HAL
open science

The state of semantic technology today: overview of the first SEALS evaluation campaigns

Lyndon Nixon, Raúl García Castro, Stuart Wrigley, Mikalai Yatskevich, Cassia Trojahn dos Santos, Liliana Cabral

To cite this version:

Lyndon Nixon, Raúl García Castro, Stuart Wrigley, Mikalai Yatskevich, Cassia Trojahn dos Santos, et al.. The state of semantic technology today: overview of the first SEALS evaluation campaigns. Proc. 7th ACM international conference on semantic systems (I-semantics), Sep 2011, Graz, Austria. pp.134-141, 10.1145/2063518.2063536 . hal-00781025

HAL Id: hal-00781025

https://inria.hal.science/hal-00781025

Submitted on 25 Jan 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The state of semantic technology today – overview of the first SEALS evaluation campaigns

Lyndon Nixon
STI International
lyndon.nixon@sti2.org

Raúl García-Castro
Universidad Politecnica de Madrid
rgarcia@fi.upm.es

Stuart Wrigley
University of Sheffield
s.wrigley@dcs.shef.ac.uk

Mikalai Yatskevich
University of Oxford
yatsevi@comlab.ox.ac.uk

Cássia Trojahn dos Santos
INRIA
cassia.trojahn@inrialpes.fr

Liliana Cabral
The Open University
L.S.Cabral@open.ac.uk

ABSTRACT
This paper describes the first five SEALS Evaluation Campaigns over the semantic technologies covered by the SEALS project (ontology engineering tools, ontology reasoning tools, ontology matching tools, semantic search tools, and semantic web service tools). It presents the evaluations and test data used in these campaigns and the tools that participated in them along with a comparative analysis of their results. It also presents some lessons learnt after the execution of the evaluation campaigns and draws some final conclusions.

Categories and Subject Descriptors
D.2.8 [**Metrics**]. D.4.8 [**Performance**]

General Terms
Documentation, Performance, Design, Experimentation.

Keywords
Evaluations, benchmarking, metrics, semantic technology.

1. INTRODUCTION

The role of the SEALS project is two-fold: to create a lasting infrastructure for evaluating semantic technologies and to organise and execute two series of international evaluation campaigns over the different types of semantic technologies covered in the project.

Over the past 18 months, the SEALS consortium has designed and implemented a general methodology for carrying out evaluation campaigns. Using the current infrastructure, we have organized five international evaluation campaigns focused on ontology engineering tools, ontology reasoning systems, ontology matching tools, semantic search tools and semantic web services tools. Each of these five evaluation campaigns was conducted during the Summer of 2010.

This paper provides a summary of these first five SEALS Evaluation Campaigns; further details about the campaigns and their results can be found in the SEALS website pages[1] and public deliverables[2] devoted to each of the campaigns.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
I-SEMANTICS 2011, 7th Int. Conf. on Semantic Systems, Sept. 7-9, 2011, Graz, Austria
Copyright 2011 ACM 978-1-4503-0621-8 …$10.00..

2. THE EVALUATION CAMPAIGNS

In the SEALS project, a common methodology and process for organizing an executing evaluation campaigns was defined, based in an analysis of previous evaluation campaigns in different domains [1]. The SEALS evaluation campaign process is composed of four main phases which are now described.

Initiation. During this phase, an initial effort was performed to initiate and coordinate all the evaluation campaigns. To this end, first, the organizers of the evaluation campaigns were identified. In SEALS there is one committee in charge of the general organization and monitoring of all the evaluation campaigns and there have been different committees in charge of organizing the evaluation scenarios of each evaluation campaign and of taking them to a successful end. Then, the different evaluation scenarios to be executed in each evaluation campaign were discussed and defined. This involved describing the evaluation to be performed over the tools and the test data to be used in it.

Involvement. In order to involve participants, the campaigns were announced using different mechanisms: the project dissemination mechanisms (e.g., portal, blog), relevant mailing lists in the community, leaflets and presentations in conferences and workshops, etc. Participant registration mechanisms were prepared in the SEALS Community portal to allow potential participants to indicate their interest in the evaluation campaigns. Even if not every material to be used in the evaluation scenarios was ready by that time, this allowed involving participants early in the campaign.

Preparation and execution. In this phase, the organizers of each evaluation campaign provided to the registered participants with all the evaluation materials needed in the evaluation (e.g., descriptions of the evaluation scenarios and test data, instructions on how to participate, etc.). These materials were made available through the SEALS Community Portal[3]. In the SEALS project we have developed the SEALS Platform to support the execution of evaluations by providing different services to manage test data, execute evaluations, manage evaluation results, and so on. Participants connected their tools with the SEALS Platform and,

[1] http://www.seals-project.eu/seals-evaluation-campaigns/1st-evaluation-campaigns

[2] http://about.seals-project.eu/deliverables

[3] http://www.seals-project.eu

in the case of some relevant tools, members of the SEALS project connected them. Once all the participating tools were connected to the SEALS Platform, the different evaluation scenarios were executed with the corresponding test data and tools. The results obtained were stored in the platform and later analysed; in most of the cases, result visualisation services were developed to facilitate this analysis.

Dissemination. The results of all the evaluation campaigns were published in public SEALS deliverables and disseminated jointly in the International Workshop on Evaluation of Semantic Technologies[4] and separately in other events. Also, a white paper has been produced to provide an overview of the five evaluation campaigns and their results[5]. Finally, all the evaluation resources used in the evaluations have been made publicly available through the SEALS Platform.

3. THE SEALS PLATFORM

The SEALS Platform offers independent computational and data resources for the evaluation of semantic technologies and, as mentioned in the previous section, we used the first versions of the evaluation services developed for the platform to execute the evaluation scenarios of each evaluation campaign.

The SEALS Platform follows a service-oriented approach to store and process semantic technology evaluation resources. Its architecture comprises a number of components, shown in Figure 1, each of which are described below.

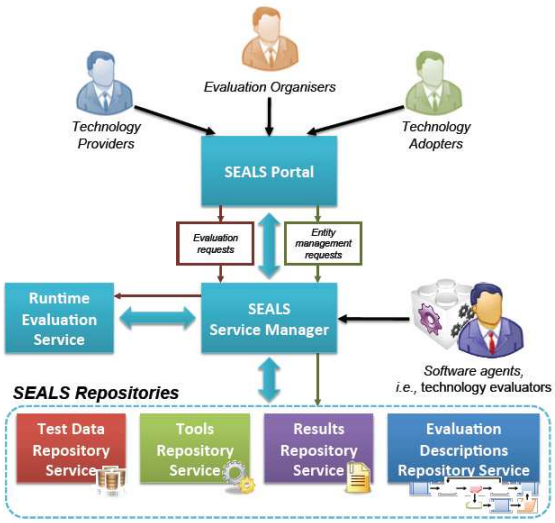

Figure 1. Architecture of the SEALS Platform.

SEALS Portal. The SEALS Portal provides a web user interface for interacting with the SEALS Platform. Thus, the portal will be used by the users for the management of the entities in the SEALS Platform , as well as for requesting the execution of evaluations. The portal will leverage the SEALS Service Manager for carrying out the users' requests.

SEALS Service Manager. The SEALS Service Manager is the core module of the platform and is responsible for coordinating the other platform components and for maintaining consistency within the platform. This component exposes a series of services that provide programmatic interfaces for the SEALS Platform. Thus, apart from the SEALS Portal, the services offered may be also used by third party software agents.

SEALS Repositories. These repositories manage the entities used in the platform (i.e., test data, tools, evaluation descriptions, and results).

Runtime Evaluation Service. The Runtime Evaluation Service is used to automatically evaluate a certain tool according to a particular evaluation description and using some specific test data.

4. ONTOLOGY ENGINEERING TOOLS EVALUATION CAMPAIGN

The SEALS Evaluation Campaign for Ontology Engineering Tools included three scenarios to evaluate the conformance, interoperability and scalability of these tools. In the conformance and interoperability scenarios we aimed to fully cover the RDF(S) and OWL specifications; in the scalability scenario we evaluated tools using both real-world ontologies and synthetic test data.

4.1 Previous evaluations

The first characteristic that we have covered in the evaluation campaign is conformance. Previously, conformance has only been measured in qualitative evaluations that were based on tool specifications or documentation, but not on running the tools and obtaining results about their real behaviour (e.g., the evaluation performed in the OntoWeb project [2] or the one performed by Lambrix and colleagues [3]).

Besides, some previous evaluations provided some information about the conformance of the tools since such conformance affected the evaluation results. This is the case of the EON 2002 ontology modelling experiment [4], the EON 2003 interoperability experiment [5], or the evaluations performed in the RDF(S) [6] and OWL [7] Interoperability Benchmarking activities.

However, currently the real conformance of existing tools is unknown since such conformance has not been evaluated. Therefore, we have evaluated the conformance of ontology engineering tools and we have covered the RDF(S) and OWL recommendations.

A second characteristic that we have covered, highly related to conformance, is interoperability. Previously, an interoperability experiment was proposed in the EON 2003 workshop [5] where participants were asked to export and import to an intermediate language to assess the amount of knowledge lost during these transformations.

Later, the RDF(S) and OWL Interoperability Benchmarking activities involved the evaluation of the interoperability of different types of semantic technologies using RDF(S) and OWL as interchange languages and provided a set of common test data, evaluation procedures and software to support these evaluations. In this evaluation campaign we have extended these evaluations with test data for OWL DL and OWL Full to fully cover the RDF(S) and OWL specifications.

Scalability is a main concern for any semantic technology, including ontology engineering tools. Nevertheless, only one effort was previously performed for evaluating the scalability of this kind of tools (i.e., the WebODE Performance Benchmark

[4] http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-666/

[5] http://www.seals-project.eu/whitepaper

Suite [8]) and it was specific to a single tool. In scalability evaluations, the generation of test data is a key issue. The WebODE Performance Benchmark Suite includes a test data generator that generates synthetic ontologies in the WebODE knowledge model according to a set of load factors and these ontologies can be later exported to several languages (RDF(S), OIL, DAML+OIL, OWL, etc.). Also, one of the most common test data generators used when evaluating ontology management frameworks is the Lehigh University Benchmark (LUBM)[9].

On the other hand, other evaluations use real ontologies as test data (e.g., subsets of ARTstor art metadata and the MIT OpenCourseWare metadata) were used in the scalability evaluation performed in the SIMILE project [10].

In this first evaluation campaign we have established the grounds for the automatic evaluation of the scalability of ontology engineering tools, using both real ontologies and generated data, with the aim of proposing an extensible approach to be further extended in the future.

4.2 Results of the SEALS evaluations

The SEALS evaluation campaign defined three scenarios to evaluate the conformance, interoperability and scalability of ontology engineering tools. The conformance and interoperability evaluations have covered the RDF(S) and OWL specifications. To this end, we will use four different test suites that contain synthetic ontologies with simple combinations of components of the RDF(S), OWL Lite, OWL DL, and OWL Full knowledge models. The RDF(S) and OWL Lite Import Test Suites already exist (they were named the RDF(S) Import Benchmark Suite[6] and the OWL Lite Import Benchmark Suite[7], respectively) and detailed descriptions of them can be found in [11]. The OWL DL and OWL Full Import Test Suites have been developed in the context of the SEALS project and are described in [12]. Next, we provide a brief description of them.

The OWL DL Import Test Suite contains OWL DL ontologies that have been generated following a keyword-driven test suite generation process implemented by the OWLDL Generator tool[8]. The OWL Full Import Test Suite is complementary to both the RDF(S) Import Test Suite and the OWL DL Import Test Suite. On the one hand, the test suite provides ontologies that are syntactically valid in OWL Full but generally invalid in OWL DL. On the other hand, the test suite makes specific use of OWL vocabulary terms and therefore goes beyond the typical content of RDF(S) ontologies. For scalability tests, we selected 20 ontologies of various sizes (up to 37.7 Mb).

In the first evaluation campaign over ontology engineering tools we have evaluated six different tools: three ontology management frameworks (Jena, the OWL API, and Sesame) and three ontology editors (the NeOn Toolkit, Protege OWL, and Protégé version 4). In the conformance and interoperability results, we can see that all those tools that manage ontologies at the RDF level (Jena and Sesame) have no problems in processing ontologies regardless of the ontology language. Since the rest of the tools evaluated are based in OWL or in OWL 2, their conformance and interoperability is clearly better when dealing with OWL ontologies.

Since the OWL Lite language is a subset of the OWL DL one, there is a dependency between the results obtained using the test suites for OWL Lite and OWL DL. In the results we can also see that, since the OWL DL test suite is more exhaustive than the OWL Lite one, the OWL DL evaluation unveiled more problems in tools than the OWL Lite evaluation. These included issues not only related to the OWL DL language, but also related to OWL Lite ontologies included in the OWL DL test suite. The results also show the dependency between the results of a tool and those of the ontology management framework that it uses; using a framework does not isolate a tool from having conformance or interoperability problems. Besides inheriting existing problems in the framework (if any), a tool may have more problems if it requires further ontology processing (e.g., its representation formalism is different from that of the framework or an extension of it) or if it affects the correct working of the framework.

However, using ontology management frameworks may help increasing the conformance and interoperability of the tools, since developers do not have to deal with the problems of low-level ontology management. Nevertheless, as observed in the results, this also requires being aware of existing defects in these frameworks and regularly updating the tools to use their latest versions.

The results of the scalability evaluation showed the linear dependence between the ontology size and export/import operations execution. However, the performance between the tools varies to a considerable extent, namely between Sesame, Jena and Protégé OWL. As the OWL API is used in the NeOn Toolkit and Protégé version 4, the performance is practically the same for them. Therefore, based on the obtained evaluation results we can conclude that Sesame is one of the most suitable tools for handling large ontologies.

5. STORAGE AND REASONING EVALUATION CAMPAIGN

The SEALS Storage and Reasoning Systems evaluation campaign focused on interoperability and performance evaluation of advanced reasoning systems. Class satisfiability, classification, ontology satisfiability and entailment evaluations have been performed on a wide variety of real world tasks.

5.1 Previous evaluations

Definition, execution, and analysis of evaluations for testing description logic based systems (DLBS) has been extensively considered in the past to compare the performances of these kind of systems and to prove their suitability for real case scenarios. The implementation of new optimisations for existing DLBS or the development of new DLBS has been disseminated together with specific evaluations to show how they improve the state of the art of DLBS.

Several attempts to systematise the evaluation of DLBS and to provide a lasting reference framework for automation of this kind of evaluations have failed in the past. The community of developers and researchers of DLBS still do not have a common open platform to execute evaluations and to study the results of these executions.

[6] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/rdfs/rdfs_import_benchmark_suite.html
[7] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/owl/import.html
[8] http://knowledgeweb.semanticweb.org/benchmarking_interoperability/OWLDLGenerator/

The test data, DLBS and evaluation results were temporally available in dispersed Web sites that after some years are no longer available. Even with recent papers it is nearly impossible to reproduce and verify the evaluation results that their authors claimed.

However, all previous work on evaluation of DLBS provides a solid foundation to accomplish our objectives towards the correct design of evaluations and the implementation of specific software components for the execution and analysis of these evaluations using the SEALS platform.

For the sake of conciseness, we will discuss only some relevant previous contributions starting with the first notorious attempt of building a framework for testing ABox reasoning. Largely inspired by the Wisconsin benchmark [13] for testing database management systems, the Lehigh University Benchmark (LUBM) is still the facto standard for testing ABox reasoning. LUBM provides a simple TBox with 43 classes and 32 properties encoded in OWL-Lite. This TBox describes Universities, their departments and some related activities. LUBM also includes a synthetic data generator for producing ABoxes of different sizes. A set of 14 predefined SPARQL queries has been specifically designed for measuring five different factors related to ABox reasoning capabilities.

LUBM was extended to provide some TBox reasoning evaluation support and to increase the complexity of the ABoxes generated. The UOBM [14] enriched the original TBox with new axioms that use most of OWL-DL constructors. In fact, UOBM provides two TBoxes, one in OWL-DL and one in OWL-Lite. The OWL-DL TBox has 69 classes and 43 properties, and the OWL-Lite TBox includes 51 classes and 43 properties. The ABox generator was also improved to provide higher connected Aboxes.

5.2 Results of the SEALS evaluations

In our setting, the standard input format is the OWL 2 language. We evaluate interoperability with the standard inference services:

- Class satisfiability;
- Ontology satisfiability;
- Classification;
- Logical entailment.

The last two are defined in the OWL 2 Conformance document, while the first two are extremely common tasks during ontology development, and are de facto standard tasks for DLBSs.

The performance criterion relates to the efficiency software characteristic from ISO-IEC 9126-1. We take a DLBS's performance as its ability to efficiently perform the standard inference services. We use the number of tests passed by a DLBS without parsing errors is a metric of a system's conformance to the relevant syntax standard. The number of inference tests passed by a DLBS is a metric of a system's ability to perform the standard inference services. An inference test is counted as passed if the system result coincides with a “gold standard". The evaluation must also provide informative data with respect to DLBS performance. The performance of a system is measured as the time the system needs to perform a given inference task. We also record task loading time to assess the amount of preprocessing used in a given system.

The testing data was used for evaluation of three DLBSs HermiT 1.2.2 4, FaCT++ 1.4.1 5 and jcel 0.8.0 6. HermiT is a reasoner for ontologies written using the OWL [15]. It is the first publicly-available OWL reasoner based on a novel hypertableau calculus which provides efficient reasoning capabilities. HermiT can handle DL Safe rules and the rules can directly be added to the input ontology in functional style or other OWL syntaxes supported by the OWL API.

FaCT++ [16] is the new generation of the well-known FaCT OWL-DL reasoner. FaCT++ uses the established FaCT algorithms, but with a different internal architecture. Additionally, FaCT++ is implemented using C++ in order to create a more efficient software tool, and to maximise portability. jcel is a reasoner for the description logic EL+. It is an OWL 2 EL reasoner implemented in Java.
The results demonstrated:

- **Class satisfiability**: FaCT++ clearly outperformed HermiT on the most of the reasoning tasks. Most errors for both FaCT++ and HermiT were related to the datatypes not supported in the systems. The evaluation tasks proved to be challenging enough for the systems. Thus, 16 and 30 evaluation tasks respectively were not solved in the given time frame. The relatively poor HermiT performance can be explained taking into account the small number of very hard tasks where FaCT++ was orders of magnitude more efficient.
- **Ontology satisfiability**: Most FaCT++ errors were related to not supported datatypes. There were several description logic expressivity related errors such as NonSimpleRoleInNumberRestriction. There also was several syntactic related errors where FaCT++ was unable to register a role or a concept.
- **Classification**: Most errors were related to the datatypes not supported in FaCT++ system. There were several description logic expressivity related errors such as NonSimpleRoleInNumberRestriction. There also were several syntax related errors where FaCT++ was unable to register a role or a concept.
- **Logical entailment**: The HermiT time was influenced by small number of very hard tasks. FaCT++ demonstrated a big number of false and erroneous results. In conclusion, The DL reasoners designed for less expressive subsets of the OWL 2 language not surprisingly demonstrated superior performance illustrating trade off between expressivity and performance. Most of the errors demonstrated by systems designed to work for more expressive language subsets were related to non supported language features.

6. ONTOLOGY MATCHING EVALUATION CAMPAIGN

The SEALS Evaluation Campaign for Ontology Matching Tools has been coordinated with the Ontology Alignment Evaluation Initiative (OAEI) 2010 campaign. The first SEALS/OAEI campaign included three scenarios to evaluate the compliance of tools results with respect to expected alignment results.

6.1 Previous evaluations

Since 2004, a group of researchers on ontology matching has organized annual evaluation campaigns for evaluating matching tools. This initiative is identified as Ontology Alignment Evaluation Initiative[9] (OAEI) campaigns. The main goal of the

[9] http://oaei.ontologymatching.org/

OAEI is to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies.

In these campaigns, participants are invited to submit the results of their systems to organizers, who are responsible for running evaluation scripts and delivering the evaluation result interpretations. Since 2010, OAEI is being coordinated with the SEALS project and the plan is to integrate progressively the SEALS infrastructure within the OAEI campaigns. A subset of the OAEI tracks has been included in the new SEALS modality. Participants are invited to extend a web service interface and deploy their matchers as web services, which are accessed in an evaluation experiment. This setting enables participants to debug their systems, run their own evaluations and manipulate the results immediately in a direct feedback cycle.

6.2 Results of the SEALS evaluations

In OAEI 2010[10], the following tracks and data sets have been selected for the SEALS evaluations:

The **benchmark test** aims at identifying the areas in which each matching algorithm is strong and weak. The test is based on one particular ontology dedicated to the very narrow domain of bibliography and a number of alternative ontologies of the same domain for which alignments are provided.

The **anatomy test** is about matching the Adult Mouse Anatomy (2744 classes) and the NCI Thesaurus (3304 classes) describing the human anatomy. Its reference alignment has been generated by domain experts.

The **conference test** consists of a collection of ontologies describing the domain of organising conferences. Reference alignments are available for a subset of test cases.

For the three data sets in the SEALS modality, compliance of matcher alignments with respect to the reference alignments is evaluated. In the case of Conference, where the reference alignment is available only for a subset of test cases, compliance is measured over this subset. The most relevant measures are precision (true positive/retrieved), recall (true positive/expected) and f-measure (aggregation of precision and recall).

The campaign had 15 participants in 2010 [17]: AgrMaker, AROMA, ASMOV, BLOOMS, CODI, Ef2Match, Falcon-AO, GeRoMeSMB, LNR2, MapPSO, NBJLM, ObjectRef, RiMOM, SOBOM and TaxoMap. Regarding the SEALS tracks, 11 participants have registered their results for Benchmark, 9 for Anatomy and 8 for Conference.

In the benchmark track, two systems are ahead: ASMOV and RiMOM, with AgrMaker as close follower, while SOBOM, GeRoMeSMB and Ef2Match, respectively, had presented intermediary values of precision and recall. In the 2009 campaign[11], Lily and ASMOV were ahead, with Aflood and RiMOM as followers, while GeRoME, AROMA, DSSim and AgrMaker had intermediary performance. The same group of best matchers has been presented in both campaigns. In general, the systems have improved their performance since last year: ASMOV and RiMOM improved their overall performance, AgrMaker and SOBOM have significantly improved their recall while MapPSO and GeRMeSBM improved precision. AROMA has significantly decreased in recall, for the three groups of tests. There is no unique set of systems ahead for all cases, what indicates that systems exploiting different features of ontologies perform accordingly to the features of each test cases.

In the Anatomy track [17], for the F-measure evaluation, AgreementMaker is followed by three participants (Ef2Match, NBJLM and SOBOM) that clearly favour precision over recall. Notice that these systems obtained better scores or scores that are similar to the results of the top systems in the previous years. One explanation can be seen in the fact that the organizers of the track made the reference alignment available to the participants. More precisely, participants could at any time compute precision and recall scores via the SEALS services to test different settings of their algorithms. This allows to improve a matching system in a direct feedback cycle.

In the Conference track [17], the matcher with the highest average F-measure (62%) is CODI which did not provide graded confidence values. Other matchers are very close to this score (e.g. ASMOV with F-Measure 0.60, Ef2Match with F-Measure 0.60, Falcon with F-Measure 0.59). However, we should take into account that this evaluation has been made over a subset of all alignments (one fifth).

In conclusion, the new technology introduced in the OAEI affected both tool developers and organizers to a large degree and has been accepted positively on both sides. For the next campaign, we plan to measure runtime and memory consumption, which cannot be correctly measured because a controlled execution environment is missing. The same holds for the reproducibility of the results. We also plan to integrate additional metrics and visualization components. Finally, we will try to find more well suited data sets to be used as test suites in the platform, what includes the development of a test generator that allows a controlled automatic test generation of high quality data sets.

7. SEMANTIC SEARCH EVALUATION CAMPAIGN

7.1 Previous evaluations

State-of-the-art semantic search approaches are characterised by their high level of diversity both in their features as well as their capabilities. Such approaches employ different styles for accepting the user query (e.g., forms, graphs, keywords) and apply a range of different strategies during processing and execution of the queries. They also differ in the format and content of the results presented to the user. All of these factors influence the user's perceived performance and usability of the tool. This highlights the need for a formalised and consistent evaluation which is capable of dealing with this diversity. It is essential that we do not forget that searching is a user-centric process and that the evaluation mechanism should capture the usability of a particular approach.

In previous evaluation efforts, Kaufmann evaluated four approaches to querying ontologies [18]. Three were based on natural language input (with one employing a restricted query formulation grammar); the fourth employed a formal query approach which was hidden from the end user by a graphical query interface. A comprehensive usability study was conducted which focused on comparing the different query languages employed by the tools. It was shown that users preferred approaches based around full natural language sentences to all other formats and interfaces. It was also noted that users favour query languages and interfaces in which they can naturally communicate their information need without restrictions on the

[10] http://oaei.ontologymatching.org/2010/

[11] http://oaei.ontologymatching.org/2009/

grammar used or having to rephrase their queries. Users were also found to express more semantics (e.g., relations between concepts) using full sentences rather than keywords.

Another work evaluated a “hybrid search" approach [19]. Their search approach consisted of an intelligent combination of keyword-based search and semantically motivated knowledge retrieval. To assess the effectiveness and the performance of the approach, an in vitro evaluation was conducted to compare it against keyword-based alone and ontology-based alone searching approaches. Additionally, the authors conducted an in vivo evaluation which involved 32 subjects who gave their opinion and comments regarding the efficiency, effectiveness, and satisfaction of the system. In both cases, the hybrid approach was observed to be superior.

The goal of the evaluation was to create a consistent and standard evaluation that can be used for assessing and comparing the strengths and weaknesses of Semantic Search approaches. This allows tool adopters to select appropriate tools and technologies for their specific needs and helps developers identify gaps and limitations with their own tools which will facilitate improving them. Furthermore, the evaluation outcomes identify new requirements of search approaches with the aim of more closely matching users' needs.

7.2 Results of the SEALS evaluations

The evaluation of each tool is split into two complementary phases: the Automated Phase and the User-in-the-loop Phase.

The user-in-the-loop phase comprises a series of experiments involving human subjects who are given a number of tasks (questions) to solve and a particular tool and ontology with which to do it. The subjects in the user-in-the-loop experiments are guided throughout the process by bespoke software which is responsible for presenting the questions and gathering the results and metrics from the tool under evaluation. Two general forms of metrics are gathered during such an experiment. The first type of metrics are directly concerned with the operation of the tool itself such as time required to input a query, and time to display the results. The second type is more concerned with the `user experience' and is collected at the end of the experiment using a number of questionnaires. The first is the System Usability Scale (SUS) questionnaire [20]. The test consists of ten normalized questions and covers a variety of usability aspects, such as the need for support, training, and complexity and has proven to be very useful when investigating interface usability.

We developed a second, extended, questionnaire which includes further questions regarding the satisfaction of the users. This encompasses the design of the tool, the input query language, the tool's feedback, and the user's emotional state during the work with the tool. An example of a question used is `The query language was easy to understand and use' with answers represented on a scale from `disagree' to `agree'.

Finally, a demographics questionnaire collected information regarding the participants. The outcome of these two phases will allow us to benchmark each tool both in terms of its raw performance but also the ease with which the tool can be used. Indeed, for semantic search tools, it could be argued that this latter aspect is the most important.

The Automated Phase used EvoOnt[12]. This is a set of software ontologies and data exchange format based on OWL. It provides the means to store all elements necessary for software analyses including the software design itself as well as its release and bugtracking information. For scalability testing it is necessary to use a data set which is available in several different sizes. In the current campaign, it was decided to use sets of sizes 1k, 10k, 100k, 1M, 10M triples. The EvoOnt data set lends itself well to this since tools are readily available which enable the creation of different ABox sizes for a given ontology while keeping the same TBox. Therefore, all the different sizes are variations of the same coherent knowledge base.

The test questions for the automated phase ranged in their level of complexity including simple ones like \Does the class x have a method called y?" and more complex ones like \Give me all the issues that were reported in the project x by the user y and that are _xed by the version z?".

The main requirement for the user-in-the-loop dataset is that it be from a simple and understandable domain for which users are able to reformulate the questions into the respective query language. We used a geographical data set, supplying both English questions, and corresponding logical queries.

Five tools participated in the campaign: K-Search, Ginseng, NLP-Reduce, Jena Arq 2.8.2 and PowerAqua. Full results and analyses can be found in SEALS deliverable D13.3. The most unexpected outcome of the automated phase was the failure of many of the participating tools to load even the smallest ontology. The EvoOnt ontologies have certain interesting characteristics which, although complex, are valid and commonly found on the Semantic Web. One of these characteristics is importing of external ontologies from the web. Also, the ontologies include orphan object and datatype properties, and finally some concepts have cyclic relations with concepts in remote ontologies. This informs the tools' scalability, conformance with standards and suitability to the Semantic Web. Unfortunately, many of the participating tools were not able to cope with these standards.

Here, we focus solely on usability results from the user-in-the-loop phase. According to the ratings of SUS scores, none of the four participating tools fell in either the best or worst category. Only one of the tools had a `Good' rating with a SUS score of 72.25, other two tools fell in the `Poor' rating while the last one was classified as `Awful'.

In conclusion, there is still work to be done to ensure semantic search tools can load or use as wide a range of ontologies and data sets as possible. The usability phase identified a number of features that end users would like: a hybrid approach to browsing the ontology and creating queries which would combine both a visual representation of the underlying ontology and natural language input; better feedback regarding the processing state of the tool (e.g., to distinguish between a query failure and an empty result set); improved result set management (sorting, filtering, ability to use as the target of a subsequent query, etc.) and the inclusion of `related' information (possibly drawn from other data sets).

8. SEMANTIC WEB SERVICES EVALUATION CAMPAIGN

8.1 Previous evaluations

The evaluation of Semantic Web Services is currently being pursued by a few initiatives using different evaluation methods.

[12] http://www.ifi.uzh.ch/ddis/evo/

The SWS Challenge[13] (SWSC) aims at providing a forum for discussion of SWS approaches based on a common application base. The approach is to provide a set of problems that participants solve in a series of workshops. In each workshop, participants self-select which scenario (e.g. discovery, mediation or invocation) and problems they would like to solve. Solutions to the scenarios provided by the participants are manually verified by the Challenge organizing committee.

The Semantic Service Selection[14] (S3) contest is about the retrieval performance evaluation of matchmakers for Semantic Web Services. S3 is a virtual and independent contest, which runs annually since 2007. It provides the means and a forum for the joint and comparative evaluation of publicly available Semantic Web service matchmakers over given public test collections. The organizers of S3 provide the SME2 evaluation system, which has a number of metrics available and provides comparison results in graphical format. They have also been involved in the development of the OWLS-TC and SAWSDL-TC test collections.

The Web Service Challenge[15] (WSC) runs annually since 2005 and provides a platform for researchers in the area of web service composition that allows them to compare their systems and exchange experiences. Starting from the 2008 competition, the data formats and the contest data are based on the OWL for ontologies, WSDL for services, and WSBPEL for service orchestrations. In 2009, services were annotated with non-functional properties. The Quality of Service of a Web Service is expressed by values expressing its response time and throughput. The WSC awards the most efficient system and also the best architectural solution. The contestants should find the composition with the least response time and the highest possible throughput.

Although these initiatives have succeeded in creating an initial evaluation community in this area, they have been hindered by the difficulties in creating large-scale test suites and by the complexity of manual testing to be done. In principle, it is very important to create test datasets where semantics play a major role for solving problem scenarios; otherwise comparison with non-semantic systems will not be significant, and in general it will be very difficult to measure tools or approaches based purely on the value of semantics. Therefore, providing an infrastructure for the evaluation of SWS that supports the creation and sharing of evaluation artifacts and services, making them widely available and registered according to problem scenarios, using agreed terminology, can benefit evaluation participants and organizers.

The work performed in SEALS regarding SWS tools is based upon the Semantic Web Service standardization effort that is currently ongoing within the OASIS Semantic Execution Environment Technical Committee (SEE-TC). A Semantic Execution Environment (SEE) is made up of a collection of components that are at the core of a Semantic Service Oriented Architecture (SOA). These components provide the means for automating many of the activities associated with the use of Web Services, thus they will form the basis for creating the SWS plugin APIs and services for SWS tools evaluation.

8.2 Results of the SEALS evaluations

Currently, we focus on the SWS discovery activity, which consists of finding Web Services based on their semantic descriptions. Tools for SWS discovery or matchmaking can be evaluated on retrieval performance, where for a given goal, i.e. a semantic description of a service request, and a given set of service descriptions, i.e. semantic descriptions of service offers, the tool returns the match degree between the goal and each service, and the platform measures the rate of matching correctness based on a number of metrics.

In SEALS we provide the SWS plugin API, available from the campaign website, that must be implemented by tool providers participating in the SWS tool evaluation. The SWS Plugin API has been derived from the SEE API and works as a wrapper for SWS tools, providing a common interface for evaluation. For our evaluation we used the OWLS-TC 4.0 test collection[16], which is intended to be used for evaluation of OWL-S matchmaking algorithms. The OWLS-TC4 version consists of 1083 semantic web services described with OWL-S 1.1, covering nine application domains (education, medical care, food, travel, communication, economy, weapons, geography and simulation). OWLS-TC4 provides 42 test queries.

The participating tools were four publicly available variants of OWLS-MX that were used for an experimental evaluation. Each OWLS-MX variant runs a different similarity measure algorithm. The full results are provided in deliverable D14.3 under the SEALS website. From our analysis, we find that public intermediate results and repeatability are important for studying the behaviour of the tools under different settings (not only best behaviour).

With respect to the datasets we noticed that all variants of OWLS-MX could retrieve all the same relevant services for a given reference set, provided that they were set with the same parameters. This might not be relevant for the sample result produced, which was not comprehensive. In fact, different variants of a tool can indeed perform very closely and would not usually compete in the same experiment with the same tuning. We also observed that the recall at number retrieved is equal to 1 for a high number of queries. This could indicate bias of the test suite against the tool, however, the tool providers argue that "it rather indicates that in turn the tools that are tested against the collection either adapt to the collection (as for example the adaptive matchmakers OWLS-MX3, iSeM, iMatcher do in S3) by themselves or have been to some extent manually optimised for the collection (though the problem with this is clearly the amount of effort required to change the matchmaker when the collection changes), or are simply as good as they are. In S3, it is known from developers that many different matchmakers have been in fact optimised for the OWLS-TC to some extent for service selection". Overall, it is important that the SWS community get engaged in creating non-biased datasets and provide alternative metrics, via SEALS. We have also noticed that some ontologies could not be read, probably affecting some results. Thus, it is important to introduce some validation procedure.

Overall, it is important that the SWS community get engaged in creating non-biased datasets and provide alternative metrics, via SEALS. We have also noticed that some ontologies could not be read, probably affecting some results. Thus, it is important to introduce some validation procedure.

9. CONCLUSIONS

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

[14] http://www-ags.dfki.uni-sb.de/~klusch/s3/index.html

[15] http://ws-challenge.georgetown.edu/wsc09/technical details.html

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

This paper has presented an overview of the first series of evaluation campaigns organized in the SEALS project for the five types of technologies covered in it: ontology engineering tools, ontology reasoning tools, ontology matching tools, semantic search tools, and semantic web services. 32 tools from all around the world were evaluated using common evaluation methods and test data. In some cases, we followed existing evaluation methods and used available test data; in other cases, we defined new evaluations methods and test data to enhance the evaluations performed in the evaluation campaigns.

We have established as a result of our experiences from the first evaluation campaigns that the chosen evaluation methodologies and test data are an appropriate basis for discovering useful evaluation results from the participating tools. In general, it can be seen that semantic tools are reaching maturity with respect to the key characteristics for their domains, and hence there is a real value to be had in comparative evaluation to guide tool selection, since different tools in the same domain still exhibit significant differences in implementation or functionality which are of importance in differing usage scenarios.

We aim in the second campaign to broaden the extent of involved tools in the evaluations, since this will improve the possibility to determine the current state of the art of the tools in the given domain, and how they compare to one another.

All the resources used in the SEALS Evaluation Campaigns as well as the results obtained in them will be publicly available through the SEALS Platform. This way, anyone interested in evaluating one of the technologies covered in the project will be able to do so, and to compare to others, with a small effort. The SEALS evaluation infrastructure will be open to all via the SEALS website, requiring only a simple preregistration in order to be able to access our Community Area[17]. Within the Community Area, there is the possibility to register a tool, describe it, upload it to SEALS and execute evaluations upon it, gaining immediately an insight into how it compares to the previously evaluated tools.

Our future plans are to extend the evaluations defined for the different types of technologies and conduct a second edition of the SEALS Evaluation Campaigns. This second Campaign is scheduled to begin in late 2011, and by the close of the SEALS project in early 2012 we will publish a second white paper on semantic tool evaluation which is intended for potential tool adopters, in order to guide them with respect to their choice of tools when seeking to benefit from the use of semantic technology within their systems and IT projects.

10. ACKNOWLEDGMENTS

The work described here has been supported by the EU project SEALS (FP7-ICT-238975).

11. REFERENCES

[1] R. García-Castro and F. Martín-Recuerda. D3.1 SEALS Methodology for Evaluation Campaigns v1. Technical report, SEALS Consortium, 2009.

[2] OntoWeb. Ontoweb deliverable 1.3: A survey on ontology tools. Technical report, IST OntoWeb, May 2002.

[3] P. Lambrix, M. Habbouche, and M. Pérez. Evaluation of ontology development tools for bioinformatics. Bioinformatics, 19(12):1564-1571, 2003.

[4] J. Angele and Y. Sure, editors. Proc. 1st International Workshop on Evaluation of Ontology-based Tools (EON2002), volume 62, Siguenza, Spain, September 2002.

[5] Y. Sure and O. Corcho, editors. Proc. 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), volume 87, Sanibel Island, Florida, USA, Oct 2003.

[6] R. García-Castro and A. Gómez-Pérez. RDF(S) interoperability results for semantic web technologies. International Journal of Software Engineering and Knowledge Engineering, 19(8):1083-1108, December 2009.

[7] R. García-Castro and A. Gómez-Pérez. Interoperability results for Semantic Web technologies using OWL as the interchange language. Web Semantics: Science, Services and Agents on the World Wide Web, 8:278-291, November 2010.

[8] R. García-Castro and A. Gómez-Pérez. Guidelines for benchmarking the performance of ontology management APIs. In Proc. 4th International Semantic Web Conference (ISWC2005), number 3729 in LNCS, pages 277-292, Galway, Ireland, November 2005.

[9] Yuanbo Guo, Zhengxiang Pan, and Je_ Hein. LUBM: A benchmark for OWL knowledge base systems. Web Semantics: Science, Services and Agents on the World Wide Web, 3(2-3):158-182, October 2005.

[10] L.Ryan. Scalability report on triple store applications. Technical report, SIMILE Project, November 2004.

[11] R. García-Castro. Benchmarking Semantic Web technology, volume 3 of Studies on the Semantic Web. IOS Press, Jan 2010.

[12] R. García-Castro, I. Toma, A. Marte, M. Schneider, J. Bock, and S. Grimm. D10.2. Services for the automatic evaluation of ontology engineering tools v1. SEALS Project, July 2010.

[13] David J. DeWitt. The wisconsin benchmark: Past, present, and future. In Jim Gray, ed., The Benchmark Handbookfor Database and Transaction Systems (2nd Edition). Morgan Kaufmann, 1993.

[14] L. Ma, Y. Yang, Z. Qiu, G. T. Xie, Y. Pan, and S. Liu. Towards a complete OWL ontology benchmark. In ESWC, pages 125-139, 2006.

[15] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. Journal of Web Semantics, 1(1):7-26, 2003.

[16] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2006), volume 4130 of Lecture Notes in Artificial Intelligence, pages 292-297. Springer, 2006.

[17] Jerome Euzenat et al. Results of the ontology alignment evaluation initiative 2010. In Proc. 5th ISWC workshop on ontology matching (OM), Shanghai (China), pages 1-35, 2010.

[18] Esther Kaufmann. Talking to the Semantic Web | Natural Language Query Interfaces for Casual End-Users. PhD thesis, Faculty of Economics, University of Zurich, Sept 2007.

[19] Ravish Bhagdev et al. Hybrid search: Effectively combining keywords and ontologybased searches. In Proc. 5th European Semantic Web Conference, LNCS, Berlin, June 2008.

[20] John Brooke. SUS: a quick and dirty usability scale. In Usability Evaluation in Industry, pages 189-194. Taylor and Francis, 1996.

[17] http://www.seals-project.eu/join-the-community