participant Concerns*, *Context of Transactions*, *Service Description*, *Behavior Model*, *Information Model*, and *Service Dependency* have a correlation with the technique. *Service Value* perspective is influenced by the input used by the technique, but not by the technique itself, and perspectives as *Service Granularity*, *Type of Conversation*, and *Quality Attributes elicitation* do not seem to have correlation with the employed techniques. Furthermore, we intend to aid practitioners to understand the consequences of using software engineering techniques employed by different SIMs and also to encourage researchers to promote improvements in this field by combining techniques or creating new ways to address the service identification perspectives.

Acknowledgments This work was partially supported by Brazilian funding agencies FAPERJ, CNPq and CAPES.

References

1. Kitchenham BA (2004) Procedures for performing systematic reviews. Technical report, Keele University/National ICT Australia Ltd
2. Kitchenham BA, Dybå T, Jørgensen M (2004) Evidence-based software engineering. In: Proceedings of the 26th international conference on software engineering (ICSE 2006). IEEE Computer Society, Washington, pp 273–281
3. Kitchenham BA, Charters S (2007) Guidelines for performing systematic literature reviews in software engineering. Technical report, Keele University/University of Durham
4. Dybå T, Dingsøyr T, Hansen GK (2007) Applying systematic reviews to diverse study types: an experience report. In: Proceedings of the first international symposium on empirical software engineering and measurement (ESEM 2007). IEEE Computer Society, Washington, pp 225–234
5. Arsanjani A (2004) Service-oriented modeling and architecture: how to identify, specify, and realize services for your SOA. http://www.ibm.com/developerworks/webservices/library/ws-soa-design1/
6. Kitchenham B, Brereton OP, Budgen D, Turner M, Bailey J, Linkman S (2009) Systematic literature reviews in software engineering: a systematic literature review. Inf Softw Technol 51(1):7–15
7. Dieste O, Grimán A, Juristo N (2009) Developing search strategies for detecting relevant experiments. Empir Softw Eng 14(5):513–539
8. Demirkan H, Kauffman RJ, Vayghan JA, Fill HG, Karagiannis D, Maglio DD (2008) Service-oriented technology and management: perspectives on research and practice for the coming decade. Electron Commer Res Appl 7(4):356–376
9. Boerner R, Goeken M (2009) Service identification in SOA governance literature review and implications for a new method. In: Proceedings of the 3rd IEEE international conference on digital ecosystems and technologies (DEST'09). IEEE Computer Society, USA, pp 588–593
10. Birkmeier D, Klöckner S, Overhage S (2009) A survey of service identification approaches: classification framework, state of the art, and comparison. Enterp Model Inf Syst Archit 4(2):20–36
11. Kohlborn T, Korthaus A, Chan T, Rosemann M (2009) Service analysis—a critical assessment of the state of the art. In: Proceedings of the 17th European conference on information systems (ECIS 2009), pp 1583–1594
12. Gu Q, Lago P (2010) Service identification methods: a systematic literature review. In: Di Nitto E, Yahyapour R (eds) Proceedings of the 3rd European conference towards a service-based internet (servicewave 2010). Lecture notes in computer science, vol 6481, pp 37–50. Springer, Berlin
13. Cai S, Liu Y, Wang X (2011) A survey of service identification strategies. In: Proceedings of the 2011 IEEE Asia-Pacific services computing conference (APSCC 2011). IEEE Computer Society, Piscataway, pp 464–470
14. OASIS (2012) Reference architecture foundation for service oriented architecture, version 1.0. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-v1.0
15. Teka AY, Condori-Fernandez N, Sapkota B (2012) A systematic literature review on service description methods. In: Regnell B, Damian D (eds) Proceedings of the 18th international working conference on requirements engineering: foundation for software quality (REFSQ'12). Lecture notes in computer science, vol 7195, pp 239–255. Springer, Berlin
16. Erl T (2007) SOA principles of service design. Prentice Hall, Englewood Cliffs
17. Ricca F, Marchetto A (2009) A "quick and dirty" meet-in-the-middle approach for migrating to SOA. In: Proceedings of the 2009 joint international and annual ERCIM workshops on principles of software evolution (IWPSE) and software evolution (evol) workshops. ACM, New York, pp 73–78
18. Shirazi HM, Fareghzadeh N, Seyyedi A (2009) A combinational approach to service identification in SOA. J Appl Sci Res 5(10):1390–1397
19. Dinh TL, Nguyen-Ngoc AV (2010) A conceptual framework for designing service-oriented inter-organizational information systems. In: Proceedings of the 2010 symposium on information and communication technology (SoICT'10). ACM, New York, pp 147–154
20. Lee J, Muthig D, Naab M (2010) A feature-oriented approach for developing reusable product line assets of service-based systems. J Syst Softw 83(7):1123–1136
21. Jin Z, Zhu H (2008) A framework for agent-based service-oriented modeling. In: Proceedings of the 4th IEEE international symposium on service-oriented system engineering (SoSE 2008). IEEE, USA, pp 160–165
22. Kazemi A, Rostampour A, Jamshidi P, Nazemi E, Shams F, Azizkandi AN (2011) A genetic algorithm based approach to service identification. In: Proceedings of the 2011 IEEE world congress on services (SERVICES 2011). IEEE Computer Society, Piscataway, pp 339–346
23. Menascé DA, Casalicchio E, Dubey V (2008) A heuristic approach to optimal service selection in service oriented architectures. In: Proceedings of the 7th international workshop on software and performance (WOSP'08). ACM, New York, pp 13–24
24. Azevedo LG, Santoro F, Baião F, Souza J, Revoredo K, Pereira V, Herlain I (2009) A method for service identification from business process models in a SOA approach. In: Halpin T, Krogstie J, Nurcan S, Proper E, Schimidt R, Soffer P, Ukor R (eds) Proceedings of the 10th international workshop on business process modeling, development and support (BPMDS 2009) and 14th international conference on exploring modeling methods in systems analysis and design (EMMSAD 2009). Lecture notes in business information processing, vol 29, pp 99–112. Springer, Berlin
25. Kang D, Song CY, Baik DK (2008) A method of service identification for product line. In: Proceedings of the third international conference on convergence and hybrid information technology (ICCIT'08), vol 2. IEEE Computer Society, Piscataway
26. Kazemi A, Rostampour A, Azizkandi AN, Haghighi H, Shams F (2011) A metric suite for measuring service modularity. In: Proceedings of the 2011 CSI international symposium on computer science and software engineering (CSSE 2011). IEEE, USA, pp 95–102
27. Dwivedi V, Kulkarni N (2008) A model driven service identification approach for process centric systems. In: Proceedings of the 2008 IEEE congress on services–part II (SERVICES-2 2008). IEEE Computer Society, Washington, pp 65–72
28. Alahmari S, de Roure D, Zaluska E (2010) A model-driven architecture approach to the efficient identification of services on service-oriented enterprise architecture. In: Proceedings of the 14th IEEE international enterprise distributed object computing conference workshops (EDOCW'10). IEEE Computer Society, Washington, pp 165–172

29. Zhang Z, Zhou DD, Yang HJ, Zhong SC (2010) A service composition approach based on sequence mining for migrating e-learning legacy system to SOA. Int J Autom Comput 7(4):584–595
30. Yun Z, Huayou S, Yulin N, Hengnian Q (2009) A service-oriented analysis and design approach based on data flow diagram. In: Proceedings of the 2009 international conference on computational intelligence and software engineering (CiSE 2009). IEEE, Piscataway
31. Huayou S, Yulin N, Lian Y, Zhong C (2009) A service-oriented analysis and modeling using use case approach. In: Proceedings of the 2009 international conference on computational intelligence and software engineering (CiSE 2009). IEEE, Piscataway
32. Abdelaziz EF, Mustapha A, Mohammed S (2011) A service-oriented approach for information systems development. In: Proceedings of the 2011 international conference on multimedia computing and systems (ICMCS 2011). IEEE, USA, pp 1–6
33. Li S, Tahvildari T (2006) A service-oriented componentization framework for Java software systems. In: Proceedings of the 13th working conference on reverse engineering (WCRE 2006). IEEE Computer Society, Piscataway, pp 115–124
34. Schmidt R (2011) A service-system based identification of meta-services for service-oriented enterprise architecture. In: Proceedings of the 15th IEEE international enterprise distributed object computing conference workshops (EDOCW'11). IEEE Computer Society, Washington, pp 293–300
35. Jain H, Zhao H, Chinta NR (2004) A spanning tree based approach to identifying web services. Int J Web Serv Res 1:1–20
36. Chang SH (2007) A systematic analysis and design approach to develop adaptable services in service oriented computing. In: Proceedings of the 2007 IEEE congress on services (SERVICES 2007). IEEE Computer Society, Piscataway, pp 375–378
37. Canfora G, Fasolino AR, Frattolillo G, Tramontana P (2008) A wrapping approach for migrating legacy system interactive functionalities to service oriented architectures. J Syst Softw 81(4):463–480
38. Andersson B, Johannesson P, Zdravkovic J (2009) Aligning goals and services through goal and business modeling. Inf Syst e-Bus Manag 7(2):143–169
39. Lee J, Sugumaran V, Park S, Sansi D (2011) An approach for service identification using value co-creation and IT convergence. In: Proceedings of the first ACIS/JNU international conference on computers, networks, systems and industrial engineering (CNSI 2011). IEEE Computer Society, Piscataway, pp 441–446
40. Park J, Moon M, Yun S, Yeom K (2009) An approach to enhancing reusabilities in service development. In: Proceedings of the 2009 international conference on hybrid information technology (ICHIT'09). ACM, New York, pp 143–150
41. Samavi R, Yu E, Topaloglou T (2008) Applying strategic business modeling to understand disruptive innovation. In: Proceedings of the 10th international conference on electronic commerce (ICEC'08). ACM, New York
42. Gacitua-Decar V, Pahl C. Automatic business process pattern matching for enterprise services design. In: Proceedings of the 2009 world conference on services–part II (SERVICES-2 2009), pp 111–118. IEEE Computer Society, Washington
43. Yousef R, Odeh M, Coward D, Sharieh A (2009) BPAOntoSOA: a generic framework to derive software service oriented models from business process architectures. In: Proceedings of the second international conference on the applications of digital information and web technologies (ICADIWT'09). IEEE, USA, pp 50–55
44. Ponnalagu K, Narendra NC (2008) Deriving service variants from business process specifications. In: Proceedings of the 1st Bangalore annual compute conference (COMPUTE'08). ACM, New York
45. Medeiros FM, Almeida ES, Meira SRL (2010) Designing a set of service-oriented systems as a software product line. In: Proceedings of the 4th Brazilian symposium on software components, architectures and reuse (SBCARS 2010). IEEE Computer Society, Washington, pp 70–79
46. Ruiz M, Pelechano V, Pastor O (2006) Designing Web services for supporting user tasks: A model driven approach. In: Roddick JF et al (eds) Proceedings of the workshops on the 25th international conference on conceptual modeling (ER 2006). Lecture notes in computer science, vol 4231, pp 193–202. Springer, Berlin
47. Asadi M, Mohabbati B, Gašević D, Bagheri E (2011) Developing families of method-oriented architecture. In: Ralyté J, Mirbel I, Deneckère R (eds) Proceedings of the 4th IFIP WG 8.1 working conference on method engineering (ME 2011). IFIP advances in information and communication technology, vol 351, pp 168–183. International federation for information processing (IFIP)/Springer, Berlin
48. Li S, Tahvildari L (2008) E-BUS: a toolkit for extracting business services from java software systems. In: Companion of the 30th international conference on software engineering. ACM, New York, pp 961–962
49. Kaabi RS, Souveyet C, Rolland C (2004) Eliciting service composition in a goal driven manner. In: Proceedings of the 2nd international conference on service oriented computing (ICSOC'04). ACM, New York, pp 308–315
50. Gordijn J, Yu E, van der Raadt B (2006) e-Service design using i* and e3 value modeling. IEEE Softw 23(3):26–33
51. Ilayperuma T, Zdravkovic J (2010) Exploring business value models from the inter-organizational collaboration perspective. In: Proceedings of the 2010 ACM symposium on applied computing (SAC'10). ACM, New York, pp 99–105
52. Bao L, Yin C, He W, Ge J, Chen P (2010) Extracting reusable services from legacy object-oriented systems. In: Proceedings of the 2010 IEEE international conference on software maintenance (ICSM 2010). IEEE, USA, pp 1–5
53. Chen F, Li S, Yang H, Wang CH, Chu WCC (2005) Feature analysis for service-oriented reengineering. In: Proceedings of the 12th Asia-Pacific software engineering conference (APSEC'05). IEEE Computer Society, Piscataway
54. Kim Y, Doh KG (2009) Formal identification of right-grained services for service-oriented modeling. In: Vossen G, Long DED, Yu JX (eds) Proceedings of the 10th international conference on web information systems engineering (WISE 2009). Lecture notes in computer science, vol 5802, pp 261–273. Springer, Berlin
55. Mosser B, Mussbacher G, Blay-Fornarino M, Amyot D (2011) From aspect-oriented requirements models to aspect-oriented business process design models: an iterative and concern-driven approach for software engineering. In: Proceedings of the 10th international conference on aspect-oriented software development (AOSD'11). ACM, New York, pp 31–42
56. Lo A, Yu E (2007) From business models to service-oriented design: a reference catalog approach. In: Parent C, Schewe KD, Storey VC, Thalheim B (eds) Proceedings of the 26th international conference on conceptual modeling (ER 2007). Lecture notes in computer science, vol 4801, pp 87–101. Springer, Berlin
57. Nguyen DK, van den Heuvel WJ, Papazoglou MP, de Castro V, Marcos E (2009) Gap analysis methodology for business service engineering. In: Proceedings of the 2009 IEEE conference on commerce and enterprise computing (CEC 2009). IEEE Computer Society, Washington, pp 215–220
58. Meertens LO, Iacob ME, Nieuwenhuis LJM (2010) Goal and model driven design of an architecture for a care service platform. In: Proceedings of the 2010 ACM symposium on applied computing (SAC'10). ACM, New York, pp 158–164
59. Aier S (2005) How clustering enterprise architectures helps to design service oriented architectures. In: Proceedings of the 2006

IEEE international conference on services computing (SCC'06). IEEE Computer Society, Washington, pp 269–272

60. Klose K, Knackstedt R, Beverungen D (2007) Identification of services: a stakeholder-based approach to SOA development and its application in the area of production planning. In: Proceedings of the 15th European conference on information systems (ECIS 2007)
61. caetano A, Silva AR, Tribolet J (2010) Identification of services through functional decomposition of business processes. In: Abramowicz W, Tolksdorf R (eds) Proceedings of the 13th international conference on business information systems (BIS 2010). Lecture notes in business information processing, vol 47, pp 144–157. Springer, Berlin
62. Zhang Z, Yang H (2004) Incubating services in legacy systems for architectural migration. In: Proceedings of the 11th Asia-Pacific software engineering conference (ASPEC 2004), pp 196–203
63. Alizadeh K, Seyyedi MA, Mohsenzadeh M (2011) Mapping service concept and enterprise ontology in service identification. In: Proceedings of the 7th international conference on networked computing (INC 2011), pp 22–27
64. Aversano L, Cerulo L, Palumbo C (2008) Mining candidate web services from legacy code. In: Proceedings of the 10th international symposium on web site evolution (WSE 2008). IEEE, Piscataway, pp 37–40
65. Asadi M, Mohabbati B, Kaviani N, Gašević D, Bošković M, Hatala M (2009) Model-driven development of families of service-oriented architectures. In: Proceedings of the first international workshop on feature-oriented software development (FOSD'09). ACM, New York, pp 95–102
66. Kenzi A, El Asri B, Nassar M, Kriouile A (2008) Multi-functional service oriented system development for user-driven adaptability. In: Proceedings of the 3rd international conference on information and communication technologies: from theory to applications (ICTTA 2008), pp 1–7
67. Wang Z, Xu X, Zhan D (2005) Normal forms and normalized design method for business service. In: Proceedings of the 2005 IEEE international conference on e-business engineering (ICEBE 2005). IEEE Computer Society, Piscataway, pp 79–86
68. de Bruin J, Kok JN, Lavrac N, Trajkovski I (2009) On the design of knowledge discovery services design patterns and their application in a use case implementation. In: Margaria T, Steffen B (eds) Proceedings of the third international symposium on leveraging applications of formal methods, verification and validation (ISoLA 2008). Communications in computer and information science, vol 17, pp 649–662. Springer, Berlin
69. Kim G, Suhh Y (2010) Ontology-based semantic matching for business process management. ACM SIGMIS Database 41(4):98–118
70. Wen B, He K, Liang P, Xu L (2010) Process semantic-enabled customisation for active service provisioning. In: Proceedings of the 13th IEEE international conference on computational science and engineering (CSE'10). IEEE Computer Society, Washington, pp 375–381
71. Kannan K, Srivastava B (2008) Promoting reuse via extraction of domain concepts and service abstractions from design diagrams. In: Proceedings of the 2008 IEEE international conference on services computing (SCC'08), vol 1. IEEE Computer Society, Washington, pp 265–272
72. Baghdadi Y (2006) Reverse engineering relational databases to identify and specify basic web services with respect to service oriented computing. Inf Syst Front 8(5):395–410
73. Patig S, Wesenberg H (2009) Role of process modeling in software service design. In: Baresi L, Chi CH, Suzuki J (eds) Service-oriented computing, Proceedings of the 7th international joint conference on service-oriented computing (ICSOC-ServiceWave 2009). Lecture notes in computer science, vol 5900, pp 420–428. Springer, Germany
74. Ren M, Wang Y (2010) Rule based business service identification using UML analysis. In: Proceedings of the 2nd IEEE international conference on information management and engineering (ICIME 2010), pp 199–204
75. Brzostowski K, Rekuć W, Sobecki J, Szczurowski L (2010) Service discovery in the SOA system. In: Nguyen NT, Le MT, Świątek J (eds) Proceedings of the second international conference on intelligent information and database systems (ACIIDS'10)–part II. Lecture notes in computer science, vol 5991, pp 29–38. Springer, Berlin
76. Cho M, Je C, Rim KH, Hong SG, Keceli Y, Park JY (2008) Service identification and modeling for service oriented architecture applications. In: Proceedings of the 7th WSEAS international conference on software engineering, parallel and distributed systems (SEAPADS'08), pp 193–199. World Scientific and Engineering Academy and Society (WSEAS), Stevens Point
77. Zhang Z, Liu R, Yang H (2005) Service identification and packaging in service oriented reengineering. In: Proceedings of the 7th international conference on software engineering and knowledge engineering (SEKE 2005), pp 241–249
78. Perin-Souza A, Cappelli C, Santoro FM, Azevedo LG, Leite JCSP, Batista TV (2011) Service identification in aspect-oriented business process models. In: Proceedings of the 6th international symposium on service oriented software engineering (SOSE 2011). IEEE, Piscataway, pp 164–174
79. Shiang WJ, Rau H, Lin YH (2009) Service identification of a collaborative design workflow in a dynamically changing environment. In: Proceedings of the 2009 international conference on networking, sensing and control (ICNSC'09), pp 685–690. IEEE Systems, Man, and Cybernetics Society, USA
80. Stewart G, Chakraborty A (2010) Service identification through value chain analysis and prioritization. In: Proceedings of the 16th Americas conference on information systems: sustainable IT collaboration around the globe
81. Kim S, Kim M, Park S (2008) Service identification using goal and scenario in service oriented architecture. In: Proceedings of the 15th Asia-Pacific software engineering conference (ASPEC 2008). IEEE Computer Society, Washington, pp 419–426
82. Chen F, Zhang Z, Li J, Kang J, Yang H (2009) Service identification via ontology mapping. In: Proceedings of the 33rd annual IEEE international computer software and applications conference (COMPSAC'09), vol 1. IEEE Computer Society, Washington, pp 486–491
83. Inaganti S, Behara GK (2007) Service identification: BPM and SOA handshake. http://w.bptrends.com/publicationfiles/THREE%2003-07-ART-BPMandSOAHandshake-InagantiBehara-Final
84. Chung S, Won D, Baeg SH, Park S (2009) Service-oriented reverse reengineering: 5W1H model-driven re-documentation and candidate services identification. In: Proceedings of the 2009 IEEE international conference on service-oriented computing and applications (SOCA 2009), pp 1–6. IEEE Computer Society, Piscataway
85. Erradi A, Anand S, Kulkarni N (2006) SOAF: an architectural framework for service definition and realization. In: Proceedings of the 2006 IEEE international conference on services computing (SCC'06). IEEE Computer Society, Piscataway, pp 151–158
86. Arsanjani A, Ghosh S, Allam A, Abdollah T, Gariapathy S, Holley K (2008) SOMA: a method for developing service-oriented solutions. IBM Syst J 47(3):377–396
87. Weigand H (2011) The pragmatics of event-driven business processes. In: Proceedings of the 7th international conference on semantic systems (I-Semantics'11). ACM, New York, pp 211–218

88. Kim Y, Doh KG (2007) The service modeling process based on use case refactoring. In: Abramowicz W (ed) Proceedings of the 10th international conference on business information systems (BIS 2007). Lecture notes in computer science, vol 4439, pp 108–120. Springer, Berlin
89. Jamshidi P, Sharifi M, Mansour S (2008) To establish enterprise service model from enterprise business model. In: Proceedings of the 2008 IEEE international conference on services computing (SCC'08), vol 1. IEEE Computer Society, Piscataway, pp 93–100
90. Jamshidi P, Khoshnevis S, Teimourzadegan R, Nikravesh A, Shams F (2009) Toward automatic transformation of enterprise business model to service model. In: Proceedings of the 2009 ICSE workshop on principles of engineering service oriented systems (PESOS 2009), pp 70–74
91. Chaari S, Biennier F, Favrel J, Benamar C (2007) Towards a service-oriented enterprise based on business components identification. In: Gonçalves RJ, Müller JP, Mertins K, Zelm M (eds) Enterprise interoperability II. Part V. Springer, United Kingdom, pp 495–506
92. Liu W, Fu M, Luo S, Zou D (2011) Use case-based service-oriented analysis and modeling. In: Proceedings of the 2011 international conference on internet computing and information services. IEEE Computer Society, Piscataway, pp 94–96
93. Mani S, Sinha VS, Sukaviriya N, Ramachandra T (2008) Using user interface design to enhance service identification. In: Proceedings of the 2008 IEEE international conference on web services (ICWS'08). IEEE Computer Society, Piscataway, pp 78–87
94. Weigand H, Johannesson P, Andersson B, Bergholtz M (2009) Value-based service modeling and design: toward a unified view of services. In: van Eck P, Gordijn J, Wieringa R (eds) Proceedings of the 21st international conference on advanced information systems engineering (CAiSE 2009). Lecture notes in computer science, vol 5565, pp 410–424. Springer, Berlin
95. Abu-Matar M, Gomaa H (2011) Variability modeling for service oriented product line architectures. In: Proceedings of the 15th international software product line conference (SPLC 2011). IEEE Computer Society, Piscataway, pp 110–119
96. Nakamura M, Igaki H, Kimura T, Matsumoto KI (2011) Identifying services in procedural programs for migrating legacy system to service oriented architecture. Int J Inf Syst Serv Sect 3(4):54–72
97. Komondoor R, Nandivada VK, Sinha S, Field J (2012) Identifying services from legacy batch applications. In: Proceedings of the 5th India software engineering conference (ISEC'12). ACM, New York, pp 13–22
98. Birkmeier DQ, Gehlert A, Overhage S, Schlauderer S (2013) Alignment of business and IT architectures in the German Federal Government: a systematic method to identify services from business processes. In: Proceedings of the 46th Hawaii international conference on systems sciences (HICSS 2013). IEEE Computer Society, Piscataway, pp 3848–3857
99. Guan Q, Feng S, Ma Y (2012) A network topology clustering algorithm for service identification. In: Proceedings of the 2012 international conference on computer science and service system (CSSS'12). IEEE Computer Society, Washington, pp 1583–1586
100. Bianchini D, Cappiello C, De Antonellis V, Pernici B (2013) Service identification in interorganizational process design. IEEE Trans Serv Comput 7(2):265–278
101. Abdelkader M, Malki M, Benslimane SM (2013) A heuristic approach to locate candidate web service in legacy software. Int J Comput Appl Technol 47(2–3):152–161
102. Upadhyaya B, Khomh F, Zou Y (2012) Extracting RESTful services from web applications. In: Proceedings of the 5th IEEE international conference on service-oriented computing and applications (SOCA 2012). IEEE Computer Society, Piscataway, pp 1–4
103. Fuhr A, Horn T, Riediger V, Winter A (2013) Model-driven software migration into service-oriented architectures. Comput Sci Res Dev 28(1):65–84
104. Han F, Moller E, Berre AJ (2009) Organizational interoperability supported through goal alignment with BMM and service collaboration with SoaML. In: Proceedings of the international conference on interoperability for enterprise software and applications China (I-ESA'09). IEEE Computer Society, Washington, pp 268–274
105. Kim Y, Doh KG (2012) Pragmatic granularity decision for right-grained services in service-oriented modelling. Int J Web Grid Serv 8(2):111–133
106. Kim Y, Doh KG (2013) Use-case driven service modelling with XML-based tailoring for SOA. Int J Web Grid Serv 9(1):35–53
107. Kohlmann F, Alt R (2007) Business-driven service modeling: a methodological approach from the finance industry. In: Proceedings of SABRE 2007
108. Fareghzadeh N (2008) Service identification approach to SOA development. In: Proceedings of the world academy of science, engineering and technology, vol 45
109. Bianchini D, Cappiello C, De Antonellis V, Pernici B (2009) P2S: a methodology to enable inter-organizational process design through web services. In: van Eck P, Gordijn J, Wieringa R (eds) Proceedings of the 21st international conference on advanced information systems engineering (CAiSE 2009). Lecture notes in computer science, vol 5565, pp 334–348. Springer, Berlin
110. Erradi A, Kulkarni N, Maheshwari P (2007) Service design process for reusable services: financial services case study. In: Krämer BJ, Lin KJ, Narashimhan P (eds) Proceedings of the 5th international conference on service-oriented computing (ICSOC 2007). Lecture notes in computer science, vol 4749, pp 606–617. Springer, Berlin
111. Flaxer D, Nigam A (2004) Realizing business components, business operations and business services. In: Proceedings of the 2004 IEEE international conference on e-commerce technology for dynamic e-business (CEC-EAST'04). IEEE Computer Society, Washington, pp 328–332
112. Vemulapalli A, Subramanian N (2009) Transforming functional requirements from UML into BPEL to efficiently develop SOA-based systems. In: Meersman R, Herrero P, Dillon T (eds) Workshop proceedings of the 2009 on the move to meaningful internet systems federated conferences and workshops (OTM 2009). Lecture notes in computer science, vol 5872, pp 337–349. Springer, Berlin
113. Sewing JH, Rosemann M, Dumas M (2006) Process-oriented assessment of web services. Int J E-Bus Res 2(1):19–44
114. Adamopoulos DX, Pavlou G, Papandreou CA (2002) Advanced service creation using distributed object technology. IEEE Commun Mag 40(3):146–154
115. Vale T, Figueiredo GB, Almeida ES, Meira SLR (2012) A study on service identification methods for software product lines. In: Proceedings of the 16th international software product line conference (SPLC 2012), vol 2. ACM, New York, pp 156–163
116. Zadeh AT, Mukhtar M, Sahran S, Khabbazi MR (2012) A systematic input selection for service identification in SMEs. J Appl Sci 12(12):1232–1244
117. Jabref Reference Manager. http://jabref.sourceforge.net/
118. Huang C, Lee GM, Crespi N (2012) A semantic enhanced service exposure model for a converged service environment. IEEE Commun Mag 50(3):32–40
119. Berners-Lee T, Hendler J, Lassila O (2001) The semantic web. Sci Am 284(5):34–43
120. Bell M (2010) SOA modeling patterns for service-oriented discovery and analysis. Wiley, Hoboken