Author Biographies

Birgitta König-Ries holds the Heinz-Nixdorf Endowed Chair for Practical Computer Science at the University of Jena, Germany. Prior to this she has worked with the Technical University of Munich, Florida International University, the University of Louisiana at Lafayette, and the University of Karlsruhe. Birgitta holds both a diploma and a PhD from the latter. Her research is focused on the transparent integration of both information and functionality. In particular, her group is working on semantic web services and on portal technology.

Ulrich Küster holds a diploma in computer science from Friedrich-Schiller-University Jena, Germany. Since 2005, he is working as a researcher in the group of Prof. König-Ries at that university. His main research interests are on semantic web services, service discovery and matchmaking languages and frameworks and evaluation of semantic service technology.

Dr. habil. Matthias Klusch is Senior Researcher and Research Fellow of the German Research Center for Artificial Intelligence (DFKI) where he is co-head of the Multiagent Systems (MAS) group and its research team on Intelligent Information Systems and Agents (I2S). He is also Adjunct Professor of Computer Science at Swinburne University of Technology in the Center for Complex Software Systems and Services. He received both his MSc (1992) and PhD (1997) in computer science from the Christian-Albrecht University of Kiel, and his Habilitation (2009) in computer science from the University of the Saarland, Germany. He published widely and serves as editorial member of several major journals on the Semantic Web, service-oriented computing, multi-agent systems, and distributed rational decision-making. He is member of the GI, IEEE, and ACM.
Full biography available at: http://www.dfki.de/~klusch/CV-Klusch.pdf

Evaluating Semantic Web Service Technologies: Criteria, Approaches and Challenges

Ulrich Küster and Birgitta König-Ries

Institute of Computer Science,
Friedrich-Schiller-University Jena
Ernst-Abbe-Platz 2-4
D-07743 Jena, Germany
Ulrich.Kuester|Birgitta-Koenig-Ries@uni-jena.de

Matthias Klusch

German Research Centre for Artificial Intelligence
Stuhlsalzenhausweg 3
D-66121 Saarbruecken, Germany
klusch@dfki.de

Abstract

In recent years a huge amount of research effort and funding has been devoted to the area of semantic web services (SWS). This has resulted in the proposal of numerous competing approaches to facilitate the automation of discovery, composition and mediation for web services using semantic annotations. However, despite of a wealth of theoretical work, too little effort has been spent towards the comparative experimental evaluation of the competing approaches so far. Progress in scientific development and industrial adoption is thereby hindered. An established evaluation methodology and standard benchmarks that allow the comparative evaluation of different frameworks are thus needed for the further advancement of the field. To this end, a criteria model for SWS evaluation is presented and the existing approaches towards SWS evaluation are comprehensively analyzed. Their shortcomings are discussed in order to identify the fundamental issues of SWS evaluation. Based on this discussion, a research agenda towards agreed upon evaluation methodologies is proposed.

INTRODUCTION

To foster reuse, state of the art software engineering has been driven over decades by the trend towards more and more component based software development. In recent years another trend towards more and more distributed and more loosely coupled systems could be observed. Service oriented architectures (*SOAs*) are the latest product of this long-reaching development. Web services in particular have become increasingly popular and are currently the most prominent implementation of a SOA. The grand vision of the web service paradigm is to have a rich library of ten thousands web services available online that provide access to information, functionality or resources of any kind and that can be easily integrated into existing applications or composed in a workflow-like fashion to form new applications.

Even though this promising technology has already proven to be an effective way of creating widely distributed and loosely coupled systems, the integration of the services is still labor intensive and thus expensive work. Thus – following the vision of the semantic web (Berners-Lee et al., 2001) – the idea of semantic web services (*SWS* in the following) was introduced (McIlraith et al., 2001), applying the principles of the semantic web to the web service paradigm.

SWS related research has attracted a huge amount of effort and funding recently. Within the sixth EU framework program[1] alone, for instance, at least 20 projects with a combined funding of more than 70 million Euros dealt directly with semantic services. This gives a good impression of the importance being put on this field of research. The huge amount of effort (and money) spent into SWS research has resulted in numerous proposals of ontology based semantic descriptions for component services (Klusch, 2008b). Based on such descriptions, a plethora of increasingly sophisticated techniques and algorithms for the automated or semi-automated dynamic discovery, composition, binding, and invocation of services have been proposed (Klusch, 2008a).

However, despite of this wealth of theoretical work, recent surveys have shown that surprisingly little effort has been spent towards the comparative evaluation of the competing approaches (Küster et al., 2007b, Klusch and Zhing, 2008). Until recently there were no comparative evaluations and it was impossible to find two systems which had been evaluated on the same use cases. Evaluations were mostly concentrated either on artificially synthesized datasets under questionable assumptions or based on one or two use cases for which it was not clear, whether they were reverse engineered from the solution. In other words: ”There are many claims for such technologies in academic workshops and conferences. However, there is no scientific method of comparing the actual functionalities claimed. […] Progress in scientific development and in industrial adoption is thereby hindered” (Lausen et al., 2007).

There are striking parallels to this situation in the history of related areas:

> ”[in the experiments] …there have been two missing elements. First […] there has been no concerted effort by groups to work with the same data, use the same evaluation techniques, and generally compare results across systems. The importance of this is not to show any system to be superior, but to allow comparison across a very wide variety of techniques, much wider than only one research group would tackle. […] The second missing element, which has become critical […] is the lack of a realistically-sized test collection. Evaluation using the small collections currently available may not reflect performance of systems in large […] and certainly does not demonstrate any proven abilities of these systems to operate in real-world […] environments. This is a major barrier to the transfer of these laboratory systems into the commercial world.”

This quote by Donna Harman (Harman, 1992) addressed the situation in text retrieval research prior to the establishment of the series of TREC conferences[2] in 1992 but seems to perfectly describe the current situation in SWS research. Harman continued:

> ”The overall goal of the Text REtrieval Conference (TREC) was to address these two missing elements. It is hoped that by providing a very large test collection and encouraging interaction with other groups in a friendly evaluation forum, a new thrust in information retrieval will occur.”

From the perspective of today, it is clear that her hope regarding the positive influence of the availability of mature evaluation methods to the progress of information retrieval research was well justified. This corresponds to a finding of Sim and colleagues who have developed a general theory of benchmarking (Sim et al., 2003). They observe that the creation and widespread use of a benchmark within a research area is frequently accompanied by rapid technical progress and community building:

”Creating a benchmark requires a community to examine their understanding of the field, come to an agreement on what are the key problems, and encapsulate this knowledge in an evaluation. Using the benchmark results in a more rigorous examination of research contributions, and an overall improvement in the tools and techniques being developed. Throughout the benchmarking process, there is greater communication and collaboration among different researchers leading to a stronger consensus on the community’s research goals.” (Sim et al., 2003)

We follow these lines and argue that today in the area of SWS related research an established evaluation methodology and standard benchmarks that allow the comparative evaluation of different frameworks are needed for the advancement of the field.

The development of such benchmarks requires answers to the fundamental research questions related to the evaluation of SWS technology: What are the appropriate criteria for evaluation? How can various fundamentally different SWS approaches be compared effectively? How can such comparison be guaranteed to be unbiased and balanced? Generally, how can the relative advantage of some SWS technology over another one, and ultimately over existing conventional programming techniques be reproducibly proven or disproven?

Without the ability to perform verifiable comparisons among different SWS technologies and of SWS technology with other software engineering techniques, SWS will remain an art, but not become a science. However, only if we succeed in transforming SWS research from art to science, industrial adoption and widespread recognition of research results will become reality. The development of commonly agreed upon evaluation methodologies and standard benchmark suites is thus absolutely indispensable.

The authors of this article have worked on establishing successful international SWS evaluation campaigns for about three years now. In the course of these activities we have learned that today, there is neither a consensus on what to evaluate nor on how to evaluate. Furthermore, the development of objective and reliable evaluation methodologies is generally far more complex than anticipated.

This article attempts to step back and discuss the general issues related to evaluating SWS technology. The approach is to learn from the existing efforts. Their current shortcomings and pitfalls are analyzed in order to develop an understanding for the general scientific problems related to SWS evaluation. Based on this analysis, a proposal of a further research agenda for SWS evaluation is laid out.

The rest of the paper is organized as follows. In the following section, an answer to the question of *what* to evaluate is presented. A comprehensive, general model of the suitable criteria for evaluation is derived from a requirement analysis for SWS. The remainder of the paper is devoted to the question of *how* to evaluate. First, the existing efforts in the area of SWS evaluation will be introduced and related to each other according to the presented criteria model. This will be followed by an in-depth analysis of the metrics and measures used to evaluate SWS. Current shortcomings are examined to identify and discuss the underlying research problems that need to be solved. Finally, the paper concludes by proposing a research agenda towards the development of standard evaluation methodologies in SWS research.

SWS EVALUATION CRITERIA

The first important question related to any evaluation endeavour regards the criteria according to which the object of interest should be evaluated, i.e. *what* to evaluate. As will be shown in the following Section, different SWS evaluation initiatives have so far focused on very different criteria and there has not been a discussion on why those criteria have been chosen or how they relate to each other. To establish a comprehensive, general model of the suitable criteria for evaluation, we follow the well established Goal-Question-Metric (GQM) approach to software evaluation. The GQM paradigm is a mechanism for defining and evaluating a set of operational goals, using measurement. It has been developed in 1984 at NASA, been used in various software engineering projects worldwide and is a recommended gold practice of the US Department of Defense Information Analysis Center[3] (Basili, 1992).

GQM is based on the assumption that the evaluation of any system should be an evaluation of fitness for purpose. Thus, any evaluation activity should be preceded by the identification of the engineering goals behind the system or technology to be evaluated. The goals are defined in an operational, tractable way by refining them into a set of quantifiable questions. These questions, in turn, are then used to define a specific set of metrics and identify the necessary data to measure according to those metrics (Basili, 1992).

The obvious overall goal of SWS is to support or (partially) automate the process of consuming functionality offered as a service. However, the precise use case motivating particular approaches to SWS is often not clearly identified. To identify the main objectives motivating SWS, we performed a review of published work with detailed and specific descriptions of envisioned use-cases(Sîrbu et al., 2006, Cobo et al., 2004, Preist et al., 2005, Friesen and Namiri, 2006, Preist, 2007, Küster et al., 2007a, Preist, 2004, Stollberg, 2004, Küster and König-Ries, 2007, Li and Horrocks, 2003, Ragone et al., 2007, Colucci et al., 2005, Balzer et al., 2004, Friesen and Grimm, 2005, Gugliotta et al., 2006, Klein et al., 2005). While this review is clearly not exhaustive, we believe that it is representative for the majority of SWS projects. We found that the published use cases can be roughly divided in two types of application domains.

The first type envisions to enable late dynamic service discovery, selection and binding at run-time. In mobile environments, the non-availability of stable services forces to discover and bind services dynamically, e.g. booking local attractions via mobile devices while travelling (Sîrbu et al., 2006, Klein et al., 2005). In B2B scenarios, the dynamic and autonomous reaction to changes in the service landscape allows taking advantage of the appearance of better or less expensive services or recovering from failures by automatically replacing faulted or offline services (Cobo et al., 2004). Many scenarios involve the dynamic selection of service instances based on similar re-appearing goal instances in B2B relationships: the location of suitable carriers to provide transportation services (Preist et al., 2005, Friesen and Namiri, 2006, Küster et al., 2007a), an intelligent procurement management for non-critical supplies (Preist, 2007, Küster et al., 2007a) or the location of the most appropriate notification service to contact a customer (Cobo et al., 2004). In B2C scenarios, SWS are often motivated by the desire to delegate a search for the best among many options to autonomous agents. In these scenarios, many providers offer similar services and the best provider depends on the concrete goal or varies over time. Typical scenarios of this type involve the discovery of the best deal to purchase a set of items, e.g. books (Preist, 2004), furniture (Stollberg, 2004), computers (Küster and König-Ries, 2007, Li and Horrocks, 2003), or used cars (Ragone et al., 2007), to find the best matching offer in an apartment rental scenario (Colucci et al., 2005), or to make travel arrangements and flight or hotel bookings (Balzer et al., 2004).

The focus in all of the above mentioned contexts is on discovery, matchmaking and precise filtering or ranking of many possible options. Usually a high degree of automation is sought, in some scenarios complete automation is required.

The second type of application scenarios deals with supporting developers in establishing or maintaining rather stable B2B or B2C relationships and setting up distributed applications. Such scenarios root in application domains like Business Process Management (BPM) and Enterprise Application Integration (EAI). In these fields, SWS are motivated by the desire to decrease the programming time and cost by semi-automating very time consuming tasks like the establishment of data and process mediation procedures. Scenarios in this category include the provision of value added services by bundling or mediating external contractors (Friesen and Grimm, 2005), the semi-automated design of processes to manage virtual ISP problems (Preist, 2004), or the development of an emergency management system in the e-government domain (Gugliotta et al., 2006).

The main focus in these scenarios is on mediation and composition rather than discovery. The goal of employing SWS in such settings is to ease the process of integrating remote systems, master the encountered heterogeneity, and decrease the level of coupling between the components. Full automation is usually not required.

In this work, we focus on the first class of application scenarios, leaving the other class to be dealt with in future work. We traced back the use cases of that first class of application scenarios to three main high-level goals of SWS. Following the GQM approach these are defined in a tractable way by refining them into a set of quantifiable questions. The goals and the defining questions are described in the following, referenced with the use cases from which they have been derived.

Goal 1 *Allow the dynamic and transparent usage of functionality in mobile or P2P environments where the availability and reliability of that desired functionality is not under local control (Sîrbu et al., 2006, Klein et al., 2005).*

1. Does the framework allow use of external functionality as if it were locally available? Is the framework able to hide the fact that the functionality is dynamically discovered and bound and supports full automation?
2. Does the framework guarantee correctness to allow for full automation?
3. If required, does the framework work under the requirements of P2P environments or the limited resources of mobile devices?

Goal 2 *Minimize the cost or optimize the quality of a consumed functionality by dynamically reacting to changes in the service landscape (Cobo et al., 2004, Preist et al., 2005, Friesen and Namiri, 2006, Preist, 2007, Küster et al., 2007a, Preist, 2004, Stollberg, 2004, Küster and König-Ries, 2007, Li and Horrocks, 2003, Ragone et al., 2007, Colucci et al., 2005, Balzer et al., 2004, Friesen and Grimm, 2005, Klein et al., 2005).*

4. Does the usage of the framework decrease the time necessary to find a good enough or the optimal option? To what extent?
5. Does the usage of the framework increase the quality of the option discovered? To what extent?

Goal 3 *Reduce failures or down-time by automatically replacing faulted or unavailable service components in a distributed application (Cobo et al., 2004, Preist et al., 2005, Preist, 2007, Küster et al., 2007a, Preist, 2004, Friesen and Grimm, 2005, Gugliotta et al., 2006).*

6. Does the framework support to react autonomously to detected failures?
7. If a human still needs to be in the loop, to what extent does the framework support that human and reduces the time necessary to recover from failures?

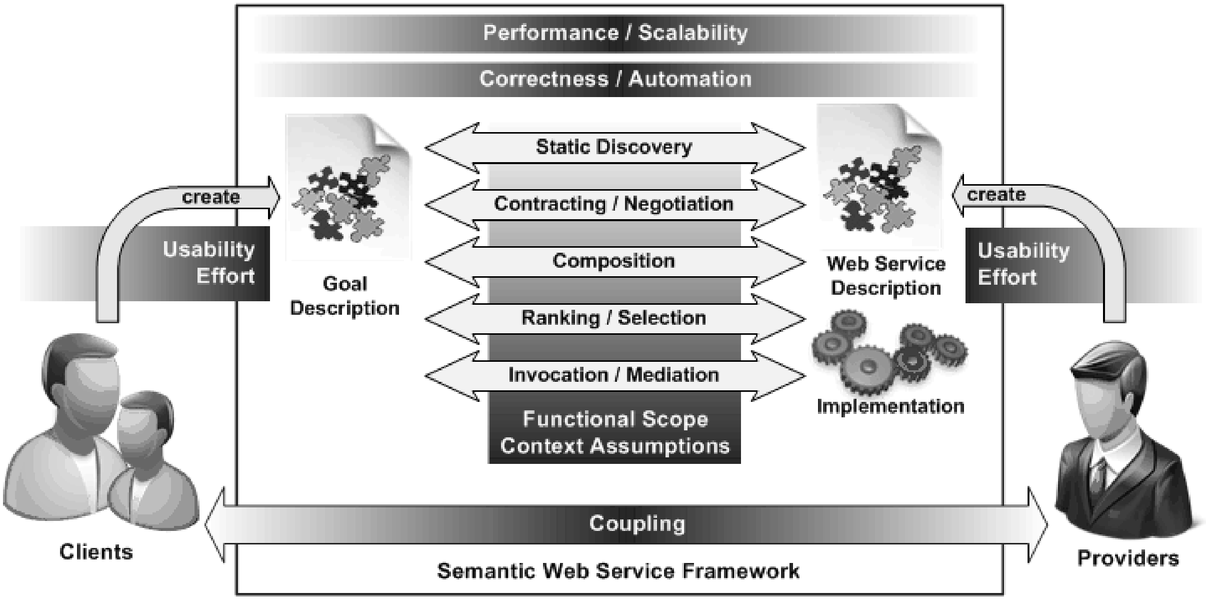

Figure 1: A model of the dimensions of evaluation in the field of automating service consumption using SWS frameworks

Furthermore, there are a number of questions related to all three goals:

8. How tightly coupled are service providers and consumers in the framework?
9. How much effort is it to use the framework, e.g. to publish service offers or formalize goals with the framework?
10. How much effort is it to set up and maintain the framework as such (e.g. agree on common ontologies if that is necessary)?
11. How good is the framework in locating and using externally available functionality? Does it act like the user it acts on behalf of? How often does it fail to find a solution even though one exists? How often does it find the optimal solution? How short of optimal is the solution chosen by the framework if it is not the optimal one?
12. How well does the framework scale?

Finally, it is essential to keep in mind that the performance of any framework will depend on the specific context parameters at hand and must not be easily generalized:

13. For which types of applications, services or use cases are the answers to the previous questions valid? How do the answers change in a changing context?

According to the GQM methodology, the questions defining the software engineering goals motivating SWS are used to define the set of criteria that should be employed to evaluate SWS with respect to the goals. An analysis of the correlations among the questions was performed to derive the conceptual criteria model for the evaluation of SWS frameworks displayed in Figure 1. This model comprises the following five dimensions of evaluation.

Performance / Scalability

regards the runtime performance and scalability characteristics of a framework. It is typically measured by the computing resources required (e.g. processor time or memory). Questions 3 and 12 are related to this dimension.

Usability / Effort

regards the usability of the framework in terms of how much effort is required to set it up, maintain it, and use it. This dimension is influenced for instance by the complexity of the framework and the available tool support. Questions 4, 5, 7, 9, and 10 are related to this dimension.

Correctness

regards the quality of the support offered by the framework, i.e. to which degree a framework acts precisely like the user it acts on behalf of. This dimension is closely related to the often used notion of *expressivity*, that captures how precisely and comprehensively a service's capabilities and a user's needs can be formalized in a framework. Questions 1, 2, 4 – 7, and 11 are related to this dimension.

Cupling

regards how tightly coupled the providers and the consumers of services are in this framework, e.g. whether they have to agree on common ontologies or not. Questions 8 and 10 are related to this dimension.

Supported Scope and Automation

define the context for the other dimensions, since assessments made with respect to these will always depend on assumptions regarding the scenarios at hand. The notions of the scope of a framework and the supported degree of automation are closely related. The former defines the phases during service consumption covered by a framework (see Figure 1) while the latter defines the degree of automation that the framework provides for these phases. Questions 1, 2, 6 and 13 are related to this dimension.

Three remarks about this model need to be made. First, it is quite obvious, that some of the criteria dimensions are correlated negatively. A framework supporting full automation even for complex use cases requires a highly expressive language. On the other hand, less expressive languages will likely be easier to use and yield better runtime performance. Therefore, SWS frameworks need to aspire a balance between competing requirements. It is thus important to evaluate the dimensions identified above conjointly to make the corresponding tradeoffs explicit.

Second, this model has been primarily designed with the use cases of the first application type in mind. These involve the automation of tasks that previously, i.e. using established technologies, always involved a human in the loop. Therefore, such tasks do not always allow a direct comparison of SWS-based approaches with conventional software engineering approaches. Nevertheless it is important to keep in mind that SWS approaches do not only need to be comparable to each other, but that it is additionally necessary to show their relative advantage over traditional software engineering approaches. This regards primarily the *effort* dimension.

Third, designing a criteria model like the one proposed above involves some degree of freedom how to design it. We have followed the GQM methodology because this methodology directly links engineering goals to evaluation criteria through the questions that are first used to define the goals and then to derive the criteria. Thus, if properly implemented, this methodology ensures that the evaluation model is complete with respect to the identified engineering goals.

CURRENT EVALUATION INITIATIVES

The criteria model presented above allows examining the existing approaches to SWS evaluation in a systematic manner. In this section, we lay the basis for such analysis by introducing to the state of the art in SWS evaluation. An overview of the different approaches is shown in Table 1.

	S3 Contest	**SWS Challenge**	**WS Challenge**	**DIANE Benchmark**
Performance & Scalability	Query response time for static discovery	n.a.	Runtime for WS composition algorithms	n.a. (not generalizable to other approaches)
Usability & Effort	n.a.	Evaluates effort to react to changes in problem scenarios	n.a.	Preliminarily assessed via questionnaire
Correctness	Matchmaking recall and precision for given test collections in specific formalisms	n.a. (solutions are only submitted if they solve a scenario correctly)	Tests for validity of solutions, extra credit for minimal composition length	Evaluates correct formalization of given sample service requests
Coupling	Decoupled setting, descriptions provided by organizers	n.a. (offers and goals formalized by the same developers)	Decoupled setting, data provided by organizers	Tests correctness of results in an explicitly decoupled setting
Scope & Automation	So far limited to static discovery only	Differentiates functional coverage via hierarchy of problem scenarios	Limited to static composition only	Assumes full automation of discovery, selection and invocation

Table 1: Overview of dimensions of evaluation covered by existing initiatives.

S3 Contest on Semantic Service Selection

The S3 Contest on Semantic Service Selection[4] is an annual international contest for the comparative performance evaluation of implemented SWS matchmakers. Its first two editions were held in 2007 and 2008 in conjunction with the 6th and 7th International Semantic Web Conferences, a third edition will be held in conjunction with the 8th International Semantic Web Conference in Washington D.C., USA (October 2009). Depending on the availability of service retrieval test collections, the 2007 edition focused on OWL-S service matchmaking, while the 2008 edition broadened its scope to also cover SAWSDL matchmakers. Basic task of semantic service selection is to return a ranked set of service offer descriptions that are semantically relevant to given service request descriptions. The retrieval performance of matchmaker implementations is evaluated by measuring the classical retrieval performance in terms of recall/precision and F1-values (*correctness* and completeness of semantic service matching and ranking), as well as the *runtime performance* in terms of the average query response time and the aggregated runtime to match the complete test collection selected.

For comparative performance evaluation, the S3 contest readily provides the first publicly available semantic service matchmaker evaluation environment SME2[5] together with test collections OWLS-TC3 and SAWSDL-TC2 for OWL-S, respectively, SAWSDL services. As of today, both collections are widely used in practice with a reasonably high number of downloads for each. For example, OWLS-TC with around 10.000 downloads is comprised of 1.007 OWL-S services from 7 domains, 29 queries with respective binary and gradual relevance sets (in its current version OWLS-TC3). This collection has been jointly created by more than 30 users from different institutions based on the semantic annotation of public WSDL services and the inclusion of readily available OWL-S services as well. Regarding the definition of query relevance sets, binary and gradual relevance judgements of services followed the standard NIST-TREC with union average pooling.

At the 2008 edition, three OWL-S and two SAWSDL matchmaker implementations participated in the contest[6]. For the upcoming 2009 edition, six OWL-S matchmakers and four SAWSDL matchmakers are planned to be comparatively evaluated and discussed.

Semantic Web Services Challenge

The SWS Challenge Initiative[7] (Petrie et al., 2008b) was launched in March 2006 and has organized seven workshops and events since then. The challenge's main purpose is to provide a certification of SWS frameworks. The W3C SWS Testbed Incubator Group[8] aims to develop a standard evaluation methodology based on experiences gathered within the SWS-Challenge (Petrie et al., 2008a).

The approach of the SWS-Challenge is to define a set of detailed and realistic scenarios, each organized in different problem levels. Participants of the challenge try to solve these scenarios with their SWS technology. So far, two mediation scenarios involve building mediators to integrate systems in a purchase order and payment management scenario. Three more discovery scenarios target the automated discovery and invocation of suitable service providers for given specific service needs.

Until 2008, nine teams have participated in the challenge and were evaluated with respect to two aspects. Based on the set of increasingly complex problem levels in the Challenge's problem scenarios, the *functional coverage* of different SWS approaches is evaluated by assessing the extent to which approaches actually solved particular problem levels. This way a certification of the capabilities of particular technologies is provided. Additionally, but so far only for the mediation scenarios, the challenge tries to evaluate and compare the level of *effort* involved in adapting solutions to changes in the underlying problem scenario. By doing so the challenge tries to investigate the fundamental assumption of SWS that an increased usage of formal, declarative semantics will make solutions more flexible and easier to adapt to change.

WS Challenge

The IEEE Web Service Challenge (Bleul et al., 2009)[9], hosted annually at the IEEE Conferences on E-Commerce Technology (CEC) and Enterprise Computing, E-Commerce and E-Services (EEE), focuses on evaluating the correctness and runtime performance of web service discovery and composition algorithms. It was started in 2005 with the evaluation of syntactic service matchmaking and composition based on the string equivalence of WSDL part names. In 2006, this was complemented by a track on semantic matchmaking and composition based on the compatibility of XML schema types (Blake et al., 2006). In 2007 and 2008, the syntactic and matchmaking tracks have been discontinued to solely focus on semantic composition (Blake et al., 2007, Bansal et al., 2008). However, the semantics used by the challenge are much less expressive than usually employed in SWS frameworks. Semantic descriptions do not include service categories, pre- or postconditions, but are restricted to input and output parameters. These parameters are defined with respect to an XML-Schema type hierarchy that, from 2007 on, is represented in a simplified OWL version. So far, no semantics beyond inheritance relationships are used in the challenge.

The challenge provides a test environment and a test data generator. It evaluates the runtime of the composition algorithms and the correctness and quality (completeness, composition length, exploitation of parallel invocations in the compositions) of the discovered compositions. In 2009 the quality of compositions will be evaluated using their response time and throughput which must be computed from the provided response time and throughput of the component services. Additionally, there is an award for the best solution architecture (Bleul et al., 2009).

DIANE Benchmark

Within the DIANE project[10], a service description language (called DSD) and an accompanying middleware supporting service discovery, composition, and invocation have been developed. DIANE is one of the projects taking part in the SWS Challenge. In addition to the evaluation provided by the challenge, considerable effort has been put into devising a benchmark suite for semantic service description frameworks[11]. This benchmark has then been applied to DSD/DIANE (Fischer, 2005).

The DIANE Benchmark focuses on three aspects. The *effort* required to use the framework is assessed by measuring the initial effort to model the necessary ontologies as well as the continuous effort to maintain and update ontologies and service descriptions with the framework. The *correctness* of the framework is evaluated by assessing how well the semantics of given services can be captured by descriptions based on the employed formalism. Finally, the level of *coupling* is evaluated by determining to which degree the framework still yields correct results, if services and goals are formalized by different people in a completely decoupled way. The Benchmark also deals with the runtime performance and the correctness of the framework's implementation. However, since the corresponding parts cannot be easily generalized from DSD to other languages, they are not relevant for this article.

Other Approaches

Toma et al. (Toma et al., 2007) presented a framework for the evaluation of semantic matchmaking frameworks by identifying different aspects of such frameworks that should be evaluated: query and advertising language, scalability, reasoning support, matchmaking versus brokering, and mediation support. They evaluate a number of frameworks with regard to these criteria. The focus of the work is rather on the survey than on the comparison framework itself. While the framework does provide guidance for a structured comparison, it does not offer concrete test suites, measures, benchmarks or procedures for an objective comparative evaluation.

Moreover we have looked into the evaluation results of various SWS research projects (see for instance (Sîrbu et al., 2006, Sîrbu, 2006, Unspecified, 2006)). Most have spent a surprisingly small share of resources on evaluation or not published details about any evaluation performed. For example RW^2, an Austrian funded research project[12], has implemented different discovery engines for request and service description in different logical languages, respectively different granularity. However, as evaluation only a relatively small set of a couple of dozen handcrafted services exist. The EU projects DIP and ASG, for instance, have also developed similar discovery engines. With respect to evaluation they quote industrial case studies which, in essence, are also just a small set of service descriptions. Moreover, due to intellectual property rights restrictions the situation is even slightly worse, since not all descriptions are publicly available and a comparative evaluation is thus impossible.

Just recently, the EU funded SEALS project (Semantic Evaluation At Large Scale) has started in June 2009[13]. The goal of SEALS is to provide a reference infrastructure for automated benchmarking in the areas of ontology engineering tools, storage and reasoning systems, matching tools, semantic search tools and semantic web service tools. Furthermore, it will organize two international benchmarking campaigns for these research areas in 2010 and 2011. The formation and funding of this project is an important step towards better and more standardized evaluations, but since the project has just started, no results are available yet.

ANALYSIS OF EVALUATION METRICS AND MEASURES

This section presents an in-depth critical analysis of the measures that the four main approaches introduced above use for evaluations along the criteria dimensions of the evaluation model introduced above. An understanding for the fundamental research problems involved in SWS evaluation is developed by identifying current shortcomings and discussing possible improvements. This allows proposing a research agenda towards standard evaluation methodologies and benchmarks in the conclusions.

Performance / Scalability

Status:

A comparative evaluation of the runtime performance of different matchmaking algorithms is primarily provided by the S3 Contest. The experimental task to perform is to compare a given set of OWL-S (or SAWSDL) based request descriptions with a given set of OWL-S (or SAWSDL) based offer descriptions and identify the set of relevant offers for each request. This task is executed by the participating matchmaker implementations multiple times and the average query response time for single queries as well as the average total time to match all requests are measured. In 2007, the results for two matchmakers were roughly similar (11 respectively 9 minutes) whereas a third matchmaker required more than 20 hours to perform the task on a significantly downsized version of the test collection. Unfortunately, a detailed interpretation of the results is not available so far. An analysis of the causes for the poor performance of the third matchmaker would be important to investigate whether that poor performance is inherent to the particular matchmaking algorithm or has to be attributed to an unoptimized proof-of-concept implementation of the algorithm. It is hoped that participants of the contest are investigating the causes for encountered performance issues and will report on corresponding improvements in subsequent editions of the contest. It is worth noting that the S3 Contest evaluates the runtime performance and the correctness of the returned results, thereby allowing to put the runtime performance measures in relation to the achieved correctness.

The WS Challenge represents a very similar evaluation approach for composition instead of discovery algorithms. A testbed consisting of service descriptions, composition requests and an evaluation environment is provided to the participants. A composition requests specifies the available inputs and the desired outputs of a service. The task is to compose the provided available component services into a WS-BPEL process such that the process requires only available inputs and provides all desired outputs. The algorithms need to assess the composability of services using type inheritance relationships of the service's interfaces. Credit is given to the systems that solve the task best (see *Correctness*) and fastest. Apart of the different scope (composition versus discovery), the main difference between the WS Challenge and the S3 Contest is the origin of the test data used. While the S3 Contest relies on manually manufactured service descriptions, the WS Challenge uses a configurable, but fully automated test data generator.

Discussion and suggestions for improvements:

It is obvious that runtime performance measures are highly dependent on the test data used. Unfortunately, no standard test collection for the evaluation of SWS exists yet. To make experimental performance evaluations possible the test collections OWLS-TC[14] and SAWSDL-TC[15] have been developed. These are the only sizeable ones currently available and employed by the S3 Contest. Due to the tremendous effort involved and in order to reflect different views and different perspectives, standard test collections can only be built by the community as a whole. The organizers of the S3 Contest therefore invite the community to help the ongoing efforts across different institutions to further extend and improve the test collections OWLS-TC and SAWSDL-TC. Since the evaluation of retrieval performance bases on the availability of such test collections, the first three editions of the S3 Contest could only focus on matchmakers for OWL-S and SAWSDL services. Unfortunately, as of now, there is still no public test collection of WSML services available which is hoped to change soon. For the future, large standard test collections of the same set of services in at least all three major formalisms, that are OWL-S, WSML and the standard SAWSDL, need to be developed.

Another viable approach of using generated test data is the one taken by the WS-Challenge. However, generating test data that reliably resembles the characteristics of real world data is a continuous challenge. Furthermore, the automatic synthesis of test data is the more difficult, the more expressive and general the used formalisms are. More research on reliable test data generators is clearly desirable here.

Scalability is not yet explicitly evaluated, neither by the S3 Contest nor the WS Challenge. However, this could be done with limited additional effort. It requires splitting the test collections in subcollections of different sizes and exploring the degradation of the runtime performance with increasing size of the test data. Obviously the remarks about sensitivity towards the composition of the employed test collections apply in the same way as discussed above.

Conclusions:

Performance and scalability measures and their associated potential pitfalls are very well understood and have been used in all areas of software engineering for decades. Their application in the area of SWS is currently primarily hampered by practical issues. In contrast to a variety of theoretic work in the area of SWS matchmaking for instance, only few implementations of the proposed matchmaking algorithms are readily available. This is particularly true for the more sophisticated algorithms proposing the use of more expressive formalisms. The lack of readily downloadable tools is a blocker for better evaluations also with respect to other criteria.

Additionally, the lack of test collections of SWS has proven to be difficult to overcome. The effects of the properties and composition of the test collections on the evaluation results need to be studied carefully. This will allow building standard test collections or standard data generators that are diverse and balanced, ensuring reliable evaluations.

Usability / Effort

Status:

An initial attempt to evaluate the usability of a framework has been made within the DIANE Benchmark. The approach is based on evaluating the initial effort to create the necessary ontologies and the continuous effort to update and maintain these. The initial effort is evaluated by measuring the time it takes an experienced developer to formalize an ontology given as a UML model in the language of the target framework. The continuous effort to maintain a framework is estimated by the DIANE Benchmark via a questionnaire that tries to assess the quality of the available tool support and documentation.

Besides the approach of the DIANE Benchmark, significant effort has been devoted to develop a methodology to assess the flexibility of solutions within the SWS-Challenge. The approach is based on evaluating the effort necessary to adapt a solution for a given complete problem scenario to variations of that base scenario. Notably, approaches based on SWS as well as more traditional software engineering technologies participate in the SWS-Challenge. This allows to investigate not only the relative advantage of one SWS approach over another, but also to compare them with traditional technologies. A detailed description of the methodology employed by the SWS-Challenge and the difficulties encountered is available as a W3C Incubator Group Report (Petrie et al., 2008a).

Discussion and suggestions for improvement:

While the SWS-Challenge relies on complete natural language descriptions of scenarios, the DIANE Benchmark follows a much more restricted approach. It is thus easier to implement and involves less effort for participants. However, the task of formalizing an ontology given as a UML model prescribes the level of detail to be formalized. Lightweight frameworks, which do not exploit many details from the descriptions of services during the matchmaking, might be penalized with the effort of formalizing aspects which are of no use to them.

Generally, the choice of the right level of detail for a formalization of a problem still constitutes one of the core research problems in the area and should not be dictated by the testbed for an evaluation. Though experience with natural language scenario descriptions within the SWS-Challenge showed that these descriptions were ambiguous in several cases, such ambiguities were discovered by the participants and could subsequently be resolved. This way even scenarios described in natural language only become sufficiently well-defined over time.

It thus seems appropriate to combine both approaches, provide complete natural language descriptions of use cases (as the SWS-Challenge does) and evaluate the time necessary to implement these with a framework (in the spirit of the DIANE Benchmark). This setup reflects the strengths and weaknesses of the frameworks more adequately. A lightweight framework, for instance, might benefit from a reduced modelling effort but later suffer from poorer measures regarding the correctness of the achieved results.

Notably, this approach has not been taken so far. Because of the amount of work involved in implementing such an approach, the SWS-Challenge has resorted to evaluating the effort of implementing changes on top of existing solutions instead of evaluating the effort of creating the initial solutions in the first place. Furthermore, there were concerns that measuring the time needed to perform the necessary adaptations would lead to an unwanted competitive atmosphere and would be overly sensitive towards the personal performance of the programmer implementing the changes. As a consequence, the current approach is to measure the amount of code that needs to be changed instead of the time needed to implement those changes. Unfortunately, this change-based approach proved to be very difficult to implement in cases where code is not written as textual instructions but by assembling processes graphically in a GUI. A satisfying solution to this issue has still to be found. Similarly, the investigation of other compromises is still an open issue. It should be possible to develop scenarios with a sufficiently limited scope to make an evaluation of the overall effort of implementing them feasible.

Regarding the complementary questionnaire approach of the DIANE benchmark it is felt that a questionnaire is a good since lightweight starting point. However, the current implementation has several problems: The answering scheme (*yes* - *partially* - *no*) is too coarse-grained, some answers cannot be verified objectively and the weighting of the single questions in the total result is not based on experimental evidence.

Conclusions:

Overall, it seems that efforts regarding the evaluation of the usability and ultimately the increase in programmer productivity achieved through SWS frameworks are in their infancy and have not received appropriate attention so far. One of the problems currently hindering more extensive usability evaluations is the already mentioned lack of implementations and tools for the proposed algorithms. This lack of ready-to-use, well documented downloadable tools needs to be overcome by the community.

The lack of ready-to-use tools might also explain the fact that current evaluations have focused on usability on a technical level, e.g. investigated how long it takes to update an ontology. However, ontologies and their management are just a means and technology to achieve higher level goals. Therefore, such evaluation efforts need to be complemented by evaluations of the increase in productivity on a higher, more goal-oriented level. Such evaluations would also improve the comparability of SWS technology with traditional software engineering technologies, a crucial factor for the adoption of SWS by industry. The attempts of the SWS-Challenge to measure the flexibility of solutions are a promising step in this direction, but also illustrate that the question how to reliably and objectively measure an increase in productivity achieved by using different SWS approaches is a still unsolved research problem. Much more effort is needed here.

Correctness

Status:

Prior to the establishment of the S3 Contest in 2007, there have not been comparative correctness evaluations of different SWS matchmaking approaches at all. To get started, the S3 Contest borrowed the well-established evaluation approach from the series of TREC conferences[16] in information retrieval (IR) using the previously discussed OWLS-TC and SAWSDL-TC. Correctness of service matchmaking is evaluated by means of the traditional IR measures precision and recall. Precision measures the proportion of retrieved services, which are indeed relevant, and recall measures the proportion of relevant services, that are correctly retrieved. Until 2008, the contest relied on binary relevance judgments, i.e. service offers are judged as either relevant or irrelevant to a request, but no further ranking is considered. In 2009 graded relevance judgments will be introduced to the Contest.

The WS-Challenge investigates the correctness of service composition algorithms and gives extra credits to algorithms that return better (e.g., shorter) solutions. However, problems are defined such that finding good correct solutions is not difficult in principle, but algorithms are forced to traverse a very large search space. This way, the primary focus of the challenge is on evaluating the speed rather than the correctness of composition algorithms.

The SWS-Challenge focuses on functional coverage of frameworks (see below) and currently does not aim at providing quantitative measures for the correctness achieved by participating approaches. An entry to the challenge is usually developed until it correctly solves a scenario and not submitted otherwise.

The DIANE Benchmark presents two approaches to evaluate correctness. The first is similar to the approach of the S3 Contest but focuses on whether correct results can be achieved in an explicitly decoupled setting. It will be covered in the following Section. The other approach complements the S3 Contest in that it focuses on how well the real world semantics of services can be captured in the formalism used by a framework. It therefore attempts to evaluate correctness by experimentally evaluating the expressivity of the employed formalism. To define the benchmark, a group of test subjects not familiar with semantic web technology were asked to formulate service requests for two different application domains. The queries the test subjects devised were formulated in natural language. This resulted in about 200 requests. Additionally, domain experts developed ontologies they deemed necessary to handle the two domains. The evaluation approach of the benchmark is to measure the proportion of the 200 requests which can be formalized in a given framework correctly. Each request can be rated green (the request can be directly formalized), yellow (the request could be formalized with extensions to the domain ontologies) or red (the request cannot be appropriately expressed using the language constructs provided by the framework).

Discussion and suggestions for improvement:

The adoption of the well-established correctness measures precision and recall from IR is a self-evident first approach towards correctness measures in the field of SWS retrieval. Obviously, the general remarks about the sensitivity of evaluation results towards the composition of the employed test collection and the discussion about the lack of standard test collections across formalisms made in Section apply here, too.

However, as argued in (Küster et al., 2007b, Küster and König-Ries, 2009), traditional IR and SWS retrieval differ in that the former typically operates directly on the original resources, whereas the latter is based on formal semantics that are explicitly manually attached to the resources to support their precise and correct retrieval. Following the TREC evaluation approach the S3 Contest presets the semantic descriptions used for the retrieval. The major benefit of this approach is twofold: it mimics real world environments, where SWS descriptions are not formalized by the developers of a SWS matchmaker (see Section on Coupling) and it limits the effort involved in participation in the Contest. It does have the drawback, however, that recall and precision alone in such a setting can only be of limited significance. The problem is, that the question whether a semantic service description matches a semantic request description should be determinable unambiguously based on the formal semantics of the employed description formalism. In this aspect, it is unclear to what extent false results of the matchmaking (and thus a low precision and recall) should be attributed to inapt service and request formalizations or to shortcomings of the evaluated matchmaking algorithms.

Thus, an ideal evaluation of SWS retrieval correctness needs to cover two aspects: First, how well the real world semantics of services can be captured in the formalism used by a framework. Second, how effectively the framework's matchmaker can then exploit this information during the matchmaking.

An evaluation where the descriptions are preset is by design restricted to evaluating only the second aspect. On an implementation level, diverse test collections that contain service descriptions at various levels of detail and with varying complexity are required to evaluate this aspect reliably. Such collections are only partially available.

With respect to the first aspect, i.e. how to experimentally measure the quality of the formalization of a service's semantics possible in a framework, the DIANE Benchmark that relies on natural language service descriptions constitutes an important first achievement. Despite of that, an analysis of the evaluation of DSD performed with the DIANE Benchmark sheds light on two problems in the current setup of this part of the benchmark. First, the distinction between green and yellow ratings seems arbitrary in many cases. It remains unclear, why certain concepts were included in the initial ontologies (leading to green ratings) while others were not (leading to yellow ratings) and why this is a relevant measure for the expressivity of a framework. It seems more appropriate to evaluate the effort necessary to implement required extensions to the ontology and use this as a measure for the usability and effort of a framework. A framework whose formalism is expressive enough to support a flexible and elegant modelling would consequently benefit from a high score on this metric. The second problem is a lack of objectivity regarding green ratings. Green ratings are supported by providing formalizations of these requests in the target formalism. However, the judgment that these formalizations fully capture the semantics of the service (justifying a green rating) is made by the subjective estimate of the expert formalizing the requests. Such estimates need to be supported by an additional recall/precision analysis.

Conclusions:

Until recently, there have not been any comparative evaluations of the correctness achieved by different SWS framework at all. It is very promising that this important issue is starting to receive the attention it deserves. However, as can be seen from the discussion above a meaningful correctness evaluation is far from trivial and the above mentioned problems illustrate the need for further research in this direction:

First, current evaluations have either focused on the correctness of the matchmaking, or the correctness (or expressivity) of the formalization, but not on both. It needs to be investigated how this can be improved to achieve more reliable and comprehensive results. To this end, an alternate new track will be implemented in the 2009 edition of the S3 Contest where participants formalize given services themselves and the effects of different annotation styles to the matchmaker performance will be studied.

Second, current evaluations of correctness via recall and precision rely on binary relevance judgments. This approach has been a natural starting point, but does not reflect that virtually all SWS matchmakers support multi-valued matchmaking degrees and does not allow evaluating the important aspect of the quality of the ranking performed by SWS matchmakers. Necessary research on better measures, e.g. based on graded instead of binary relevance, has started and is ongoing (Tsetsos et al., 2006, Küster and König-Ries, 2008a). In fact, the 2009 edition will initially introduce the usage of such measures in the S3 Contest.

Third, the previously mentioned lack of standard test collections of SWS is even more critical for correctness evaluations than for performance evaluations. Reliable and meaningful correctness evaluations require diverse and realistic test data which can not be generated automatically. This test data needs to be available in natural language to experimentally evaluate the expressivity of a formalism employed by a framework. Additionally, complete and high quality semantic descriptions for a common set of services are required in different formalisms to effectively compare the correctness achieved by the various algorithms. Generally, the desirable properties of test collections need to be investigated more thoroughly and procedures how to obtain the necessary data and ensure its quality need to be developed (Küster and König-Ries, 2008c). It should be stressed that the indispensable standard test collections can only be built by the community as a whole. Everybody interested is thus invited to join the corresponding efforts.

Coupling

Status:

An evaluation of the level of coupling was so far not in the scope of the SWS-Challenge. Within the participating teams the same developers typically formalize all goal and offer descriptions. The WS-Challenge uses generated test data using one common schema type hierarchy. An investigation of the effects of decoupled creation of service annotations is not within its scope. Similarly it has not been explicitly in the focus of the S3 Contest so far.

The DIANE Benchmark presents an experimental setup to evaluate the degradation of delivered correctness in an explicitly decoupled setting. A number of inexperienced users are given an introduction to a framework and description formalism to be used. Subsequently they are divided into two groups that are not allowed to communicate with each other. A number of natural language service descriptions are provided as test data to the groups. The first group is asked to formalize them as offer descriptions, the second as request descriptions. Afterwards, the framework is used to match the resulting offer and request descriptions and precision and recall of the matchmaking are determined by considering request and goal descriptions relevant to each other if and only if they originate from the same natural language service description.

Discussion and suggestions for improvements:

The experience from applying this experimental setup to DIANE/DSD highlights an important issue: in practice, even using predefined ontologies, a high correctness is not easy to achieve in a decoupled setting. In the experiment a service that books a train ticket has been formalized as a service after whose execution a ticket is *reserved* by the first group. In contrast, the second group formalized the same service as a service after whose execution a ticket is *owned*. Subsequently, these different formalizations of the same real world semantics resulted in a false fail when the two descriptions were matched. This emphasizes the negative effects of variance in possible ways to formalize the real world semantics of a service. Such variance will inevitably be encountered in real world environments. It can be assumed that formalisms differ with respect to the likelihood of such modelling differences and that frameworks differ in how well they are able to handle them. Thus, a corresponding evaluation provides important clues about the performance of a framework in real world settings.

On a practical level, the DIANE Benchmark experiment needs to be considered preliminary. First, the remarks about binary relevance judgments in the context of SWS matchmaking made above apply here, too. Second, the test data defined by the DIANE Benchmark for this experiment (ten services) is currently much too small to support reliable results in practice. Further work is required to address both issues. To this end an improved implementation of the experiment based upon graded relevance measures and a much larger realistic testbed will be implemented as part of the 2009 edition of the S3 Contest[17].

Conclusions:

The importance of evaluating the level of coupling and its effects within SWS frameworks is illustrated by the experience from the preliminary experiment in the context of DIANE/DSD. Yet, this aspect has received much too little attention so far. Typically, research, development, and evaluation of a given SWS framework is performed within a single research team and thus in a tightly coupled setting. In contrast, the envisioned use cases for SWS target strongly decoupled settings. It is essential to start investigating the issues which may result from this discrepancy and to research methodologies to evaluate the tolerance of SWS frameworks towards these.

Scope and Automation

Status:

It is the main evaluation goal of the SWS-Challenge to evaluate the functional scope of participating frameworks. The approach is to define a set of problem scenarios, which consist of increasingly complex problem levels. This may be illustrated by an example from one of the discovery scenarios. The shipping scenario deals with discovering, binding and invoking a suitable shipping service for a given concrete shipment request. Five services with different pricing models and different functional restrictions, e.g. on the operation range, are specified. The various problem levels are defined by concrete requests that require taking more and more aspects of a service into consideration during the matchmaking. The first level requires to discover a shipper based on operation range, the second level requires to take restrictions regarding the weight of the parcel into account, the third level includes price restrictions, which in the case of one service requires to dynamically inquire price information at the service’s endpoint, the fourth level requires basic composition capability to support shipping of multiple packages and the fifth level requires to reason about temporal constraints regarding pickup times and shipping durations. Participating solutions are certified at the Challenge workshops with respect to whether they are able to solve a particular problem level correctly. A review of the code during the workshop ensures, that frameworks actually solve the problems by reasoning about the formalized problem semantics and not hard-wiring the correct solution.

An evaluation of the scope of frameworks is neither performed by the S3 Contest or the WS Challenge nor within the scope of the DIANE Benchmark. The S3 Contest is limited to static discovery of services, i.e. discovery which identifies relevant services based on static descriptions only. The WS Challenge is (since the discontinuation of the discovery track in 2007) concerned with fully automated service composition exclusively. The DIANE Benchmark assumes support for dynamic discovery, ranking, selection and invocation and does not provide a fine-grained evaluation of frameworks, which only support some of these tasks.

Conclusions:

The evaluation and certification of the functional scope of SWS frameworks is at the focus of the main SWS evaluation initiative and has been subject of a corresponding W3C Incubator Group[18]. It therefore does not come as a surprise that the underlying methodology is quite mature meanwhile. However, the SWS-Challenge comprises only five scenarios so far which cover only parts of the problem space yet. Many more scenarios are needed to provide a more complete coverage of the problem space. More scenarios would also bring in the different perspectives and assumptions of different research groups in the area and thus help to confirm or revise the existing evaluation results. A continuous call for submission of scenario proposals is thus part of the SWS-Challenge[19]. More community response is still desired here. Fundamentally, also more research on methodologies that help ensuring the relevance and a certain completeness and balance of the testbed of scenarios is required. Experience from building similar testbeds in other fields of computer science may serve as important input here.

SUMMARY AND DIRECTIONS FOR FUTURE RESEARCH

Doing science means producing reproducible results that can be independently evaluated. This is an indispensable requirement for scientific progress and industrial adoption of research results. In the area of SWS, only three years ago, it was impossible to evaluate and compare different approaches by different groups across formalisms in a fair and objective way. The meanwhile formation of international open evaluation campaigns like the SWS-Challenge, the S3 Contest and the WS Challenge is a very promising and significant achievement towards reproducibility and third-party verification of results and thus, ultimately, towards the vision of bringing semantic web services to reality. These campaigns have been possible only by community participation.

In this section, we summarize once more the main conclusions from the last sections and suggest a research agenda to further improve the existing evaluation initiatives and SWS evaluations in general. Like the work already achieved, these further steps can only be realized collaboratively and require even more help and participation from the community. After all, evaluations become the more objective, reliable, and meaningful the more groups contribute to the testbeds and participate in the evaluation events. You are therefore invited to join now to participate in the further evolution of the SWS evaluation campaigns with respect to the issues listed below and others that you may bring in.

Summary of the Status

Here is a very brief summary of the status in SWS evaluation as discussed in the previous section:

- With respect to *performance and scalability* on the one hand more and better implementations of matchmakers are needed, on the other hand, standard SWS test collections need to be build.
- To meaningfully evaluate SWS frameworks' *usability and amount of effort* more fundamental work is needed, in particular, suitable measures on a high level of abstraction need to be identified.
- Concerning *correctness* what is lacking is a unified approach to evaluating correctness of matchmakers and formalisms, fine grained criteria that are suitable to measure correctness more precisely, and sufficiently large standard test collections.
- *Decoupling* has not been regarded in depth yet, so reliable measures need to be defined. A foundation of those would be, again, standard test collections.
- *Functional Scope and Level of Automation* is probably the most thoroughly investigated of all the criteria. Nevertheless, to reach meaningful results, a more diverse set of scenarios and a closer analysis of the dependence between scenario, approach, and performance are needed.

Overall, SWS evaluation is an emerging field. Much more research is necessary to develop measures that are sufficiently mature to become standard. Below, we suggest the most important activities towards such standard measures. Closely related areas of computer science have many years of experience in developing their evaluations. Among many others, there are TREC, NTCIR[20] and INEX[21] from information retrieval, the ICAPS Competitions[22] from the planning community, the Trading Agent Competitions[23] from the agent community, or the series of EON workshops[24] from the ontology evaluation community. These initiatives have succeeded to set approved standards in their communities. They provide valuable experience and input to the SWS evaluation domain that obviously should be taken into account in any further activities. Successful such examples are the collaboration of the SWS-Challenge with the last EON workshop[24] or the proposed adoption of retrieval evaluation measures from INEX and NTCIR to the SWS evaluation domain (Küster and König-Ries, 2008a).

Building Standard SWS Test Collections

We have outlined above that one major, still lacking prerequisite for meaningful evaluation of SWS frameworks with respect to virtually all criteria are standardized SWS test collections. Research on how to build such collections is required. In our opinion, they need to:

- support several formalisms to make comparisons across approaches feasible,
- be diverse with respect to use cases, their complexity, domains covered etc.,
- contain realistic services that are described in sufficient detail to take advantage of the power of semantic approaches,
- provide natural language descriptions of the services to simplify usability and cross-formalism evaluation,
- be sufficiently large to support a statistically significant number of tests,
- contain offers and requests that were developed independent of each other to allow testing for decoupling,
- and contain services contributed by as many different groups as possible to avoid an unintended bias towards a particular approach.

Such test collections can not be provided by an individual group, not only because that would violate the last requirement, but also because the effort involved in building such a collection is tremendous. Therefore, the community has to work together to create these collections. In order to support community involvement, suitable tools are needed. These had been lacking, but improvements in this respect were recently achieved with the releases of the OPOSSum Portal[25] (Küster and König-Ries, 2008b).

Making Evaluations More Reliable

Current approaches build on a limited number of services as well as a limited number of scenarios. In order to overcome the first, we already identified the need for standard test collections. These alone will not suffice, though. What is also needed is a more systematic investigation of the context parameters of SWS usage scenarios. Testbeds should contain different types of scenarios and somewhat redundant similar scenarios. Once these scenarios have been developed, it will become possible to investigate how results change in different scenarios, whether this change is due to the type of scenario or change, and which influence different assumptions (about the degree of automation desired, the complexity of choreography, the diversity of underlying services etc.) have on the outcome of the evaluation.

Only such research will allow controlling the influence of context parameters on the evaluations. This will make them not only more reliable, but also more useful in providing potential users with the guidance they wish to have for their decisions regarding which approach to use for a particular task.

Unifying Existing Evaluation Approaches

As outlined in the section on evaluation criteria, evaluations are much more significant, if they cover all dimensions of evaluation conjointly. As can be seen from Table 1 this is not yet the case. Existing testbeds and evaluations need to be integrated more closely.

Further Activities

Up to now, we have listed a number of activities that must come from inside the semantic web services community to improve the situation with regard to evaluation of their research results. To support such activities, here, we list measures that could be taken more from the outside or more on a meta-level to help with this:

First, funding agencies should put more emphasis on proper evaluations. They should both make sure that evaluations promised in the proposals really take place and should explicitly fund research geared towards evaluation.

Second, the relevant conferences in the field should think about adding evaluation tracks comparable to the one the VLDB conferences has started in its 2008 edition. Together, these measures will ensure that evaluation gets more visibility and a higher priority - something that our field is in dire need of.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the contributions of Dr. Charles Petrie, who co-authored a prior version of this article.

Notes

[1] http://cordis.europa.eu/fp6/projects.htm

[2] http://trec.nist.gov/

[3] https://www.goldpractices.com/practices/gqm/

[4] http://www.dfki.de/~klusch/s3/

[5] http://projects.semwebcentral.org/projects/sme2/

[6] http://dfki.de/~klusch/s3/html/2008.html

[7] http://sws-challenge.org/

[8] http://www.w3.org/2005/Incubator/swsc/

[9] http://www.ws-challenge.org/

[10] http://hnsp.inf-bb.uni-jena.de/DIANE/

[11] http://hnsp.inf-bb.uni-jena.de/Diane/benchmark/

[12] http://rw2.deri.at/

[13] http://seals-project.eu

[14] http://projects.semwebcentral.org/projects/owls-tc/

[15] http://www.semwebcentral.org/projects/sawsdl-tc/

[16] http://trec.nist.gov/

[17] http://fusion.cs.uni-jena.de/professur/jgdeval

[18] http://www.w3.org/2005/Incubator/swsc/

[19] http://sws-challenge.org/wiki/index.php/Scenarios

[20] http://research.nii.ac.jp/ntcir/

[21] http://inex.is.informatik.uni-duisburg.de/

[22] http://icaps-conference.org/index.php/Main/Competitions

[23] http://www.sics.se/tac/

[24] http://sws-challenge.org/wiki/index.php/EON-SWSC2008

[25] http://fusion.cs.uni-jena.de/OPOSSum/

References

Balzer, S., Liebig, T., and Wagner, M. (2004). Pitfalls of OWL-S: a practical semantic web use case. In Proceedings of the Second International Conference on Service-Oriented Computing (ICSOC2004), New York, NY, USA.

Bansal, A., Blake, M. B., Kona, S., Bleul, S., Weise, T., and Jaeger, M. C. (2008). WSC-08: Continuing the web services challenge. In Proceedings of the 10th IEEE International Conference on E-Commerce Technology (CEC2008) / 5th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE2008), pages 351–354, Washington, DC, USA.

Basili, V. R. (1992). Software modeling and measurement: the goal/question/metric paradigm. Technical report, University of Maryland at College Park, College Park, MD, USA.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. Scientific American, 5.

Blake, M. B., Cheung, W., Jaeger, M. C., and Wombacher, A. (2006). WSC-06: the web service challenge. In Proceedings of the Eighth IEEE International Conference on E-Commerce Technology (CEC 2006) and Third IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE 2006), Palo Alto, California, USA.

Blake, M. B., Cheung, W. K.-W., Jaeger, M. C., and Wombacher, A. (2007). WSC-07: evolving the web services challenge. In Proceedings of the 9th IEEE International Conference on E-Commerce Technology (CEC 2007), Tokyo, Japan.

Bleul, S., Weise, T., and Geihs, K. (2009). The web service challenge - a review on semantic web service composition. In Proceedings of the Workshop on Service-Oriented Computing at KIVS 2009, Kassel, Germany.

Cobo, J. M. L., Losada, S., Corcho, Ó., Benjamins, V. R., Niño, M., and Contreras, J. (2004). SWS for financial overdrawn alerting. In Proceedings of the Third International Semantic Web Conference (ISWC2004), Hiroshima, Japan.

Colucci, S., Noia, T. D., Sciascio, E. D., Donini, F. M., and Mongiello, M. (2005). Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. Electronic Commerce Research and Applications, 4(4):345–361.

Fischer, T. (2005). Entwicklung einer Evaluationsmethodik für Semantic Web Services und Anwendung auf die DIANE Service Descriptions (in German). Master's thesis, IPD, University Karlsruhe.

Friesen, A. and Grimm, S. (2005). DIP deliverable D4.8: Discovery specification. Technical report.

Friesen, A. and Namiri, K. (2006). Towards semantic service selection for B2B integration. In Proceedings of the Joint Workshop on Web Services Modeling and Implementation using Sound Web Engineering Practices and Methods, Architectures and Technologies for e-service Engineering (SMIWEP-MATeS'06) at the Sixth International Conference on Web Engineering (ICWE06), Palo Alto, CA, USA.

Gugliotta, A., Tanasescu, V., Domingue, J., Davies, R., Gutiérrez-Villarías, L., Rowlatt, M., Richardson, M., and Stinčić, S. (2006). Benefits and challenges of applying semantic web services in the e-government domain. Semantics 2006.

Harman, D. (1992). Overview of the first Text REtrieval Conference (TREC-1). In Proceedings of the first Text REtrieval Conference (TREC-1), Gaithersbury, MD, USA.

Klein, M., König-Ries, B., and Müssig, M. (2005). What is needed for semantic service descriptions - a proposal for suitable language constructs. International Journal on Web and Grid Services (IJWGS), 1(3/4):328–364.

Klusch, M. (2008a). Semantic web service coordination. In M. Schumacher, H. H., editor, CASCOM - Intelligent Service Coordination in the Semantic Web, chapter 4. Springer.

Klusch, M. (2008b). Semantic web service description. In M. Schumacher, H. H., editor, CASCOM - Intelligent Service Coordination in the Semantic Web, chapter 3. Springer.

Klusch, M. and Zhing, X. (2008). Deployed semantic services for the common user of the web: A reality check. In Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC2008), Santa Clara, CA, USA.

Küster, U. and König-Ries, B. (2007). Supporting dynamics in service descriptions - the key to automatic service usage. In Proceedings of the Fifth International Conference on Service Oriented Computing (ICSOC07), Vienna, Austria.

Küster, U. and König-Ries, B. (2008a). Evaluating semantic web service matchmaking effectiveness based on graded relevance. In Proceedings of the 2nd International Workshop SMR^2 on Service Matchmaking and Resource Retrieval in the Semantic Web at the 7th International Semantic Web Conference (ISWC08), Karlsruhe, Germany.

Küster, U. and König-Ries, B. (2008b). On the empirical evaluation of semantic web service approaches: Towards common SWS test collections. In Proceedings of the 2nd IEEE International Conference on Semantic Computing (ICSC2008), Santa Clara, CA, USA.

Küster, U. and König-Ries, B. (2008c). Towards standard test collections for the empirical evaluation of semantic web service approaches. International Journal of Semantic Computing, 2(3):381–402.

Küster, U. and König-Ries, B. (2009). Relevance judgments for web services retrieval - a methodology and test collection for sws discovery evaluation. In Proceedings of the 7th IEEE European Conference on Web Services (ECOWS09), Einhoven, The Netherlands.

Küster, U., König-Ries, B., Klein, M., and Stern, M. (2007a). DIANE - a matchmaking-centered framework for automated service discovery, composition, binding, and invocation on the web. International Journal of Electronic Commerce (IJEC) - Special Issue: Semantic Matchmaking and Resource Retrieval on the Web, 12(2):41–68.

Küster, U., Lausen, H., and König-Ries, B. (2007b). Evaluation of semantic service discovery - a survey and directions for future research. In Proceedings of the 2nd Workshop on Emerging Web Services Technology (WEWST07) at the 5th IEEE European Conference on Web Services (ECOWS07), Halle (Saale), Germany.

Lausen, H., Petrie, C., and Zaremba, M. (2007). W3C SWS testbed incubator group charter. Available online at http://www.w3.org/2005/Incubator/swsc/charter.

Li, L. and Horrocks, I. (2003). A software framework for matchmaking based on semantic web technology. In Proceedings of the 12th World Wide Web Conference (WWW2003), Budapest, Hungary.

McIlraith, S. A., Son, T. C., and Zeng, H. (2001). Semantic web services. IEEE Intelligent Systems, 16(2):46–53.

Petrie, C., Küster, U., and Margaria-Steffen, T. (2008a). W3C SWS challenge testbed incubator methodology report. W3C incubator report, W3C. available online at http://www.w3.org/2005/Incubator/swsc/XGR-SWSC/.

Petrie, C., Lausen, H., Zaremba, M., and Margaria, T., editors (2008b). Semantic Web Service Challenge - Results from the First Year. Springer, Semantic Web and Beyond, Vol. 8.

Preist, C. (2004). A conceptual architecture for semantic web services (extended version). Technical Report HPL-2004-215, HP Laboratories Bristol.

Preist, C. (2007). Goals and vision: Combining web services with semantic web technology. In Semantic Web Services: Concepts, Technologies, and Applications, pages 159–178. Springer-Verlag New York, Inc.

Preist, C., Cuadrado, J. E., Battle, S. A., Grimm, S., and Williams, S. K. (2005). Automated business-to-business integration of a logistics supply chain using semantic web services technology. In Proceedings of the Fourth International Semantic Web Conference, Galway, Ireland.

Ragone, A., Straccia, U., Noia, T. D., Sciascio, E. D., and Donini, F. M. (2007). Vague knowledge bases for matchmaking in P2P e-marketplaces. In Proceedings of the 4th European Semantic Web Conference (ESWC2007), Innsbruck, Austria.

Sim, S. E., Easterbrook, S. M., and Holt, R. C. (2003). Using benchmarking to advance research: A challenge to software engineering. In Proceedings of the 25th International Conference on Software Engineering (ICSE2003), Portland, Oregon, USA.

Sîrbu, A. (2006). DIP deliverable D4.14: Discovery module prototype. Technical report.

Sîrbu, A., Toma, I., and Roman, D. (2006). A logic based approach for service discovery with composition support. In Proceedings of the ECOWS06 Workshop on Emerging Web Services Technology, Zürich, Switzerland.

Stollberg, M. (2004). SWF use case. WSMO working draft D3.5. available at http://swf.deri.at/usecase/20041019/SWFUseCase-20041019.pdf.

Toma, I., Iqbal, K., Roman, D., Strang, T., Fensel, D., Sapkota, B., Moran, M., and Gomez, J. M. (2007). Discovery in grid and web services environments: A survey and evaluation. International Journal on Multiagent and Grid Systems, 3(3).

Tsetsos, V., Anagnostopoulos, C., and Hadjiefthymiades, S. (2006). On the evaluation of semantic web service matchmaking systems. In Proceedings of the 4th IEEE European Conference on Web Services (ECOWS2006), Zürich, Switzerland.

Unspecified (2006). RW2 project deliverable D2.3: Prototype implementation of the discovery component. Technical report.