Testing in Service Oriented Architectures with Dynamic Binding: A Mapping Study

Marcos Palacios [1*], José García-Fanjul [2], Javier Tuya [2]

Department of Computer Science.
University of Oviedo.
Campus de Viesques, 33204 Gijón. Asturias. Spain.

[1] palacios@lsi.uniovi.es
[2] {jgfanjul | tuya}@uniovi.es

[*]Corresponding author. Tel.: +34 985 182 153. Fax: +34 985 181 986.
E-mail address: palacios@lsi.uniovi.es (M. Palacios).

Abstract

Context: Service Oriented Architectures (SOA) have emerged as a new paradigm to develop interoperable and highly dynamic applications.
Objective: This paper aims to identify the state of the art in the research on testing in Service Oriented Architectures with dynamic binding.
Method: A mapping study has been performed employing both manual and automatic search in journals, conference/workshop proceedings and electronic databases.
Results: A total of 33 studies have been reviewed in order to extract relevant information regarding a previously defined set of research questions. The detection of faults and the decision making based on the information gathered from the tests have been identified as the main objectives of these studies. To achieve these goals, monitoring and test case generation are the most proposed techniques testing both functional and non-functional properties. Furthermore, different stakeholders have been identified as participants in the tests, which are performed in specific points in time during the life cycle of the services. Finally, it has been observed that a relevant group of studies have not validated their approach yet.
Conclusions: Although we have only found 33 studies that address the testing of SOA where the discovery and binding of the services are performed at runtime, this number can be considered significant due to the specific nature of the reviewed topic. The results of this study have contributed to provide a body of knowledge that allows identifying current gaps in improving the quality of the dynamic binding in SOA using testing approaches.

Keywords

Software testing
Service Oriented Architectures
SOA
Dynamic binding
Mapping study
Systematic literature review

1 Introduction

Testing Service Oriented Architectures (SOA) presents new challenges to researchers because some traditional testing techniques need to be suitably adapted due to the unique features of this new paradigm [1]. Canfora and Di Penta [2] identify some key issues that make the task of testing service-oriented systems difficult: namely, lack of observability of the source code, lack of control of the services, the cost of testing when services are not deployed in the tester's infrastructure and the invocations are charged on a *per-use* basis (the cost derived from an exhaustive test suite could be unmanageable) and the dynamic behaviour that allows discovering and binding a service at runtime. Testing this dynamic binding is one of the most challenging tasks in SOA because the final bound services cannot be known until the moment of the invocations. Hence, there have been a number of recent studies that aim to improve the quality of the dynamic binding and to provide clients with more confidence in the service provision using testing approaches.

This article focuses on identifying the state of the art in the research on testing SOA with dynamic binding. The objective of this review is to search, analyze and discuss the different approaches that have been proposed by performing a mapping study [3]. This is a form of systematic literature review (SLR) [4] that aims to identify and categorise the available research on a specific topic [5]. The mapping study has been performed following a protocol that was developed to guide the search, selection and synthesis of the studies that address the testing of SOA with dynamic binding. This protocol includes the formulation of a set of research questions, the establishment of the search process, the decision about inclusion/exclusion criteria and, finally, the development of a quality assessment study and data extraction guidelines.

The main goals of the analysed studies have been classified and the system under test has been identified. We have also categorised the set of stakeholders that take part in the tests and the points in time when this process is performed. The application of both new testing techniques and modified traditional ones has been discussed. Finally, the current trends in standards and technologies that are being used and the types of validation of the approaches have also been analysed.

The remainder of this article is organized as follows. A brief overview of Service Oriented Architectures and software testing is first presented in Section 2. In Section 3 the details of the review protocol are described. Section 4 reports the results of the mapping study. In Section 5 a discussion about the results and related work is presented. Finally, the conclusions and recommendations for further research are described in the last section.

2 Background

To provide a context for this review, in this section a brief description about the main principles of Service Oriented Architectures and some concepts about software testing are explained.

2.1 Service Oriented Architectures

Service Oriented Architectures have become an emerging paradigm to develop distributed applications by integrating available services over the web. Such services are autonomous and platform-independent entities that can be described, published, discovered and dynamically assembled for developing rapid, low-cost, interoperable and evolvable distributed applications [6]. Web services are the most used SOA based

technology and they are supported with a set of W3C XML based standards: Simple Object Access Protocol (SOAP) [7], Web Service Description Language (WSDL) [8] and Universal Description, Discovery and Integration (UDDI) [9].
In Figure 1 the typical SOA triangle, adapted from [10], with the roles that each stakeholder plays and the operations that can be performed is depicted.

----------Figure 1----------

Service Oriented Architectures allow the interaction between service providers and clients. In this architecture, the provider publishes the description of the services, generally specified in WSDL, in a registry. This registry is often implemented using the UDDI standard and it is in charge of storing service descriptions and acts as an intermediary between providers and clients. After the services are published, a client sends a query to the registry to find the desired service. The registry matches the client's request with the available information and returns to the client a set of service interface descriptions that satisfy its requirements. The client has to select the most suitable service and bind with its provider performing the invocation of the service and receiving the corresponding response.
A client can decide at design time which service is going to be executed so the binding is considered as static. However, a challenging feature of SOA is the possibility to select and invoke a service at runtime. There are two scenarios where this binding can be considered as dynamic. In the first scenario, a set of potential candidate services is available at design time although the client does not know exactly which one is going to be invoked until the moment of the binding. In the second, the discovery, selection and invocation can be performed at runtime using a registry. In this case, until the discovery, the client has no knowledge about the potential services that can be invoked. Hence, all the studies that have been reviewed in this work aim to improve the quality of the binding in these two scenarios.

2.2 Software testing

The concept of software testing has been used with different meanings in the literature. According to the ISO/IEC 24765 (Software and Systems Engineering Vocabulary) [11], testing is "an activity in which a system is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system". In this paper, said definition has been used and, thus, testing means executing the software in order to observe these executions and give a verdict on them. For example, such information can allow detecting faults in the system under test or evaluating the Quality of Service (QoS) of different implementations in order to select the most suitable. Hence, applying this definition to the SOA scope, we have also considered those studies that execute services in order to gather metrics about the QoS.
A currently emerging approach in software testing is the passive observation of real time executions with the aim of detecting any deviation from the expected behaviour of the system during its operation [12]. This concept has been defined as on-line testing by Bertolino in [13] and it is also known as monitoring. We have included in this review such works that monitor services executions or receive information about clients' invocations in order to detect faults or make a decision on the basis of this information.
The studies included in this review test both functional and non-functional properties of the system. Functional testing involves evaluating the compliance of a system with the logic of the specification while testing non-functional properties aims to check the

behaviour of the system with respect to some observable attributes, such as reliability or efficiency.
There are different stakeholders that may play an active role in the testing process. Because each role has different requirements and has different aptitudes for performing specific testing tasks [2], it is usual to find several different roles taking part in the testing process. For example, one stakeholder may be in charge of generating and storing the test cases whereas another stakeholder may be in charge of executing the tests.

3 Review method

The review protocol of the mapping study has been developed following the guidelines for performing SLRs provided in [14] and later updated in [4] by Kitchenham. These guidelines have been used because the method for searching for the primary studies, taking a decision about the inclusion or exclusion of a study and performing the quality studio assessment is very similar between an SLR and a mapping study. The main difference between them is the formulation of the research questions and the analysis of the available information [3]. Among the most common factors to undertake a systematic review, our aim is to identify any gap in current research in order to suggest areas for further investigation and to provide a background to appropriately position new research activities.

3.1 Research questions

All the activities of this study are driven by the research questions as they will identify the scope of the selected publications [15]. The research questions (RQ) for this review are listed hereafter and all of them are related to the scope of testing SOA with dynamic binding.

RQ1. What is the main objective of the research?
RQ2. Which testing techniques and methods are used?
RQ3. Which of the different stakeholders take part in the testing processes?
RQ4. When are the tests performed?
RQ5. Which are the most common technologies and standards being researched?
RQ6. What method is being used to validate the research?

These are the main questions to be answered after having undertaken the mapping study, but other questions which do not concern to the research topic may be relevant. We have also addressed two additional questions concerning the number of studies and where they were published:

RQ7. How much activity about dynamic binding SOA testing research has there been during recent years?
RQ8. Where have the researches been published?

Some of the aforementioned research questions have a direct relation with the different dimensions common to software testing proposed in [13]. In that paper, Bertolino specifies different views or aspects that are related to the execution and observation of the tests using questions. Our RQ1 refers to the question WHY because the objectives

of the different approaches are identified. RQ2 is related to the question HOW because we identify the testing techniques that are employed for testing purposes. RQ3 is related to the question WHERE because the identification of the stakeholder that participates in the tests allows distinguishing where the tests are performed. The point in time when this process is carried out is identified in RQ4, with regard to the question WHEN. RQ5 is related to WHAT is being researched. Finally, RQ7 identifies HOW MUCH effort is being dedicated to test SOA with dynamic binding.

3.2 Search process

The search process is the set of tasks that have to be performed in order to decide what literature sources will be searched and how this search is going to be carried out. This process includes the selection of the search terms and the establishment of the search strategy.

Selection of the search terms

First of all, the most suitable words, synonyms, acronyms or alternative spelling within the research field have been identified according to the three viewpoints recommended by Kitchenham [4] (population, intervention and outcomes). In this mapping study we have considered the scope of SOA and, specifically, those applications where the discovery, selection and binding of the services are performed at runtime. Thus, the population terms have been selected from two complementary points of view. The first criterion involves terms that are related to the technologies and standards which are being researched in the context of SOA while the second criterion includes the specific terms that are related to the dynamic binding in web service compositions. Hence, the population terms are a sum of these two criteria:

Population = **Pop. Criterion1** AND **Pop. Criterion2**

The population terms are the following:

Pop. Criterion1: web services, service oriented, composition, composite web services, compose, SOA, SOAP, WSDL, UDDI, SLA, WSLA, WS-Agreement, OWL-S, BPEL, WS-BPEL, BPEL4WS, BPMN.

Pop. Criterion2: discovery, selection, binding, dynamic, linking, self-healing, self-adapting, adaptive, adaptation, interoperability, compatibility, capability, broker, matching, matchmaking, runtime.

With respect to the intervention point of view, such terms that are mostly used in testing aware studies have been selected. These are the intervention terms:

Intervention: testing, test cases, monitoring, monitor, checking, validation, verification, quality.

In this scope, the outcomes should be the features that are going to be tested, i.e. functional properties, performance, availability, etc. However, no constraint has been

set concerning the outcomes of each study. Therefore, in this review, no terms have been selected from the outcomes point of view.

Once the search terms have been chosen, we combined them using the following logic formula in order to perform the search:

Population AND **Intervention**

A Boolean OR has been used among the terms in each viewpoint. Although some of the chosen terms are very generic (i.e. selection, runtime, capability, etc.) and seem to generate a great number of results within the search, this does not cause a major problem because these terms are used in conjunction (Boolean AND) with specific terms in the scope of SOA and software testing. Therefore, it can be assured that the majority of the found studies will deal with a topic of interest for this review.

Establishment of the search strategy

A three-phased strategy has been selected as the most proper way to perform the search. This strategy is shown in Figure 2.

----------Figure 2----------

In the first phase, a manual search has been carried out through the set of the most representative and specific journals, conferences and workshops that have previously published studies related with the research field. These sources are listed in Table 1. The selection of these sources has been performed after having reviewed the Impact Factor of the journals that are included in the Journal Citation Reports (JCR) of the ISI (Institute for Scientific Information) and the CORE rankings of conferences and workshops. We have also included other sources of information, based on our experience, where a great number of studies related to the paradigm of SOA and software testing have been published.

----------Table 1----------

In order to perform this manual search, the digital library DBLP (Digital Bibliography & Library Project) [16] has been used, where most of the publications of these proceedings are listed, and also the web pages themselves of these sources. For each (journal, conference and workshop), we have started our search in 2000, when the first specification of the Web Service Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) was published by the World Wide Web Consortium (W3C). A Service Oriented Architecture is built on the specifications of SOAP, WSDL and UDDI so we have considered that their publication year is a reasonable starting point from which to carry out the search. However, at the time of performing this search, a significant set of journal volumes and conference proceedings in 2010 had still not been published so we have agreed to finish our search in 2009. This means that the time span for this review is between the years 2000 and 2009.

During the second phase, the search through the electronic databases that are listed in Table 2 has been performed using the previously constructed search string. These

databases have been selected because they are known to cover most of the relevant journals and conference/workshop proceedings.

----------Table 2----------

Finally, a new search during the third phase of the review has been performed to complete the set of studies found during the first two phases. In this stage, the sources of the search have been the web pages of the researchers who wrote all the primary studies found during previous phases and the reference lists that are included in the studies found during the first two phases. Finally, we have also contacted all the corresponding authors of the selected primary studies asking for their expert opinion in order to identify other published studies that have been a useful contribution to the development of this mapping study.

3.3 Study selection criteria

During the search stage, some studies which do not address the research topic will be found. Hence, some criteria have been identified in order to include or exclude studies from the set of found ones according to our research topic. It has been decided to exclude:

C1: The studies that do not address the paradigm of SOA.
C2: The studies that do not address software testing.
C3: The studies that do not aim to test the dynamic binding.

The strategy for the selection of the final studies is as follows (Figure 3):

----------Figure 3----------

First of all, we have searched and stored all the studies found during the manual search within journals, conferences and workshops (first phase) and the automatic search through the databases listed in Table 2 (second phase). After each search, those studies that were found by more than one search source have been considered as duplicates and removed. In addition to this, we have agreed not to include in this review more than one study that belongs to the same research line. In those cases when the authors of these studies have periodically improved or completed their work in different sources, we have asked the authors for their most representative study, considering the previous studies as duplicates.
At a second stage, after having removed the duplicates, criteria C1 and C2 (only for studies found during the automatic search) have been applied to the title and abstract in order to exclude those studies that address neither software testing nor service oriented paradigm.
The following step involves joining the studies that have been found during the first phase with the set of studies, found within electronic databases, which have passed the title and abstract filter (C1 and C2). After that, the final criterion (C3) has been applied to the full text of the resultant set of studies in order to remove those that are not related to specific testing methods or techniques that aim to improve the trustworthiness of the applications where the discovery, selection and binding of the services are performed at runtime.

Finally, the selection strategy concludes with a manual search across the list of references of the previously found primary studies and the personal web pages of the authors in order to find representative studies that have not been discovered in the first two phases. We have also gathered those publications that have been recommended by experts in the field of software testing in SOA, such as the authors of the primary studies. As we stopped the search in the year 2009 and some of these received studies [22, 23, 24] were published during the first months of the year 2010, we have agreed not to include them as primary studies in this review. All of the studies found during the third phase of the search have been filtered (C3) in order to add the final primary studies for this review. In those cases where the detailed selection criteria were not so clear to decide whether a study should be removed or not, a consensus has been reached among the researchers.
Before applying our inclusion and exclusion criteria, we found 392 papers. Almost 37% (145) of the studies were identified as duplicate. After removing these duplicate studies, criteria C1 and C2 were applied to those studies that were found during the second phase. Of the 125 non-duplicate studies found during the aforementioned phase, only 28 passed both C1 and C2 and 97 studies were excluded (77.6%). After this process, criterion C3 was applied to 150 studies (101 from the first phase, 28 from the second phase and 21 from the third phase). A total of 117 (78%) of these studies did not pass this criterion, so 33 primary studies were selected. This value represents 8.41% of the total of studies found and 13.36% of the total of non-duplicate studies.

3.4 Study quality assessment and data extraction

In addition to the detailed inclusion/exclusion criteria, some issues have been considered in order to assess the quality of the primary studies. Each of the studies that passed all the filters applied during the selection strategy has been judged according to the following criteria. Some of them have been extracted from the quality criteria described in [4, 15, 25, 26].

- QA1: Is the reader able to understand the aim of the research?
- QA2: Does the paper include a discussion of related research?
- QA3: Is there a review about the related work of the problem?
- QA4: Is there a description of the testing method or technique used in the research?
- QA5: Has the approach been validated?
- QA6: Do the conclusions relate to the aim and purpose of research defined?
- QA7: Does the study recommend further research?

Each of these criteria has been graded on a dichotomous (“Yes” or “No”) scale [15, 26,] whether the primary studies covered them or not. Table 3 shows the results of applying the quality criteria to each primary study.
Two of the criteria (QA4 and QA5) are related to the description of the testing technique and the validation of the approach. The study of these two criteria is performed through the research questions RQ2 and RQ6 so discussion is to be found in Section 4 whereas in this quality assessment, we have only checked whether the primary studies fulfil the criteria or not.
Although some of the studies do not fulfil all of the quality criteria, we have decided to include all the studies within this review. This decision has been taken bearing in mind

that testing SOA with dynamic binding is a recent research topic so there are not many studies that address it. Hence we have tried not to miss any source of information. Furthermore, some of these studies have been published recently so it is very difficult to foresee their impact in the future in the scope of SOA testing.

----------Table 3----------

After having applied the quality assessment criteria, we have extracted the most relevant information from the set of finally selected primary studies. To enable this task and reduce the potential bias, a data extraction form has also been designed within the review protocol. This form is shown in Table 4.

----------Table 4----------

4 Results

In this section the results obtained from the selected primary studies are described according to the aforementioned research questions. Each subsection provides information to answer these questions regarding the objective of the studies, the testing technique applied, the execution time and the different stakeholders that take part in the testing process, the validation methods, the technologies and, finally, the distribution of studies per year and publication source. Throughout this section, different tables are shown to represent the results of this review. In most of them, some studies are represented in more than one category in each table. For instance, these studies may propose to use more than one testing technique or that different stakeholders take part in the testing process.

First of all, a brief summary of the selected primary studies is presented in Table 5. In the first column, the authors and the reference are identified while the second column contains a short description of the approach proposed in each study.

----------Table 5----------

4.1 Objective of the testing

All of the 33 primary studies have been selected because they propose the use of testing techniques or methods in order to improve the trustworthiness of the dynamic binding in SOA. The objective of these tests is to detect faults or obtain information to be used in future decision making. Hereafter, the main goal of each study (regarding the research question RQ1: What is the main objective of the research?) is represented in Table 6. The primary studies are represented in each row of the table. In the columns, the objective of each study and the system that is going to be tested are represented. Regarding the objective, we have identified if the study aims to detect faults in SOA with dynamic binding (fault detection) or to test the service executions in order to make a decision (decision making). In each of these approaches, the kind of properties that the study tests have been identified: functional and non-functional. If the testing is related to non-functional properties, we have also listed the explicit QoS attributes or metrics that are mentioned. Finally, in the last two columns of the table, we report the type of software that is going to be tested (system under test – SUT).

Regarding the objective, most of the studies (19) aim to detect faults. As can be seen in the table, twelve of these studies test functional properties and the same number of studies deals with non-functional properties. Within this set of studies, there are five ([30, 32, 38, 52, 53]) that propose to test both functional and non-functional properties. In contrast, there are fourteen studies that test services in order to make a decision based on the extracted information. Almost all of these studies (13) test services with respect to non-functional properties and only three test functional properties. Here again, there are two studies ([35, 48]) that address the testing of both functional and non-functional properties.

The results represented in Table 6 show that although there are a relevant number of studies (12) that test functional properties to detect faults, there is a lack of studies that make a decision about the dynamic binding based on the results of the functional tests (only three studies). Furthermore, most of these twelve studies do not specify the type of functional properties that they are going to test. For example, one of the few studies that mention a specific functional attribute is [34], where the objective is to test the interoperability between the set of services that are published in the registry and the new incoming services.

Twenty-five studies test a variety of different non-functional QoS attributes in the context of both fault detection (12 studies) and decision making (13 studies). As can be seen in Table 6, although nine studies did not identify a specific attribute, 16 studies identified individual QoS attributes. The most frequently noted attributes were response time (15) and availability (12) while cost (6), reliability (6), throughput (6), successful execution rate (4), reputation (4) and accuracy (1) were also mentioned. We have identified that the primary studies do not present alignment with any standard model, apart from [47] which is aligned with ISO/IEC 9126 Standard Quality Model [60]. Despite not having been aligned, most of the attributes that have been identified in the primary studies can be mapped to the same characteristics that are specified in this standard.

----------Table 6----------

Apart from the main goal of each primary study, the extracted information about the type of software that is going to be tested is shown in the last two columns of Table 6. More than three quarters (26) of the primary studies test individual services, so more emphasis is being dedicated to testing atomic services as opposed to service compositions. Several actions are commonly performed with the results of these tests. For example, the publication of the services in the registry can be forbidden if the test execution is not successful [41, 53] or the services can be deleted from the registry if they have failed tests [30, 54]. These actions provide some confidence to clients because it is assured that services published in the registry have been forced to pass some tests. In addition, information gathered from the tests about the QoS of each service can be stored and made publically available to the clients [27, 39]. As is shown in the table, almost in a half of the approaches (16), a service composition is the system under test. In these studies, the most common goal involves monitoring the service composition to detect any fault or violation in the Service Level Agreement (SLA) in order to perform self-healing actions, for example, the dynamic re-binding to a different service [32, 33, 36, 38, 45, 56]. Here again, the sum (26 + 16) of these classified studies is greater than the total of primary studies (33) because there are nine studies that allow testing both atomic services and service compositions.

4.2 Testing technique

Regarding the research question RQ2 (Which testing techniques and methods are used?), we have extracted the specific testing technique which is applied in each one of the primary studies. This information is represented in Table 7 where the number of studies that use such techniques is also shown.
Two thirds of the primary studies (23) propose monitoring-based techniques. Almost 80% of these studies (18) use online monitoring [61] so the system can trigger adaptive action to recover from a faulty situation. In contrast, eight studies utilise offline monitoring because they gather data while the service executes but use the data later. In addition to this, four of these studies ([35, 39, 40, 42]) perform a technique known as feedback based monitoring. In these four studies, monitoring is based on client feedback because the client monitors his executions and sends information about them to a registry or a broker. As this information is stored and used in the future, this specific technique is included in the offline testing approaches. A combination of both online and offline monitoring is proposed in [39] and [42].
In spite of the fact that the generation of test cases is proposed in ten studies within this review, only four of them ([43, 48, 54, 41]) describe a specific technique to derive these test cases. In the first two, traditional partition testing is used to derive test cases from the WSDL and a document where the behaviour of those services are specified using Graph Transformation Rules. In the third, a technique named Swiss Cheese, which is described in [62], analyzes the specification and generates the test cases. This technique allows both positive and negative testing in order to verify the required functionality and assure the robustness of the services. In the latter, the Stream X-machines Testing Method is proposed. This method is a generalization of the W-Method [63] and allows the generation of test cases from the specification of the services using Stream X-machines [64]. In the remaining six studies no specific technique is mentioned although some approaches propose different artefacts that can be used to derive the test cases: the WSDL document [29, 52], a BPEL or OWL-S specification [29] and UML 2.0 diagrams [34]. Furthermore, there are studies that allow using an already existing test case generation technique or even ad-hoc testing, without specifying the testing technique.
A couple of studies propose a technique named Group Testing that allows testing groups of web services which share the same functionality and ranking these services according to the test results and different ranking strategies.
Finally, there are three studies that use testing in order to supply a quality driven selection mechanism. In these studies, web services are executed to gather QoS attributes values although these approaches are not based on a specific testing technique.

----------Table 7----------

4.3 Stakeholders and points in time

The responsibility of testing in SOA is shared among the different stakeholders that interoperate through the Internet to use or provide services [65]. The common SOA architecture (Figure 1) includes providers (responsible for delivering the services), registries (responsible for storing, browsing and retrieving services) and clients (final users of the services). In addition to this, a new third-party entity named broker takes

part in the SOA testing process. Brokers act as independent and objective entities that aim to provide more confidence in the results of the tests.
In the context of software testing, the needs, advantages and drawbacks of these and other stakeholders (developer, provider, integrator, third party and client) have been identified by Canfora and Di Penta in [2]. Regarding these testing perspectives, we have united the developer, provider and integrator in just one entity as all the primary studies selected in this review refer to them as a generic provider. According to this classification, the third party would include both the registry and the broker but we have considered the registry as an independent entity because there are a relevant number of papers that explicitly mention it. Hence, the broker plays the role of the third-party certifier in this review.
In Figure 4 the typical SOA triangle is extended with the role of the broker that participates in testing. Furthermore, the different points in time when the tests are performed (from t1 to t4) are also represented. According to the information extracted from the primary studies, the service registration process is considered to be the first point in time to test the new services (t1). While services are published in the registry, such services stand deployed in the provider infrastructure and both the registry and clients may not be informed when changes occur in the deployed services. For example, the implementation of the service may be modified or a service may become unavailable. So this is the second point in time (t2) identified as suitable to perform the tests. Once the services are published, a client makes its request to the registry in order to find a set of services that fulfil its requirements. Hence, the client has to make a decision about the selection of one of the retrieved services to bind it. This is the third point in time (t3) when dynamic binding testing is performed. Finally, the last point in time that has been identified (t4) as suitable to execute tests is during the execution of the services performed by the clients.

----------Figure 4----------

Stakeholders

Both the stakeholders and the points in time are represented in Table 8. For each primary study represented in a row, we have identified in columns which stakeholders take part in the tests (regarding RQ3: Which of the different stakeholders take part in the testing processes?) and the points in time when these tests are performed (regarding RQ4: When are the tests performed?).
As it can be seen in Table 8, 12 studies consider more than one stakeholder role to be involved in testing. The client is the most frequently cited stakeholder appearing in almost three quarters (24) of the primary studies, whereas the registry role is cited in more than a third of the studies (12). The UDDI standard for web services registry does not actually support specific testing capabilities, it only provides storage, browsing and retrieval features. To support testing, some studies propose to extend UDDI registries with additional capabilities, for instance, extra storage to keep QoS information or a new discovery algorithm to browse and retrieve services according to a QoS model. Regarding the other two stakeholders, a third of the studies (11) propose using a broker in the testing process, to provide independent test results to the client. Finally only four studies suggest that the service provider should perform tests directly related to improving the dynamic binding.

Points in time

Regarding the test points in time, twenty-one of the studies propose to test the services during their execution (t4) whereas other points in time (t1, t2 and t3) are proposed in far fewer studies.
Nine studies propose testing the services before their publication in the registry. Seven of these studies, which are represented in the t1 column, execute tests as a requirement to publish the services supplied by the provider in the registry while the other two studies perform tests in order to obtain QoS attribute values and store this information in the registry.
Seven studies propose executing tests periodically while services are published in the registry (t2). These studies identify different reasons for such testing. It may be to eliminate from the registry those services that do not provide the expected test results and to help the client to select the most valuable service (4 studies). Alternatively the tests can gather information about the quality of service (QoS) levels in order to store it in the registry (3 studies). This information can be used by the client when selecting among a group of services which provide the same functionality. Although QoS values can be supplied by the service provider, it is more reliable to obtain the information by testing the services. A third of the studies (11) execute tests and gather information in order to select the best service from a set of potential candidates. This set of studies is represented in the t3 column.
Finally, all of the twenty-three studies from the t4 column use monitoring based testing techniques to detect faults during the execution of the service based system or to gather information such as QoS attribute values. A common objective is that the system can perform corrective action (self-healing) to recover from any misbehaviour during the execution or to gather QoS metrics that will help during the decision making. These monitoring tasks are commonly performed by the client of the application so it can be seen that the values of the t4 column are practically a subset of the values of the client's column.

----------Table 8----------

Participation of each stakeholder in the points in time

In each of the aforementioned points in time (t1, t2, t3 and t4), different stakeholders are proposed to take part in the testing process so it is interesting to identify which of the stakeholders participates in the tests in each point in time. This information is shown in Table 9, where the stakeholders are represented in rows and the points in time when the tests are performed in columns. The reference of the primary study is represented in a cell if the stakeholder in that row participates in the tests during the point in time specified in that column.

----------Table 9----------

Before the registration of the services (t1), both the provider and the registry and even an independent broker may participate in the tests. Obviously, a client cannot execute tests in this point in time because he does not know the binding information until the service has been published in the registry. As can be seen in the t1 column, this task can be carried out by an enhanced registry with testing capabilities (eight studies).
Regarding the point in time when the services are published in the registry (t2), no study in this review proposes that the service provider executes tests in this point in time.

Both the registry (five studies) and the broker (three studies) are in charge of executing tests of the services that are published. The results of these tests allow the deletion of services from the registry when a problem is detected, for example, when a service has become unavailable. Furthermore, QoS data are gathered through the tests and stored publicly in the registry so clients can check it when the decision making has to be performed, for instance in [27].
All the eleven studies that propose executing tests just before the binding (t3) share the same objective: provide additional information that helps the client in the decision making. At this point in time, the provider does not take part in the testing process because it has previously published its services in the registry and it is unaware who is going to invoke these services. Hence, the cell that represents the union of the provider and the t3 point in time in the table is empty. Although both the registry and the broker are also proposed to perform tests within this point in time, in most of the studies that are represented in the t3 column (9), it is the client who performs the tests in order to obtain relevant information that allows him to select the best service.
All studies that are represented within the t4 column of the table propose to monitor service execution. In seventeen of these studies, only one stakeholder takes part in monitoring task: the client (13 studies), a broker (three studies) or a registry (one study). However, there are six studies where there is more than only one stakeholder participating in the testing process. A paradigmatic example of these approaches is [42] where a registry stores QoS information which it gathers from the service provider, monitored data from a third party and feedback from the client's execution. As can be seen, during service execution, the client is the most frequently cited stakeholder (19 studies). This represents that in more than half of the primary studies, the client is proposed to perform tests during the service execution using monitoring techniques.

4.4 Technologies

The set of technologies and standards that are used within the testing processes in SOA with dynamic binding (regarding the research question RQ5: Which are the most common technologies and standards being researched?) are identified and listed in Table 10. We have organized them in a hierarchy according to their objective. First of all, we have identified such languages that are used to describe the atomic services and the behaviour of the composite services. Furthermore, there are almost twenty primary studies that use a registry to publish the services so we have identified how these registries are specified. Finally, it has been considered relevant to identify the different languages that are used to describe the terms agreed between the provider and the client in an SLA. This hierarchy of both technologies and standards is represented in Table 10.

----------Table 10----------

Though the description of the atomic services has been specified in different languages within the set of primary studies, most of these studies use WSDL for this specification. For example, three works generate test cases based on the web services description specified in WSDL language [29, 52, 53] while another study extend this WSDL description with QoS attributes [39]. In addition to this, three studies use semantic technologies such as OWL-S (Ontology Web Language for Web Services) [66] and SAWSDL (Semantic Annotation for WSDL) [67] to describe service behaviour. The latter study uses a Stream-X machines model that provides the description and it is attached to the service using the SAWSDL document.

Another of the main features of SOA is the chance to compose different services to provide certain functionality. The BPEL language [68] has been standardised by OASIS and it is broadly used to specify web service compositions. In eight of the studies, the SUT is a service composition specified in BPEL language. Furthermore, in one study OWL-S is used to represent the service composition process model. Regarding the rest of the studies where the system under test (SUT) is both an atomic service and a service composition (Table 6), the language used to describe such services is not specified because it is not relevant for testing purposes.

Eighteen studies propose publishing services in a registry. This number is higher than the one represented in Table 8 because there are some studies where the registry plays a passive role in testing, only storing and retrieving the services. On the other hand, there are scenarios where the registry is extended with extra capabilities that allow it to play an active role in the testing process. Although most of the studies (13) use UDDI as a standard language, five studies that discussed use of a registry did not specify any standard language.

Many studies test services to detect if there is a violation in the SLA established between the service provider and the client. Although SLAs are mentioned in seven studies, only in [38], [40] and [50] a language to specify them is identified. In the first two cases, WSLA is the language used to describe the agreements whilst in the third WS-Policy is used to specify policy assertions about the quality of the services. Another language that is being currently used in software testing to specify service level agreements [69, 70] is WS-Agreement [71] but it has not been used in any of the selected primary studies.

In each of the first three categories shown in Table 10, there is one technology that is used by a majority of studies: the atomic service description is commonly specified using WSDL, the behaviour of the composite services is often described by means of BPEL language and the functionalities of the registry are specified using UDDI. However, for specification of the SLAs there is no standard language.

4.5 Validation

Researchers use different methods to demonstrate that their approaches are valid. Regarding the research question RQ6 (What method is being used to validate the research?), we have identified the method that has been used to validate each approach in all the primary studies, using the five different categories that Shaw proposes in [72]: analysis (for example, carefully designing experiments with statistically significant results), experience based on real-world scenarios, evaluation using feasibility studies or pilot projects, realistic or standard examples and persuasion. We have considered opinion as a validation method in the same category as persuasion. In addition to this, most of the studies supply examples to describe their approach or evaluate if the proposal is a valid one. Hence, the types of examples that are being used have also been identified. In Table 11 each primary study is represented in rows while all of the validation methods and types of examples are listed in columns.

----------Table 11----------

We found only two studies where an extensive analysis is performed. In the first one [36], service compositions are executed using different re-binding techniques in order to analyze the variation of QoS of the system when these techniques are applied. The other

study [44] performs an empirical assessment about five criteria that aim to select the best web service.
Regarding the validation method, it is also relevant that experience has only been used in two studies. In the first of these studies, Bai et al. have applied their approach in the Hard X-ray Modulation Telescope (HXMT) Project [30]. They have implemented their monitoring infrastructure in a satellite grand application project where the data centre receives and processes a huge amount of information. Service interfaces have been instrumented with sensors and monitoring agents are deployed on different hosts to gather the information. In the second study, Ben Halima et al. [33] have implemented and monitored a real WS-based complex application on the French grid Grid500 in order to provide self-healing strategies.
On the other hand, fourteen studies extract some quantitative information from the execution of their approaches in order to evaluate the results of the tests. According to its definition in [72], evaluation is considered as the method applied to validate these studies. All of these studies perform this evaluation after having gathered information through the execution of examples. However, there are six studies that propose an example in order to illustrate their approach but neither analysis nor evaluation is performed. These studies are represented in the Examples column of Table 11.
Finally, there is a significant group of studies (9) that do not describe their validation method. These studies are represented in the Opinion/Persuasion column of Table 11. A possible reason for this lack of validation can be because of the early state of the research so some original ideas or approaches are proposed but the technique has not been implemented or tested yet.
Regarding the types of examples that are being used in the primary studies, in more than half of the cases (15), these examples are designed ad hoc and they are not reused from any previous work so it is impossible to compare the results with different testing approaches. On the other hand, there is one study [59] that reuses a travel planning application that has later been used by different researchers in other contexts [73]. Also, the example proposed in [32] has later been used in [35]. In addition to this, there is a group of studies (6) that use public services in order to test and evaluate their approaches. Finally, real scenarios are used in two studies to validate their approaches [30, 33] and an example extracted from the BPEL standard specification has been used in one study [35].
Considering the results, evaluation is the most used method to validate the works proposed in the primary studies while analysis and experience have been used in a reduced number of studies. Furthermore, a third of these works do not present any type of validation. Bearing in mind the examples, it is relevant to mention that only one of the studies has used an example described in a standard specification and two studies have applied their approach to realistic examples. With the use of complex and realistic standard examples, it would be possible to compare different studies with the same objective and evaluate whether the new approaches present better results than the older ones.

4.6 Distribution of studies

In addition to the specific analysis of the primary studies about testing features which has been presented in the previous sections, we also extracted both the date and the source where each study was published. With this information, it should be possible to assess whether there are trends that imply that interest in research on testing SOA with dynamic binding is increasing and whether enough effort has been dedicated to it during

recent years (regarding RQ7: How much activity about dynamic binding SOA testing research has there been during recent years?). Furthermore, it is also interesting to analyze where these studies have been published: the type of study (journal article, conference/workshop paper or book chapter) and the source (regarding RQ8: Where have the researches been published?). In Table 12 this information is summarized for each primary study. In the first column, the primary study is identified while the publication year is listed in the second column. The last two columns of the table represent the type of publication and the source, respectively.

----------Table 12----------

Table 13 reports the number of primary studies published per year between 2003 and 2009.

----------Table 13----------

Although for this review we have considered studies that have been published since the year 2000, no primary study was found until three years later (2003). This may be because SOA were only beginning to be investigated in the early 2000s and the need for testing techniques capabilities such as dynamic binding had not yet been recognized. During the next few years a similar number of studies were found ranging between three and six works per year. It is in 2008 where we have found a relevant increase in the number of approaches (12 studies). This fact indicates that testing SOA with dynamic binding is a recent area of research and more effort is being dedicated to adapt classic techniques or propose new approaches in this field. However, during 2009 only three have been found related to the research topic. Hence, it would be necessary to observe whether this trend is maintained during the next few years and the testing of SOA with dynamic binding will still be a challenging and promising task or, on the other hand, whether the peak in the number of studies found in 2008 was an exceptional situation.

Table 14 represents the number of primary studies that have been selected according to their publication source.

----------Table 14----------

The first row of Table 14 indicates that six primary studies were published as journal articles. Journal of Systems and Software (JSS) was the only journal that published more than one primary study (2) while the rest of these primary studies were published in different journals. As can be seen in the second row, more than half of the primary studies (21) have been published in conference proceedings. Seven of these twenty-one studies were presented in the IEEE Computer Software and Applications Conference (COMPSAC). Furthermore, the IEEE International Conference on Web Services (ICWS) and the International World Wide Web Conferences (WWW) published five and three studies respectively. Workshops have also been the source of four primary studies of this review. Two of these workshops are specifically focused on testing research: Web Service: Modelling and Testing (WS-MATE) and Monitoring, Adaptation and Beyond (MONA+). Finally, two of the works that have been selected as primary studies were published as a book chapter. The content of one of these books [74] addresses specific approaches related to SOA verification and validation.

Taking this data into account, it can be said that there are not many studies that address the problem of testing Service Oriented Architectures where the discovery and selection of services are performed at runtime. However, as we have focused on a novel and specialized research topic, the number of primary studies can be considered significant. On the other hand, none of the primary studies has been published in either journals or conferences that focus on testing topics, for example, the Software Testing, Verification and Reliability journal (STVR), the International Conference on Software Testing (ICST) or the International Symposium on Software Testing and Analysis (ISSTA).

5 Discussion

5.1 Summary of reviewed studies

As a result of this review, the objectives of testing are grouped into two categories: studies that aim to detect faults in the service oriented application (57.57%) or studies that make a decision about the service to be invoked based on the test results (42.42%). The proposed testing approaches focus more on non-functional characteristics rather than on functional.

Regarding the applied testing techniques, the results of this review show that, currently, two thirds of the studies apply monitoring approaches to improve the dynamic binding. These approaches check properties of the executing system in order to perform an adaptive action (for example, rebind to another service) when a deviation from the expected behaviour is detected. These properties may be both functional and non-functional. In addition, there are ten studies that generate and execute test cases with the aim of detecting problems or gathering data to make a decision about the binding.

Although there are different stakeholders that participate in the testing process, it is the client who plays the most active role with 24 of the 33 studies proposing that the client takes part in the tests. The registry and a broker also represent stakeholders that are often proposed to take part in the tests. However, only four studies suggested that the service providers should participate in the dynamic binding testing process.

This review has identified four points in time when the execution of tests may improve the dynamic binding of the services. Service execution is the most frequently recommended point in time to perform tests using monitoring techniques. The points in time before the publication of the services in a registry, during the time they are published and just before their binding are also considered as suitable times to test services.

The description of the atomic services and service compositions that represent the system under test is almost always specified using WSDL for the former and BPEL for the latter, although there are a reduced number of examples that use semantic technologies such as OWL-S or SAWSDL. In such studies where a registry is in charge of storing the services, the UDDI standard is the only mentioned specification. However, in the context of testing SOA with dynamic binding, there is no standard language for representing the terms agreed in an SLA.

The validation methods used in the primary studies have several limitations. Firstly, almost a third of these studies do not present any type of validation of the proposed approach. In addition, the most frequently used validation method is evaluation, with very few studies performing a rigorous analysis of their results and only two studies applying the approach to a real scenario. Most of the examples used in the studies were designed ad hoc and in only one case has the example been extracted from a standard specification.

5.2 Related work

Testing the dynamic binding in SOA is a very specific topic and, to the best of our knowledge, this is the first article which identifies and classifies the available research on testing service-based software with dynamic binding. In [1] Canfora and Di Penta present a survey of testing Service Oriented Architectures. They analyze, as we do, recent research from the viewpoints of different stakeholders and classify it into four testing levels: functional, regression, integration and non-functional. Although they cover particular characteristics of these systems, they do not specifically focus on any, whereas we address dynamic binding. Bozkurt et al. [75] present a survey that extends [1], classifying research according to the testing techniques used, as also identified on our research, and including areas not covered by Canfora and Di Penta's survey.
A systematic review about formal approaches to test service-based software is provided by Endo and Simao in [76]. They focus on specific formal methods to test both atomic services and service compositions and they analyze where and when these studies have been published, as we also consider in this review. The main difference with our study is that their review is restricted by the testing technique applied (formal methods) whereas our review aims at identifying research that addresses the improvement of the dynamic binding using any testing approach. A more specific study is [77], where Zakaria et al. perform a systematic review about testing web service compositions specified in BPEL language from a unit testing level. In our review, we consider both web service compositions and atomic services. Furthermore, we have not established constraints based on the technology or specification language to perform the search. Finally, a systematic review about QoS in SOA systems is performed by Oriol in [78]. This study focuses on quality attributes for web services and they review approaches that use monitoring techniques to obtain the value of these attributes at runtime.

5.3 Limitations of this review

This mapping study has been performed systematically following a protocol that has been developed to avoid bias within the search and selection process. However, there are a number of limitations due to different factors. First of all, a review protocol that contains the research questions has been developed. These questions guide the selection of the search terms that enable to identify the existing literature. These search terms and keywords are selected before starting the search so there is a risk that new relevant terms can be identified during the review process. In order to mitigate this risk, the protocol has been modified and refined during the review process. Furthermore, a significant group of information sources has been selected to perform the search, including electronic databases and the most relevant journals, conferences and workshops proceedings that publish studies in the scope of SOA testing. However, it is possible that relevant studies, according to the scope of this review, may have been omitted if they were published in sources not selected in the protocol.
With respect to assessing the quality of primary studies, some of the quality criteria could only be assessed subjectively, potentially reducing the accuracy of the quality assessment.
In the context of data extraction some detailed information was missing. The main problems were found in the descriptions of the testing techniques applied and the validation methods used by each study. Hence, although a predefined extraction form has been previously designed (Table 4), the testing technique and the correctness of the validation methods cannot be precisely identified from the reading of the studies.

Our decision to include only one study for the same research line may mean another limitation of this review. This situation has occurred when more than one study has been published improving or completing a previous work. We have tried to mitigate this limitation asking the authors of the primary studies for their opinion. In such cases, the most complete and representative study has been included in the set of primary studies of this review. Hence, the results concerning the final number of studies and the distribution of such studies per publication source and year could have been slightly different.

6 Conclusions and further work

A systematic mapping study has been performed to select and evaluate the available research in the scope of testing Service Oriented Architectures where the discovery and binding of the services are performed at runtime (dynamic binding). From an initial set of 392 studies, 33 have been selected as primary studies. Due to the specialized nature of the topic of this review, this can be considered as a significant number of studies. The sources of information have been electronic databases, journal articles and conference/workshop proceedings as well as expert opinion, list of references and authors' web pages.

A review protocol has been designed to ensure that the selection process was unbiased. The selection of the studies has been performed according to a search process that includes the identification of the most suitable terms and the establishment of a search strategy. From each final primary study, a studio quality assessment has been performed and information has been extracted in order to answer the set of previously defined research questions. The synthesis of the results has allowed identifying the objective and the testing techniques applied in each approach. Furthermore, the stakeholders that participate in the tests and the points in time when this process is carried out have been described. We have also discussed the most common technologies and standards used in this field as well as the validation methods of each study. Statistical information about the source and the publication year of each study has also been provided.

The results of this mapping study may offer additional information to researchers who are interested in improving the quality of the dynamic binding in SOA using testing approaches and may contribute to provide a body of knowledge that allows identifying current gaps in this research topic. First of all, we have outlined that most of the studies reviewed in this work propose the use of monitoring techniques [79] that check the behaviour of the system during execution. These are reactive approaches because they detect faults in a service oriented application during its operation in order to trigger adaptive actions, so the faults are discovered very late in the life cycle. However, it is not adequate to deploy an application in the production environment without having previously performed a set of tests that assure a minimum level of quality in both functional and non-functional properties. Hence, it could be useful to develop proactive approaches [80] to design test requirements for SOA applications with dynamic binding, and execute tests before their operational deployment. Also after deployment, the test requirements may facilitate the identification of singular conditions that are not exercised during the usual invocations of the clients. These conditions could go undetected by monitoring approaches, but may represent potential problems in future executions.

According to the results of the review, the client is the most active stakeholder monitoring the services during their executions. However, it is the provider who is in charge of assuring that he is capable of providing the functionality and the quality of service agreed with the client. This quality assurance may be performed through the

design and execution of tests before the publication of the services in a registry and also executing regression tests while the services are published in such registry. For example, in [53] the provider can obtain from the registry a set of test cases designed by clients or other service providers in order to assure that such services meet the requirements and they are ready to be dynamically bound by the clients. However, these approaches have not yet received enough effort, according to the results of this mapping study.

A common problem that may derive in a dynamic re-binding of the service arises when the client detects a violation in the terms specified in the SLA with the provider. Although there are studies that use monitoring to detect violations in the SLA [81], there are very few studies, for example Di Penta et al. [82], where the generation and execution of test cases allow the detection of problems that may later result in SLA violations in service compositions. In such a scenario, probabilistic approaches can be applied to the test results with the aim of detecting or foreseeing situations that are on the verge of causing future violations of the agreement terms. Alternatively, such violations may be simulated during the tests with the aim of checking whether the software triggers suitable adaptive tasks (for example, re-binding to another service) immediately the problems are detected.

The use of standard technologies for testing purposes has also been analysed within the primary studies. The results show that BPEL and UDDI are the most accepted standard to orchestrate the behaviour of the service compositions and to specify the functionalities of the registry, respectively. However, it is worth mentioning that there does not seem to be a *de facto* standard language to specify the conditions of a service level agreement so different technologies are proposed but there is not a relevant set of studies within this review that use the same SLA specification language for testing purposes. Regarding the service description, there are not many studies that propose the use of semantic technologies with the aim of describing the features of the services. In addition to this, UDDI registries do not support the search and retrieval of services using queries with semantic content so it is not possible either for the client or the registry itself to know if one service is going to provide the expected functionality. Hence, it would be interesting to apply techniques that allow the testing of services that are described using standard semantic technologies when they are published in a registry.

Although there are a relevant number of studies that aim to detect faults testing functional properties of the system, almost all the studies that test services in order to make a decision about the service to be invoked are related with non-functional characteristics. Thus, it would be an interesting research line to design methods or techniques to test functional characteristics so the decision making will also take into account the results of these functional tests. Here again, the use of semantic technologies may be used to specify the functional properties of the services.

Testing SOA with dynamic binding is a relatively new topic and much research has not been yet validated so we conclude that it is necessary to dedicate more effort to the validation process. In addition to this, it would also be interesting to use real scenarios or standard examples, with the aim of being able to make comparisons with other closely related studies.

Acknowledgements

We are grateful to Barbara Kitchenham and the anonymous reviewers for their useful comments and suggestions which have helped us improve this work. We would also

like to thank all the cited authors who received a preliminary version of the paper and responded with their feedback, often in considerable detail.
This work was partially funded by the Department of Science and Innovation (Spain) and ERDF funds within the National Program for Research, Development and Innovation, projects Test4SOA (TIN2007-67843-C06-01) and Test4DBS (TIN2010-20057 -C03-01) and FICYT (Government of the Principality of Asturias) Grant BP09-075.

References

[1] G. Canfora, M. Di Penta, Service-oriented architectures testing: a survey, in: A. De Lucia and F. Ferrucci (Eds.): ISSSE 2006-2008, LNCS 5413, 2009, pp. 18-105.

[2] G. Canfora, M. Di Penta, Testing services and service-centric systems: Challenges and opportunities, in IT Professional 8 (2) (2006)10–17.

[3] D. Budgen, M. Turner, P. Brereton, B.A. Kitchenham, Using mapping studies in Software Engineering, in: Proceedings of PPIG, Lancaster University, 2008, pp. 195-204.

[4] B.A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Technical Report, EBSE-2007-001, UK, July 2007.

[5] B.A. Kitchenham, D. Budgen, P. Brereton, The value of mapping studies – A participant-observer case study, in: EASE'10: Proceedings of Evaluation and Assessment in Software Engineering, 2010.

[6] M. Papazoglou, P. Traverso, S. Dustdar, F. Leymann, Service-oriented computing: State of the art and research challenges, in: IEEE Computer 40 (11) (2007) 38-45.

[7] Simple Object Access Protocol (SOAP). http://www.w3.org/TR/2007/REC-soap12-part0-20070427/ (Accessed April 2010)

[8] Web Service Description Language (WSDL). http://www.w3.org/TR/wsdl20/ (Accessed April 2010)

[9] Universal Description, Discovery and Integration (UDDI). http://www.uddi.org/pubs/uddi_v3.htm (Accessed April 2010)

[10] M. Papazoglou, W.J.A.M. van den Heuvel, Service oriented architectures: Approaches, technologies and research issues, in: Very Large Database Journal 16 (3) (2007) 389-415.

[11] ISO/IEC 24765, "Software and Systems Engineering Vocabulary," 2006.

[12] E. Di Nitto, C. Ghezzi, A. Metzger, M. Papazoglou, K. Phol, A journey of highly dynamic, self-adaptive service-based applications, in: Automated Software Engineering 15 (3-4) (2008) 313-341.

[13] A. Bertolino, Software testing research: Achievement, challenges and dreams, in: FOSE'07: Future of Software Engineering, 2007, pp. 85-103.

[14] B.A. Kitchenham, Procedures for performing systematic reviews, Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.

[15] M. Staples, M. Niazi, Experiences using systematic review guidelines, in: Journal of Systems and Software 80 (9) (2007) 1425-1437.

[16] DBLP: Digital Bibliography & Library Project. http://www.informatik.uni-trier.de/~ley/db (Accessed April 2010)

[17] IEEE Xplore. http://ieeexplore.ieee.org (Accessed April 2010)

[18] ACM Digital Library. http://portal.acm.org/dl.cfm (Accessed April 2010)

[19] Scopus. http://www.scopus.com/home.url (Accessed April 2010)

[20] EI Compendex. http://www.engineeringvillage2.org (Accessed April 2010)

[21] ISI Web of Knowledge. http://www.isiknowledge.com (Accessed April 2010)

[22] L. Baresi, S. Guinea, Self-supervising BPEL processes, in: IEEE Transactions on Software Engineering, 02 Mar. 2010. IEEE Computer Society Digital Library.

[23] D. Kourtesis, E. Ramollari, D. Dranidis, I. Paraskakis, Increased reliability in SOA - environments through registry-based conformance testing of Web services, in: Production Planning & Control: The Management of Operations 21 (2) (2010) 130-144.

[24] D. Bianculli, W. Binder, M. L. Drago, Automated performance assessment for service-oriented middleware: a case study on BPEL engines, in: WWW-2010: Proceedings of the 19th International World Wide Web Conference, 2010.
[25] W. Afzal, R. Torkar, R. Feldt, A systematic review of search-based testing for non functional system properties, in Information and Software Technology 51 (2009) 957-976.
[26] T. Dybå, T. Dingsøyr, Empirical studies of agile software development: A systematic review, in: Information and Software Technology 50 (2008) 833-859.
[27] E. Al-Masri, Q. Mahmoud, Toward quality driven web service discovery, IT Professional, 10 (3) (May/June 2008) 24-28.
[28] X. Bai, Y. Chen, Z. Shao, Adaptive web services testing, in: COMPSAC'07: Proceedings of the 31st Annual International Computer Software and Applications Conference, IEEE Computer Society 2007, pp. 233-236.
[29] X. Bai, D. Xu, G. Dai, W.T. Tsai, Y. Chen, Dynamic reconfigurable testing of service oriented architecture, in: COMPSAC'07: Proceedings of the 31st Annual International Computer Software and Applications Conference, IEEE Computer Society 2007, pp. 368-378.
[30] X. Bai, S. Lee, W.T. Tsai, Y. Chen, Collaborative web services monitoring with active service broker, in: COMPSAC'08: Proceedings of the 32nd Annual International Computer Software and Applications Conference, IEEE Computer Society 2008, pp. 84-91.
[31] W.T. Balke, J. Diederich, A quality and cost based selection model for multimedia service composition in mobile environments, in: ICWS'06: Proceedings of the IEEE International Conference on Web Services, 2006, pp. 621-628.
[32] L. Baresi, S. Guinea, L. Pasquale, Self-healing BPEL processes with Dynamo and the Jboss rule engine, in: ESSPE'07: International Workshop on Engineering of Software Services for Pervasive Environments, 2007, pp. 11-20.
[33] R. Ben Halima, K. Drira, M. Jmaiel, A QoS-oriented reconfigurable middleware for self-healing web services, in: ICWS'08: Proceedings of the IEEE International Conference on Web Services, 2008, pp. 104-111.
[34] A. Bertolino, A. Polini, The audition framework for testing web services interoperability, in: 31st EUROMICRO Conference on Software Engineering and Advanced Applications, 2005, pp. 134-142
[35] D. Bianculli, W. Binder, L. Drago, C. Ghezzi, Transparent reputation management for composite web services, in: ICWS'08: Proceedings of the IEEE International Conference on Web Services, IEEE Computer Society Press, September 2008, pp. 621-628.
[36] G. Canfora, M. Di Penta, R. Esposito, M.L. Villani, A framework for QoS-aware binding and re-binding of composite web services, The Journal of Systems and Software 81 (2008) 1754-1769.
[37] M.D. Ernst, R. Lencevicius, J.H. Perkins, Detection of web service substitutability and composability, in: WS-MATE'06: International Workshop on Web Services Modelling and Testing, 2006, pp. 123-135.
[38] A. Erradi, P. Maheswari, V. Tosic, WS-Policy based monitoring of composite web services, in: ECOWS'07: The 5th European Conference on Web Services, 2007, pp. 99-108.
[39] A. Gorbenko, A. Romanovsky, V. Kharchenko, How to enhance UDDI with dependability capabilities, in: COMPSAC'08: Proceedings of the 32nd Annual International Computer Software and Applications Conference, IEEE Computer Society 2008, pp. 1023-1028.

[40] R. Jurca, W. Binder, B. Faltings, Reliable QoS monitoring based on client feedback, in WWW'07: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 1003-1012.
[41] D. Kourtesis, E. Ramollari, D. Dranidis, I. Paraskakis, Discovery and selection of certified web services through registry-based testing and verification, in: Camarinha-Matos L. and Pickard W. (Eds.): Pervasive Collaborative Networks, IFIP, Vol. 283, Springer Boston, 2008, pp. 473-482.
[42] Y. Liu, A.H. Ngu, L. Zeng, QoS Computation and policing in dynamic web service selection., in WWW'04: Proceedings of the 13th international World Wide Web Conference, 2004, pp. 66-73.
[43] M. Lohmann, L. Mariani, R. Heckel, A Model-driven approach to discovery, testing, and monitoring of web services, in: Test and Analysis of Web Services, (eds) L. Baresi, E. diNitto, Springer, 2007, pp. 173-204.
[44] N.C. Mendonça, J.A.F. Silva, R.O. Anido, Client-side selection of replicated web services - An empirical assessment, The Journal of Systems and Software 81 (2008) 1346-1363.
[45] O. Moser, F. Rosenberg, S. Dustdar, Non-intrusive monitoring and service adaptation for WS-BPEL, in WWW'08: Proceedings of the 17th International Conference on World Wide Web, 2008, pp. 21-25.
[46] A. Mosincat, W. Binder, Enhancing BPEL processes with self-tuning behavior, in: SOCA'09: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications, 2009.
[47] M. Oriol, J. Marco, X. Franch, D. Ameller, Monitoring adaptable SOA systems using SALMon, in: MONA+: Workshop of Service Monitoring, Adaptation and Beyond, in ServiceWave Conference, 2008.
[48] Y. Park, W. Jung, B. Lee, C. Wu, Automatic discovery of web services based on dynamic black-box testing, in COMPSAC'09: Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, IEEE Computer Society 2009, vol. 1, pp. 107-114.
[49] S. Ran, A model for web services discovery with QoS, ACM SIGecom Exchanges 4 (1) (2003) 1-10.
[50] Y. Seo, H. Jeong, Y. Song, A study on web services selection method based on the negotiation through quality broker A MAUT-based approach, in: Z.Wu (Eds.): ICESS 2004, LNCS 3605, 2005, pp. 65-73.
[51] T. Szydlo, K. Zielinski, Method of adaptive quality control in service oriented architectures, in: M. Bubak et al. (Eds.): ICSS 2008, Part I, LNCS 5101, 2008, pp. 307-316.
[52] D. Tosi, G. Denaro, M. Pezzè, Towards autonomic service-oriented applications, International Journal of Autonomic Computing 1 (1) (2009) 58-80.
[53] W.T. Tsai, R. Paul, Z. Cao, A. Saimi, B. Xiao, Verification of web services using an enhanced UDDI server, in Proceedings of IEEE WORDS 2003, pp. 131-138.
[54] W.T. Tsai, R. Paul, H. Huang, X. Zhou, X. Wei, Adaptive testing, oracle generation and test case ranking for web services, in: COMPSAC'05: Proceedings of the 29th Annual International Computer Software and Applications Conference, IEEE Computer Society 2005, pp. 101-106.
[55] B. Verheecke, M. A. Cibrán, V. Jonckers, Aspect oriented programming for dynamic web service monitoring and selection, in: ECOWS'04: European Conference on Web Services, 2004, Lecture Notes in Computer Science, 2004, pp. 15-29.

[56] G. Wu, J. Wei, T. Huang, Flexible pattern monitoring for WS-BPEL through stateful aspect extension, in: ICWS'08: Proceedings of the IEEE International Conference on Web Services, 2008, pp. 577-584.
[57] J. Xia, QoS based service composition, in: COMPSAC'06: Proceedings of the 30th Annual International Computer Software and Applications Conference, IEEE Computer Society 2006, pp. 359-361.
[58] S. H. Yoon, D.J. Kim, S.Y. Han, WS-QDL containing static, dynamic and statistical factors of web services quality, in: ICWS'04: Proceedings of the IEEE International Conference on Web Services, 2004, pp. 808-809.
[59] L. Zeng, B. Benatallah, A.H.H. Ngu, M. Dumas, J. Kalagnaman, H. Chang, QoS-aware middleware for web services composition, IEEE Transactions on Software Engineering 30 (5) (2004) 311-327.
[60] International Organization for Standardization. ISO/IEC Standard 9126: Software Engineering - Product Quality, part1. 2001.
[61] G. Canfora, M. Di Penta, Testing and self checking, in: WS-MATE'06: Proceedings of International Workshop on Web Services – Modelling and Testing, 2006, pp. 3-12.
[62] W.T. Tsai, X. Wei, Y. Chen, R. Paul, B. Xiao, Swiss cheese test case generation for web services testing, in IEICE - Transactions on Information and Systems, E88-D, 12 (December 2005), 2691-2698.
[63] T.S. Chow, Testing software design modelled by finite state machines, IEEE Transactions on Software Engineering 4 (1978) 178-187.
[64] D. Dranidis, D. Kourtesis, E. Ramollari, Formal verification of web service behavioural conformance through testing, Annals of Mathematics, Computing & Teleinformatics 1 (5) (2007) 36-43.
[65] W.T. Tsai, Y. Chen, R.A. Paul, N. Liao, H. Huang, Cooperative and group testing in verification of dynamic composite web services, in: Workshop on Quality Assurance and Testing of Web Based Applications, in conjunction with COMPSAC, 2004, pp. 170–173.
[66] OWL-S: Semantic Markup for Web Service. http://www.w3.org/Submission/OWL-S (Accessed April 2010)
[67] Semantic Annotations for WSDL. http://www.w3.org/TR/sawsdl (Accessed April 2010).
[68] OASIS: Web Services Business Process Execution Language (WSBPEL). http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html (Accessed April 2010)
[69] A. Bertolino, G. de Angelis, A. Polini, A QoS test-bed generator for web services, in: ICWE: Proceedings of International Conference on Web Engineering, 2007, pp. 16-20.
[70] A. Bertolino, G.D. Angelis, L. Frantzen, A. Polini, Model-based generation of testbeds for web services, in: TESTCOM/FATES: Testing of Communicating Systems and Formal Approaches to Software Testing, Vol. 5047 in LNCS, Springer, 2008, pp. 266-282.
[71] A. Andrieux, K. Czajkowski, A. Dan, K. Keahey, H. Ludwig, T. Nakata, J. Pruyne, J. Rofrano, S. Tuecke, M. Xu, Web Services Agreement Specification (WS-Agreement). http://www.ogf.org/documents/GFD.107.pdf (Accessed April 2010)
[72] M. Shaw, Writing good software engineering research papers, in: ICSE'03: Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 726-736.

[73] L. Zeng, B. Benatallah, P. Nguyen, A.H.H. Ngu, AgFlow: Agent-based cross-enterprise workflow management system, in: Proceedings of the 27th International Conference on Very Large Data Bases, 2001, pp. 697-698.
[74] L. Baresi, E. D. Nitto, Test and analysis of web services, Springer, Berlin, 2007.
[75] M. Bozkurt, M. Harman, Y. Hassoun, Testing web services: a survey, Technical report TR-10-01, Department of Computer Science, King's College London, 2010.
[76] A.T. Endo, A.S. Simao, A systematic review on formal testing approaches for web services (to appear), in: 4th Brazilian Workshop on Systematic and Automated Software Testing (SAST 2010) - in conjunction with the 22nd IFIP International Conference on Testing Software and Systems (ICTSS'10), 2010
[77] Z. Zakaria, R.Atan, A.A.A. Ghani, N.F.M. Sani, Unit testing approaches for BPEL: a systematic review, in: APSEC: Proceedings of the Asia-Pacific Software Engineering Conference, 2009, pp. 316-322.
[78] M. Oriol, Quality of Service (QoS) in SOA Systems. A Systematic Review. Master Thesis, UPC, Departamento LSI, 2009 [Biblioteca Rector Gabriel Ferraté de la Universitat].
[79] L. Baresi, C. Ghezzi, S. Guinea, Smart monitors for composed services, in: ICSOC'04: Proceedings of the Second International Conference on Service Oriented Computing, 2004, pp. 193–202.
[80] J. Hielscher , R. Kazhamiakin, A. Metzger, M. Pistore, A framework for proactive self-adaptation of service-based application based on online testing, in: 1st European Conference on Towards a Service Based Internet, Vol. 5377 in LNCS, Springer, 2008, pp. 122-133.
[81] K. Mahbub, G, Spanoudakis, Monitoring WS-Agreements: an event calculus based approach, in: Test and Analysis of Web Services, (eds) L. Baresi, E. diNitto, Springer, 2007, pp. 265-306.
[82] M. Di Penta, G. Canfora, G. Espósito, V. Mazza, M. Bruno, Search-based testing of service level agreements, in: GECCO: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM Press 2007, pp. 1090-1097.

Figure Captions:

Table Captions:

Figure 1: SOA architecture. Roles and operations

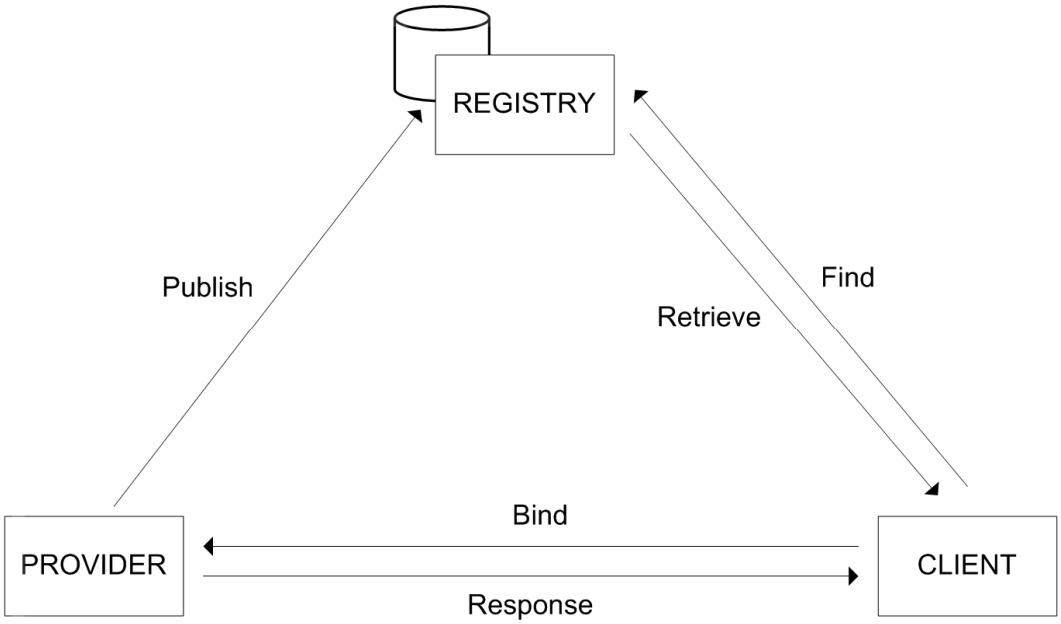

Figure 2: Three-phase search strategy

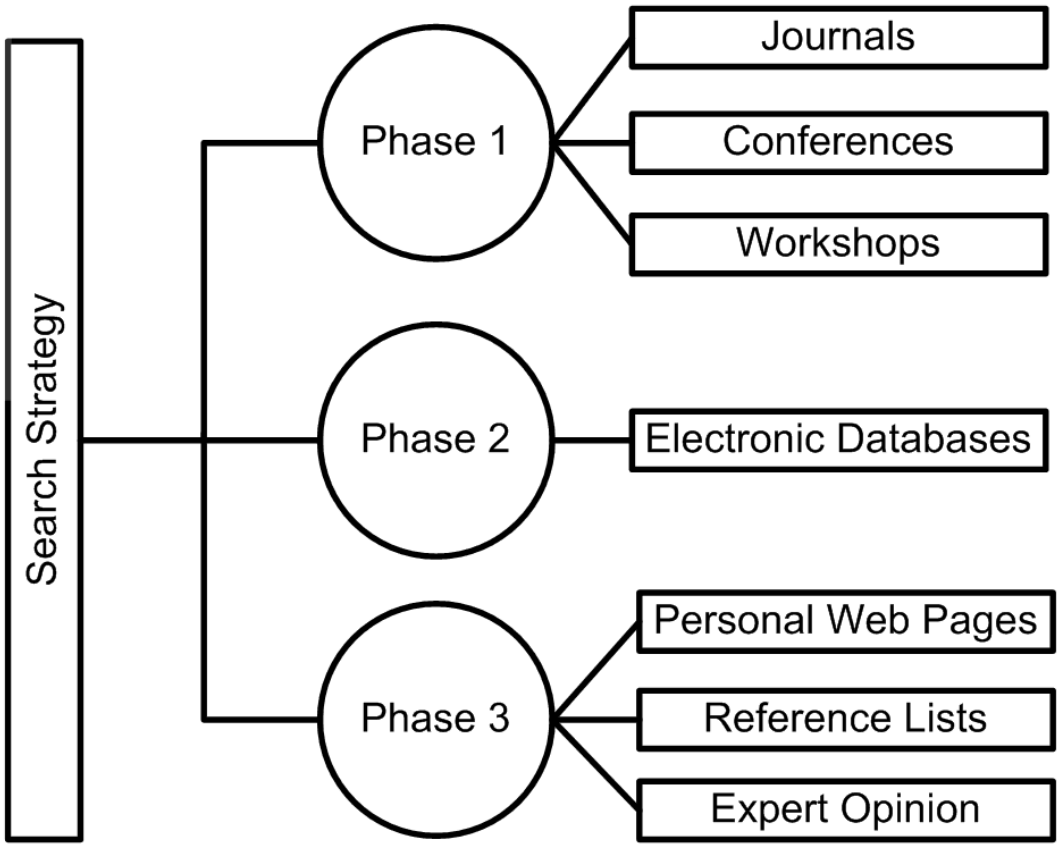

Figure 3: The selection of the primary studies strategy

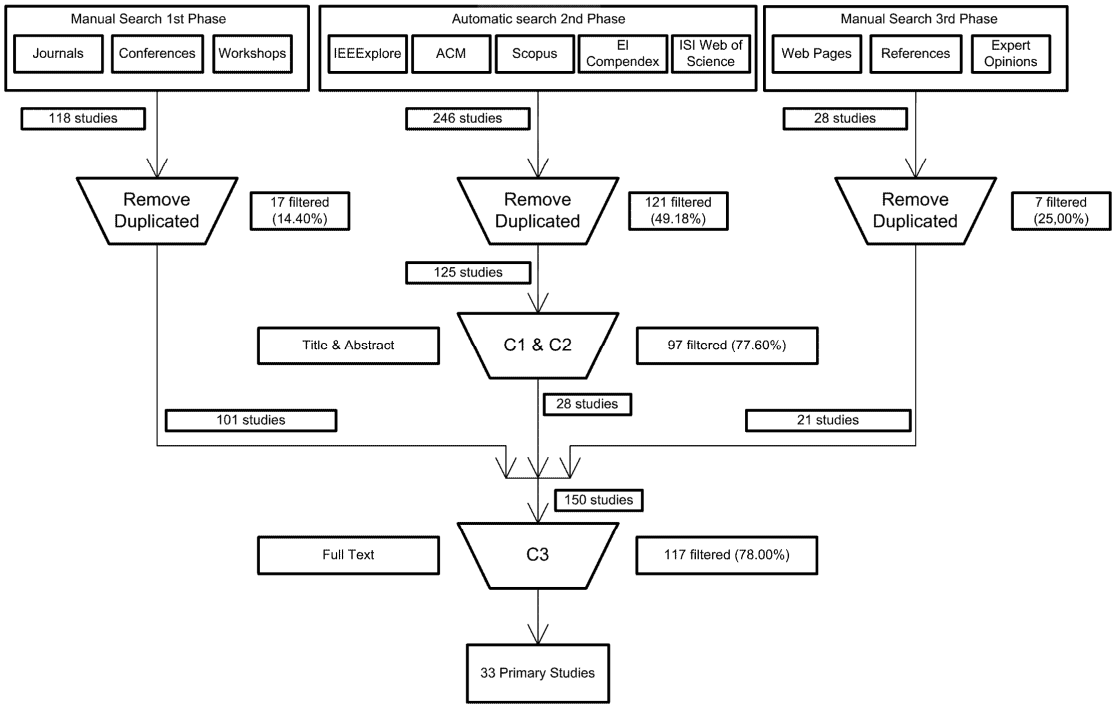

Figure 4: SOA testing architecture

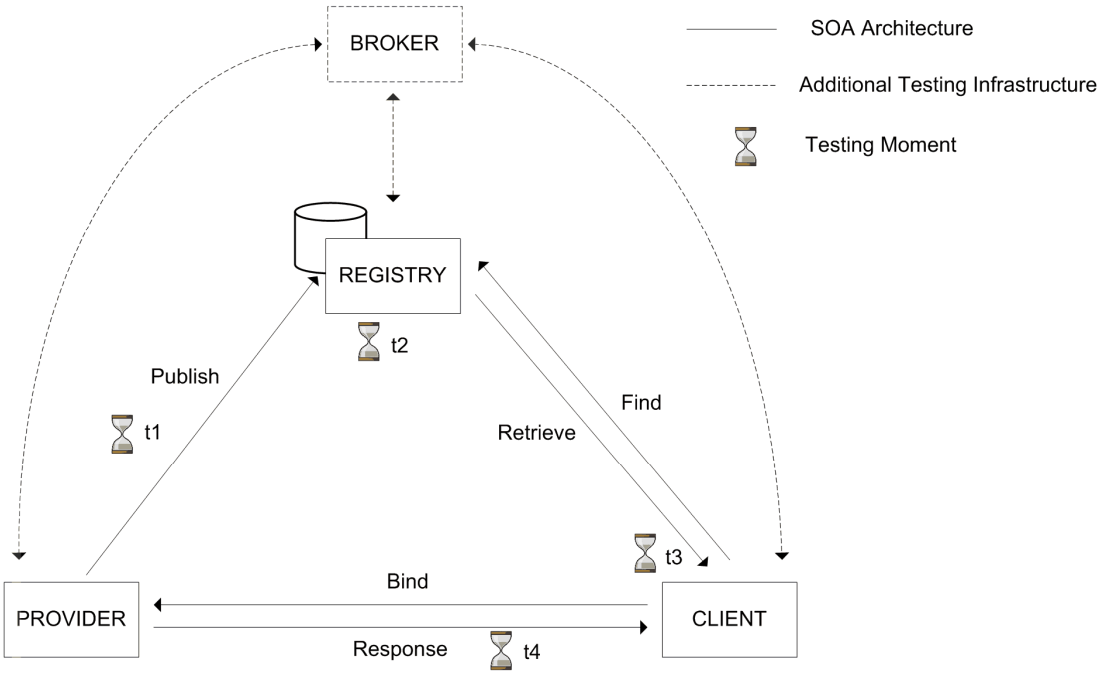

Table 1: Journals, conferences and workshops

	Acronym	Source
Journals	ACM Computing Surveys	ACM Computing Surveys
	ACM TOSEM	ACM Transactions on Software Engineering and Methodology
	ESE	Empirical Software Engineering
	IEEE Computer	IEEE Computer
	IEEE Internet Computing	IEEE Internet Computing
	IEEE Software	IEEE Software
	IEEE TSE	IEEE Transactions on Software Engineering
	IJSEKE	International Journal of Software Engineering and Knowledge Engineering
	JWSR	International Journal of Web Services Research
	IST	Information and Software Technology
	JSS	Journal of Systems and Software
	SIGPLAN Notices	SIGPLAN Notices
	SIGSOFT	ACM SIGSOFT Software Engineering Notes
	SQJ	Software Quality Journal
	STVR	Software Testing, Verification & Reliability
Conferences	ASE	Automated Software Engineering
	CAV	Computer Aided Verification
	COMPSAC	International Computer Software and Applications Conference
	ECOWS	European Conference on Web Services
	EDOC	Enterprise Distributed Object Computing Conference
	ESEM	Empirical Software Engineering and Measurement
	ESEC	European Software Engineering Conference
	ETAPS	European Joint Conference on Theory and Practice of Software
	FORTE	Formal Techniques for Networked and Distributed Systems
	FSE	Foundations of Software Engineering
	ICSE	International Conference on Software Engineering
	ICSOC	International Conference on Service Oriented Computing
	ICST	International Conference on Software Testing, Verification and Validation
	ICWE	International Conference on Web Engineering
	ICWS	International Conference on Web Services
	IEEE Services	IEEE Services
	ISSTA	International Symposium on Software Testing and Analysis
	QSIC	International Conference on Quality Software
	SAC	Symposium on Applied Computing
	SCC	International Conference on Services Computing
	SEKE	Software Engineering and Knowledge Engineering
	TAIC PART	Testing: Academic & Industrial Conference - Practice And Research Techniques
	TAP	International Conference on Tests and Proof
	TestCom	International Conference on Testing Communicating Systems
	WWW	World Wide Web Conference
Workshops	A-MOST	Advances in Model-Based Testing
	AST	Applications of Semantic Technologies
	FATES	International Workshop on Formal Approaches to Testing of Software
	Mutation	International Workshop on Mutation Analysis
	SBST	International Workshop on search based software testing
	SOSE	International Workshop on Service-Oriented Software Engineering / International Workshop on Service-Oriented Systems Engineering
	SPIN	SPIN Workshop on Model Checking of Software
	TAV-WEB	Workshop on Testing, Analysis and Verification of Web Software
	WESOA	International Workshop on Engineering Service Oriented Applications
	WS-Testing	Web Services Testing

Table 2: Electronic databases

Electronic Databases
IEEEXplore [17]
ACM Digital Library [18]
Scopus [19]
El Compendex [20]
ISI Web of Science (WoS) [21]

Table 3: Study Quality Assessment

	Al-Masri and Mahmoud [27]	Bai et al.[28]	Bai et al. [29]	Bai et al. [30]	Balke and Diederich [31]	Baresi et al. [32]	Ben Halima et al. [33]	Bertolino and Polini [34]	Bianculli et al. [35]	Canfora et al. [36]	Ernst and Lencevicious [37]	Erradi et al. [38]	Gorbeko et al.[39]	Jurca et al.[40]	Kourtesis et al. [41]	Liu et al. [42]	Lohmann et al. [43]	Mendonça et al. [44]	Moser et al. [45]	Mosincat and Binder [46]	Oriol et al. [47]	Park et al. [48]	Ran [49]	Seo et al. [50]	Syzdlo and Zielinski [51]	Tosi et al. [52]	Tsai et al.[53]	Tsai et al.[54]	Verheecke et al. [55]	Wu et al. [56]	Xia [57]	Yoon et al. [58]	Zeng et al. [59]
QA1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QA2	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QA3	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓
QA4	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓
QA5	✓	X	✓	✓	✓	✓	✓	X	✓	✓	✓	✓	X	✓	X	✓	✓	✓	✓	✓	X	✓	X	X	✓	✓	X	✓	✓	✓	✓	X	X	✓
QA6	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
QA7	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	X	✓	✓	X	✓	✓	✓	✓	✓	✓	✓	✓	X	X	✓	✓	X	✓	✓	✓

Table 4: Data Extraction Form

ID	Field	Description	Research Question
	Internal Information		
1	Identifier	Unique Identifier for the primary study	
2	Reviewer	Name of the researcher who reviews the study	
3	Date	Date of the data extraction	
	Reference Information		
4	Title	Title of the study	
5	Authors	Authors of the study	
6	Year	Publication Year	RQ7
7	Type of study	Where has the study been published? (Options: Journal Article, Conference/Workshop paper, Book chapter…)	RQ8
8	Name of the journal/conf./works./book	Name of the journal, conference, workshop, book... where the study has been published	RQ8
9	Reference	Rest of the reference information: Volume, Number, Pages…	
	Content Information		
10	Abstract	Abstract of the study (copied verbatim)	
11	Objectives	What are the objectives of the study?	RQ1
12	System Under Test	What is the system which is going to be tested?	
13	Technologies & Standards	Which are technologies and standards being researched?	RQ5
14	Testing Technique	Which is the testing technique applied to the system?	RQ2
15	Test Executer	Who executes the tests? (Options: Provider, Client, Registry, Third Party Certifier, Other)	RQ4
16	Execution Moment	When are the tests executed? (Options: Before the registration, While the services are published, Just before the binding, During the execution...)	RQ3
17	Validation Method	Which is method is used to validate the study?	RQ6
18	Examples	If examples are used to validate the study, list of the examples	
19	Conclusions	Conclusions of the article (copied verbatim)	
20	Additional Notes	Space to write additional notes about the study	

Table 5: Primary studies

Reference	Description
Al-Masri and Mahmoud [27]	Broker to test public web services and gather QoS attributes values
Bai et al. [28]	Framework to generate test cases for the services published in the UDDI registry
Bai et al. [29]	Framework to perform dynamic reconfigurable testing of SOA
Bai et al. [30]	Framework to test services before their publication in a registry and monitor the executions
Balke and Diederich. [31]	Algorithm to perform monitoring of the executions and replacement of web services
Baresi et al. [32]	Framework to monitor BPEL processes and perform self-healing actions
Ben Halima et al. [33]	Framework to monitor QoS features and perform self-healing actions
Bertolino and Polini [34]	Framework to test web services interoperability
Bianculli et al. [35]	Infrastructure to perform feedback based monitoring and pro-active service selection
Canfora et al. [36]	Framework to monitor QoS features to bind or rebind web services in a composition
Ernst and Lencevicious [37]	Algorithm to test web services detecting substitutability and composability
Erradi et al. [38]	Middleware to monitor web services and execute adaptive strategies
Gorbenko et al. [39]	Extension of UDDI registry with monitoring capabilities to publish dependability parameters
Jurca et al. [40]	Mechanism to perform feedback based monitoring with economic incentives of web services
Kourtesis et al. [41]	Mechanism to detect mismatches between specification and implementation in the registry
Liu et al. [42]	Framework to monitor web services execution and receive client's feedback
Lohmann et al. [43]	Extension of UDDI registry to generate test cases using graph transformation rules
Mendonça et al. [44]	Empirical assessment to evaluate five services selection criteria
Moser et al. [45]	System to monitor BPEL processes and perform the replacement of services
Mosincat and Binder [46]	Infrastructure to enhance BPEL processes with self-tuning behaviour
Oriol et al. [47]	SOA system to monitor QoS features in order to detect SLA violations
Park et al. [48]	Approach to perform black-box testing based discovery of web services
Ran [49]	Extension of UDDI registry to test the provider's QoS claims about the services
Seo et al. [50]	Architecture to monitor QoS in order to select a service
Szydlo and Zielinski [51]	Mechanism to monitor QoS features and detect SLA violations
Tosi et al. [52]	Methodology to generate test cases for web services and perform adaptation strategies
Tsai et al. [53]	Extension of UDDI registry to generate and execute test cases performing check-in and check-out mechanisms
Tsai et al. [54]	Technique to generate test cases, create the oracle and rank web services and test cases
Verheecke et al. [55]	Mechanism to perform aspect based monitoring and selection of web services
Wu et al. [56]	Extension to BPEL to support aspect based monitoring of BPEL processes and execution of adaptive actions
Xia [57]	Architecture to test QoS features in order to select services to be composed
Yoon et al. [58]	Extension of UDDI registry with QoS testing capabilities
Zeng et al. [59]	Middleware to test QoS features and select services to be composed

Table 6: Distribution of studies per testing objective

	Objective																				SUT	
	Fault Detection										Decision Making											
		Non-functional										Non-functional										
	Functional	Accuracy	Availability	Cost	Reliability	Reputation	Response Time	Successful Exec. Rate	Throughput	Not Specified	**Functional**	Accuracy	Availability	Cost	Reliability	Reputation	Response Time	Successful Exec. Rate	Throughput	Not Specified	**Atomic Service**	**Service Composition**
Al-Masri and Mahmoud [27]													X		X		X	X	X		X	
Bai et al. [28]	X																				X	X
Bai et al. [29]	X																				X	X
Bai et al. [30]	X									X											X	X
Balke and Diederich. [31]			X	X																		X
Baresi et al. [32]	X									X												X
Ben Halima et al. [33]			X				X		X												X	X
Bertolino and Polini [34]	X																				X	
Bianculli et al. [35]											X				X		X		X			X
Canfora et al. [36]										X												X
Ernst and Lencevicious [37]											X										X	
Erradi et al. [38]	X		X		X		X															X
Gorbenko et al. [39]													X		X		X				X	
Jurca et al. [40]																				X	X	
Kourtesis et al. [41]	X																				X	
Liu et al. [42]														X		X	X				X	
Lohmann et al. [43]	X																				X	
Mendonça et al. [44]																	X				X	
Moser et al. [45]		X	X				X														X	X
Mosincat and Binder [46]																	X		X		X	X
Oriol et al. [47]			X				X														X	
Park et al. [48]											X		X	X		X	X	X			X	
Ran [49]										X											X	
Seo et al. [50]													X	X	X		X		X		X	
Szydlo and Zielinski [51]										X												X
Tosi et al. [52]	X									X											X	
Tsai et al. [53]	X									X											X	
Tsai et al. [54]	X																				X	X
Verheecke et al. [55]																				X	X	X
Wu et al. [56]	X																					X
Xia [57]													X	X			X	X	X		X	
Yoon et al. [58]													X		X	X	X				X	
Zeng et al. [59]													X	X		X	X	X			X	X
Total	**12**	1	5	1	1	0	4	0	1	7	**3**	0	7	5	5	4	11	4	5	2	**26**	**16**
		12										**13**										
		19										**14**										

Table 7: Distribution of studies per testing technique / method

	Testing Technique / Method							
	Monitoring		**Test Case Generation**				**Group Testing**	**Not Specified**
	Online	**Offline**	**Partition Testing**	**Swiss Cheese**	**Stream X-mach. Testing Method**	**Not Specified**		
Al-Masri and Mahmoud [27]								X
Bai et al. [28]						X	X	
Bai et al. [29]						X		
Bai et al. [30]	X							
Balke and Diederich. [31]	X							
Baresi et al. [32]	X							
Ben Halima et al. [33]	X							
Bertolino and Polini [34]						X		
Bianculli et al. [35]	X							
Canfora et al. [36]	X							
Ernst and Lencevicious [37]		X						
Erradi et al. [38]	X							
Gorbenko et al. [39]	X	X						
Jurca et al. [40]		X						
Kourtesis et al. [41]					X			
Liu et al. [42]	X	X						
Lohmann et al. [43]	X		X					
Mendonça et al. [44]		X						
Moser et al. [45]	X							
Mosincat and Binder [46]	X							
Oriol et al. [47]	X							
Park et al. [48]			X					
Ran [49]								X
Seo et al. [50]	X							
Szydlo and Zielinski [51]	X							
Tosi et al. [52]	X					X		
Tsai et al. [53]						X		
Tsai et al. [54]				X			X	
Verheecke et al. [55]	X							
Wu et al. [56]	X							
Xia [57]		X				X		
Yoon et al. [58]								X
Zeng et al. [59]		X						
Total	**18**	**7**	**2**	**1**	**1**	**6**	**2**	**3**
	23		**10**					

Table 8: Distribution of studies per stakeholders and points in time

	Stakeholders				**Points in time**			
	Provider	Registry	Third Party / Broker	Client	Before the publication (t1)	While services are published (t2)	Just before the binding (t3)	During the execution (t4)
Al-Masri and Mahmoud [27]		X			X	X		
Bai et al. [28]		X			X	X		
Bai et al. [29]			X	X		X		
Bai et al. [30]	X	X		X	X	X		X
Balke and Diederich. [31]			X	X				X
Baresi et al. [32]				X				X
Ben Halima et al. [33]	X		X	X				X
Bertolino and Polini [34]		X			X			
Bianculli et al. [35]		X		X				X
Canfora et al. [36]				X			X	X
Ernst and Lencevicious [37]				X			X	X
Erradi et al. [38]				X				X
Gorbenko et al. [39]		X	X	X		X		X
Jurca et al. [40]			X	X				X
Kourtesis et al. [41]		X	X	X	X		X	
Liu et al. [42]	X	X		X				X
Lohmann et al. [43]		X			X			X
Mendonça et al. [44]				X			X	X
Moser et al. [45]				X				X
Mosincat and Binder [46]				X			X	X
Oriol et al. [47]				X				X
Park et al. [48]			X	X			X	
Ran [49]			X		X			
Seo et al. [50]			X			X		X
Szydlo and Zielinski [51]				X				X
Tosi et al. [52]				X			X	X
Tsai et al. [53]	X	X		X	X		X	
Tsai et al. [54]		X		X	X	X	X	
Verheecke et al. [55]				X				X
Wu et al. [56]				X				X
Xia [57]			X				X	X
Yoon et al. [58]		X					X	
Zeng et al. [59]			X					X
Total	**4**	**12**	**11**	**24**	**9**	**7**	**11**	**23**

Table 9: Distribution of studies combining participants and points in time

Stakeholder / Point in time	Before the registration (t1)	While services are published (t2)	Just before the binding (t3)	During the execution (t4)
Provider	[53]	-	-	[30][33][42]
Registry	[27][28][30][34] [41][43][53][54]	[27][28][30][39][54]	[58]	[30][35][42][43]
Third Party / Broker	[41][49]	[29][39][50]	[48][57]	[31][33][40][50][57][59]
Client	-	[29][39]	[36][37][41][44] [46][48][52][53][54]	[30][31][32][33][35][36][37][38] [39][40][42][44][45][46] [47][51][52][55][56]

Table 10: Distribution of studies per technology / standard

	Atomic Service Description			Service Composition Behaviour		Registry Specification		Service Level Agreement (SLA)			
	WSDL	OWL-S	SAWSDL	BPEL	OWL-S	UDDI	Not Specified	WSLA	WS-Agreement	WS-Policy	Not Specified
Al-Masri and Mahmoud [27]						X					
Bai et al. [28]						X					
Bai et al. [29]	X	X		X		X					
Bai et al. [30]					X		X				
Balke and Diederich. [31]											
Baresi et al. [32]				X							
Ben Halima et al. [33]	X			X							
Bertolino and Polini [34]						X					
Bianculli et al. [35]				X		X					
Canfora et al. [36]											X
Ernst and Lencevicious [37]											
Erradi et al. [38]				X						X	
Gorbenko et al. [39]	X					X					
Jurca et al. [40]						X		X			
Kourtesis et al. [41]			X				X				
Liu et al. [42]							X				
Lohmann et al. [43]							X				
Mendonça et al. [44]						X					
Moser et al. [45]	X			X							
Mosincat and Binder [46]				X							
Oriol et al. [47]											X
Park et al. [48]	X										
Ran [49]						X					
Seo et al. [50]						X		X			
Szydlo and Zielinski [51]											X
Tosi et al. [52]	X										
Tsai et al. [53]	X										
Tsai et al. [54]		X					X				
Verheecke et al. [55]											
Wu et al. [56]				X							
Xia [57]						X					X
Yoon et al. [58]						X					
Zeng et al. [59]						X					
Total	**7**	**2**	**1**	**8**	**1**	**13**	**5**	**2**	**0**	**1**	**4**

Table 11: Distribution of studies per validation method

	Validation Method					Type of examples				
	Analysis	Experience	Evaluation	Example	Persuasion / Opinion	Ad-hoc	Existing Example	From Standards	Public Services	Real Scenarios
Al-Masri and Mahmoud [27]			X						X	
Bai et al. [28]					X					
Bai et al. [29]				X		X				
Bai et al. [30]		X								X
Balke and Diederich. [31]			X			X				
Baresi et al. [32]			X			X				
Ben Halima et al. [33]		X								X
Bertolino and Polini [34]					X					
Bianculli et al. [35]			X				X	X		
Canfora et al. [36]	X					X				
Ernst and Lencevicious [37]			X						X	
Erradi et al. [38]			X			X				
Gorbenko et al. [39]					X					
Jurca et al. [40]				X		X				
Kourtesis et al. [41]				X		X				
Liu et al. [42]			X			X				
Lohmann et al. [43]				X					X	
Mendonça et al. [44]	X								X	
Moser et al. [45]			X			X				
Mosincat and Binder [46]			X			X				
Oriol et al. [47]					X					
Park et al. [48]			X						X	
Ran [49]					X					
Seo et al. [50]					X					
Szydlo and Zielinski [51]			X			X				
Tosi et al. [52]				X		X			X	
Tsai et al. [53]					X					
Tsai et al. [54]			X			X				
Verheecke et al. [55]				X		X				
Wu et al. [56]			X			X				
Xia [57]					X					
Yoon et al. [58]					X					
Zeng et al. [59]			X				X			
Total	**2**	**2**	**14**	**6**	**9**	**15**	**2**	**1**	**6**	**2**

Table 12: Distribution of studies per year and source

Primary Study	Year	Type	Source
Al-Masri and Mahmoud [27]	2008	Journal Article	IT Professional
Bai et al. [28]	2007	Conference Paper	COMPSAC
Bai et al. [29]	2007	Conference Paper	COMPSAC
Bai et al. [30]	2008	Conference Paper	COMPSAC
Balke and Diederich. [31]	2006	Conference Paper	ICWS
Baresi et al. [32]	2007	Workshop Paper	ESSPE
Ben Halima et al. [33]	2008	Conference Paper	ICWS
Bertolino and Polini [34]	2005	Conference Paper	Euromicro SSAEW
Bianculli et al. [35]	2008	Conference Paper	ICWS
Canfora et al. [36]	2008	Journal Article	JSS
Ernst and Lencevicious [37]	2006	Workshop Paper	WS-MATE
Erradi et al. [38]	2007	Conference Paper	ECOWS
Gorbenko et al. [39]	2008	Conference Paper	COMPSAC
Jurca et al. [40]	2007	Conference Paper	WWW
Kourtesis et al. [41]	2008	Book Chapter	Pervasive Collaborative Networks
Liu et al. [42]	2004	Conference Paper	WWW
Lohmann et al. [43]	2007	Book Chapter	Test and Analysis of Web Services
Mendonça et al. [44]	2008	Journal Article	JSS
Moser et al. [45]	2008	Conference Paper	WWW
Mosincat and Binder [46]	2009	Conference Paper	SOCA
Oriol et al. [47]	2008	Workshop Paper	MONA+
Park et al. [48]	2009	Conference Paper	COMPSAC
Ran [49]	2003	Journal Article	ACM SIGecom Exchanges
Seo et al. [50]	2004	Conference Paper	ICESS
Szydlo and Zielinski [51]	2008	Conference Paper	ICCS
Tosi et al. [52]	2009	Journal Article	I. J. of Autonomic Computing
Tsai et al. [53]	2003	Workshop Paper	WORDS
Tsai et al. [54]	2005	Conference Paper	COMPSAC
Verheecke et al. [55]	2004	Conference Paper	ECOWS
Wu et al. [56]	2008	Conference Paper	ICWS
Xia [57]	2006	Conference Paper	COMPSAC
Yoon et al. [58]	2004	Conference Paper	ICWS
Zeng et al. [59]	2004	Journal Article	IEEE TSE

Table 13: Distribution of studies per year

Publication Year	Number of primary studies	%
2003	2	6.06
2004	5	15.15
2005	2	6.06
2006	3	9.09
2007	6	18.18
2008	12	36.36
2009	3	9.09

Table 14: Distribution of studies per source

Type of publication	Number of primary studies	%
Journal Article	6	18.18
Conference Paper	21	63.63
Workshop Paper	4	12.12
Book Chapter	2	6.06