HAL
open science

Automating OAEI Campaigns (First Report)

Cassia Trojahn dos Santos, Christian Meilicke, Jérôme Euzenat, Heiner Stuckenschmidt

To cite this version:

Cassia Trojahn dos Santos, Christian Meilicke, Jérôme Euzenat, Heiner Stuckenschmidt. Automating OAEI Campaigns (First Report). Proc. 1st ISWC international workshop on evaluation of semantic technologies (iWEST), Nov 2010, Shanghai, China. No pagination. hal-00793287

HAL Id: hal-00793287

https://inria.hal.science/hal-00793287

Submitted on 22 Feb 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.

Automating OAEI Campaigns (First Report)

Cássia Trojahn[1], Christian Meilicke[2],
Jérôme Euzenat[1], Heiner Stuckenschmidt[2]

[1] INRIA & LIG, Grenoble, France
[2] University of Mannheim, Mannheim, Germany

Abstract. This paper reports the first effort into integrating OAEI and SEALS evaluation campaigns. OAEI is an annual evaluation campaign for ontology matching systems. The 2010 campaign includes a new modality in coordination with the SEALS project. This project aims at providing standardized resources (software components and data sets) for automatically executing evaluations of typical semantic web tools, including ontology matching tools. A first version of the software infrastructure is based on a web service interface wrapping the functionality of the matching tool to be evaluated. In this setting, the evaluation results can be visualized and manipulated immediately in a direct feedback cycle. We describe how parts of the OAEI 2010 evaluation campaign have been integrated into the SEALS software infrastructure. In particular, we discuss technical and organizational aspects related to the use of the new technology for both participants and organizers of the OAEI.

1 Introduction

The Ontology Alignment Evaluation Initiative[3] (OAEI) is a coordinated international initiative that organizes the evaluation of ontology matching systems [6]. The main goal of OAEI is to compare systems and algorithms on the same basis and to allow anyone for drawing conclusions about the best matching strategies. The ambition is that from such evaluations, tool developers can learn and improve their systems. The OAEI annual campaign provides the evaluation of matching systems on consensus test cases, which are organized by different groups of researchers. OAEI evaluations have been carried out since 2004.

Although OAEI has been the basis for ontology matching evaluation over the last years, additional efforts have to be made in order to catch up with the growth of ontology matching technology, specially in two main directions: large scale evaluation and automation of the evaluation process. The SEALS project[4] aims at providing standardized data sets, evaluation campaigns for typical semantic web tools and, in particular, a software infrastructure for automatically executing evaluations. One of the five semantic areas covered by SEALS is ontology matching. The SEALS infrastructure will allow developers to run their tools

[3] `http://oaei.ontologymatching.org`
[4] Semantic Evaluation at Large Scale `http://about.seals-project.eu/`

Proceedings of the International Workshop on Evaluation of Semantic Technologies (IWEST 2010). Shanghai, China. November 8, 2010.

on an execution environment in both the context of an evaluation campaign and on their own for a formative evaluation of their tool versions.

OAEI and SEALS are closely coordinated and the plan is to integrate progressively the SEALS infrastructure within the OAEI campaigns. The 2010 OAEI campaign is the first effort in this direction. A subset of the OAEI tracks have been included in this new modality. Participants are invited to extend a web service interface[5] 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. On the other hand, runtime and memory consumption cannot be correctly measured because a controlled execution environment is missing. Further versions of the SEALS infrastructure will include the deployment of tools in such a controlled environment.

In this paper, we report the first efforts on integrating OAEI campaigns into the SEALS infrastructure. We describe the preparation of the evaluation campaign as well as we comment on how the evaluation itself is conducted, taking into account its partial automation. Furthermore, we present the SEALS evaluation service that we have developed for this purpose and show how it has been used at hand of a concrete example.

The rest of the paper is structured as follows. We firstly present the evaluation design of the 2010 evaluation campaign (§2). We comment on evaluation workflows, data sets and criteria and metrics specified for this evaluation. Secondly, we detail how the designed evaluation is being conducted (§3). Third, we present the overview of the main software components of the SEALS infrastructure (§4) together with an example of running evaluations (§5). Preliminary results of the campaign are then presented (§6). Finally, we comment on the lessons learned (§7) and conclude the paper (§8).

2 Evaluation Design

The design of an evaluation campaign is conducted previously to the execution of the campaign itself. It involves to specify the data sets, criteria and metrics to be considered in the campaign, as well as how the several components (matchers, test providers, evaluators, etc.) interact in an evaluation experiment, i.e., the evaluation workflow.

2.1 Evaluation workflow

An alignment can be characterized as a set of pair of entities (e and e'), coming from the ontologies to be aligned (o and o'), related by a particular relation (r) together with some confidence measure (n) expressing a degree of trust in the fact that the relation holds [2, 1, 3]. From this characterization it is possible to ask any alignment method to output an alignment, given: (i) two ontologies to be

[5] `http://alignapi.gforge.inria.fr/tutorial/tutorial5/`

aligned; (ii) partial input alignment (possibly empty); and (iii) a characterization of the wanted alignment (e.g. one-to-one vs. many-to-many alignments).

The quality of the generated alignment can be assessed regarding different criteria. Figure 1 shows the evaluation workflow representing an OAEI evaluation experiment, where several matchers are evaluated. The first step of the workflow is to retrieve from a database the test cases to be considered in such an evaluation, where each test case consists of the two ontologies to be matched and the corresponding reference alignment. Next, each matching system performs the matching, taking as input parameters the two ontologies o and o', and generates the alignment A using a certain set of resources and parameters. An evaluation component receives this alignment and computes a (set of) quality measure(s) m – typically precision and recall – by comparing it to the reference alignment R. Finally, each result interpretation is stored into the result database.

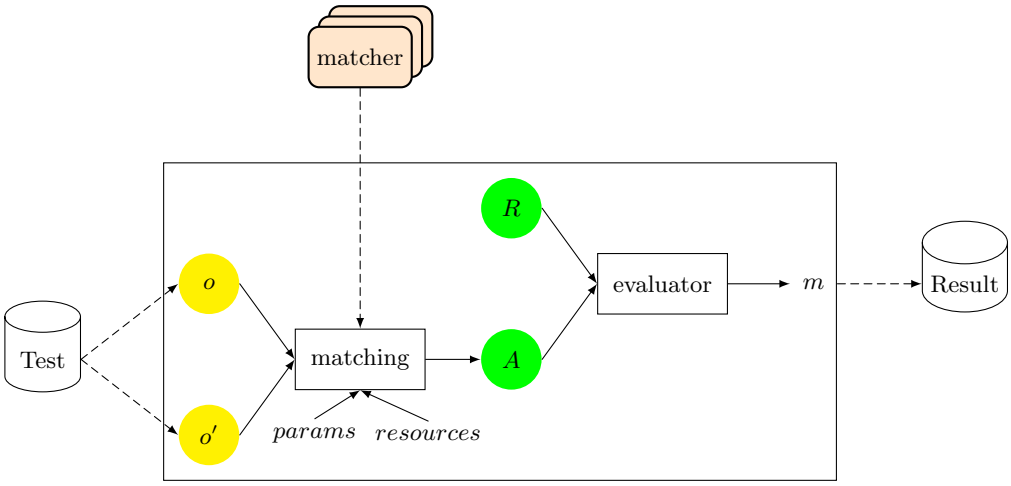

Fig. 1. OAEI typical evaluation workflow.

This workflow represents a typical OAEI evaluation workflow. However, for some data sets, which have no complete reference alignments, extensions for this typical workflow have been designed. Usually, in such cases, the user has the role of *evaluator* and alternative approaches, such as manual labeling, data mining and logical reasoning, are applied for supporting the evaluation task. For instance, according to Figure 1, for each test case, the available matchers are executed and their generated alignments (A) are stored into the database. This content is then used later by data mining techniques, whose results will be finally analysed by the user.

In previous OAEI campaigns, this workflow has been realized as follows: the required test cases have been made available to the participants for download and posteriorly used by the participants to generate the matching results with their tool. The results have then been submitted to the OAEI organizers who used evaluation scripts to apply measures and store the results.

2.2 OAEI data sets

OAEI data sets have been extended and improved over the years. In the OAEI 2010 campaign, the following tracks and data sets have been selected:

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.

The directories and thesauri test cases propose web directories (matching website directories like open directory or Yahoo's), thesauri (three large SKOS subject heading lists for libraries) and generally less expressive resources.

The instance matching test cases aim at evaluating tools able to identify similar instances among different data sets. It features web data sets, as well as a generated benchmark.

Anatomy, *Benchmark* and *Conference* have been included in the SEALS evaluation modality. The reason for this is twofold: on the one hand these data sets are well known to the organizers and have been used in many evaluations contrary to the test cases of the instance data sets, for instance. On the other hand these data sets come with a high quality reference alignment which allows for computing the compliance based measures, such as precision and recall.

2.3 Evaluation criteria and metrics

The diverse nature of OAEI data sets, specially in terms of the complexity of test cases and presence/absence of (complete) reference alignments, requires to use different evaluation measures. 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). These metrics are also partially considered or approximated for the other data sets which are not included in the SEALS modality (standard modality).

For *Conference*, alternative evaluation approaches have been applied. These approaches include manual labeling, alignment coherence [7] and correspondence patterns mining. These approaches require a more deep analysis from experts than traditional compliance measures. For the first version of the evaluation service, we concentrate on the most important compliance based measures because

they do not require a complementary step of analyse/interpretation from experts, which is mostly performed manually and outside an automatic evaluation cycle. However, such approaches will be progressively integrated into the SEALS infrastructure.

Nevertheless, for 2010, the generated alignments are stored in the results database (as detailed in §4) and can be retrieved by the organizers easily. It is thus still possible to exploit alternative evaluation techniques subsequently, as it has been done in the previous OAEI campaigns.

All the criteria above are about alignment quality. A useful comparison between systems also includes their efficiency, in terms of runtime and memory consumption. The best way to measure efficiency is to run all systems under the same controlled evaluation environment. In previous OAEI campaigns, participants have been asked to run their systems on their own and to inform about the elapsed time for performing the matching task. Using the web based evaluation service, runtime cannot be correctly measured due the fact that the systems run in different execution environments and, as they are exposed as web services, there are potential network delays.

3 Evaluation Process

Once the evaluation design has been specified, the evaluation campaign takes place in four main phases:

Preparatory phase ontologies and alignments are provided to participants, which have the opportunity to send observations, bug corrections, remarks and other test cases;

Preliminary testing phase participants ensure that their systems can load the ontologies to be aligned and generate the alignment in the correct format (the Alignment API format [3]);

Execution phase participants use their algorithms to automatically match the ontologies;

Evaluation phase the alignments provided by the participants are evaluated and compared.

The four phases are the same for both standard and SEALS modality. However, different tasks are required to be performed by the participants of each modality. In the *preparatory phase*, the data sets have been published on web sites and could be downloaded as zip-files. In the future, it will be possible to use the SEALS portal to upload and describe new data sets. In addition, the test data repository supports versioning, which is an important issue regarding bug fixes and improvements that have taken place over the years.

In the *phase of preliminary testing*, the SEALS evaluation service pays off in terms of reduced effort. In the past years, participants submitted their preliminary results to the organizers, who analyzed them semi-automatically, often detecting problems related to the format or to the naming of the required results

files. Via a time-consuming communication process these problems have been discussed with the participants. It is now possible to check these and related issues automatically (as detailed in §5).

In the *execution phase*, standard OAEI participants run their tools on their own machines and submit the results via mail to the organizers, while SEALS participants run their tools via web service interfaces. They get a direct feedback on the results and can also discuss and analyse this feedback in their results paper[6]. Prior to the hard deadlines, for many of the data sets, results could not be delivered in the past to participants by the organizers in time.

Finally, in the *evaluation phase*, organizers are in charge of evaluating the received alignments. For the SEALS modality, this effort has been minimized due the fact that the results are automatically computed by the services in the infrastructure, as detailed in the next section.

4 Evaluation Service Architecture

The evaluation service is composed of three main components: a web user interface, a BPEL workflow and a set of web services. The web user interface is the entry point to the application. This interface is deployed as a web application in a Tomcat application-server behind an Apache web server. It invokes the BPEL workflow, which is executed on the ODE[7] engine. This engine runs as a web application inside the application server.

The BPEL process accesses several services that provide different functionalities:

- *validation service* ensures that (a) the matcher web service specified via its endpoint (URL) is available; (b) this service implements correctly the interface we have specified; and (c) the matcher generates an alignment in the correct format (the validation service uses two simple ontologies in order to test if the matcher generates alignments in the correct format). If it is not the case, an output error message is generated to the user. This validation is done prior to any evaluation.
- *redirect service* is used to redirect the request for running a matching task to the matcher service endpoint.
- *test iterator service* is responsible for iterating over test cases and providing a reference to the required files. These files are the source ontology, the target ontology and the reference alignment. All the operations of this service make use of the SEALS test data repository.
- *evaluation service* computes measures such as precision and recall for evaluating the alignments generated by the matching system.
- *result service* is used for storing evaluation results in a relational database.

[6] Notice that each participant in the OAEI, independently of the modality, has to write a paper that contains a system description and an analysis of results from the point of view of the system developer.

[7] ODE BPEL Engine `http://ode.apache.org/`

The user can start an evaluation by specifying the web service endpoint via the web interface. This data is then forwarded to BPEL as an input parameter. The complete evaluation workflow is executed as a series of calls to the services listed above. The specification of the web service endpoint becomes relevant for the invocation of the validation and redirect services. They implement internally web service clients that connect to the URL specified in the web user interface.

Test and result services require to access additional data resources. For test data, the test web service accesses the SEALS repository, extracts the relevant information and forwards the URLs of the required documents (source and target ontologies and reference alignment) via the redirect service to the matcher, which is currently evaluated. The result web service uses a connection to the database to store the results for each execution of an evaluation workflow. For visualizing and manipulating the stored results, an OLAP (Online Analytical Processing) application is available. Results can be re-accessed at any time e.g., for comparing different tool versions against each other.

5 Running an Evaluation

For illustrating a complete evaluation cycle, we have extended the Anchor-Flood system [8] with the web service interface[8]. This system has participated in the two previous OAEI campaigns and is thus a typical evaluation target. The current version of the web application described in the following is available at `http://seals.inrialpes.fr/platform/`. In order to start an evaluation, one must specify the URL of the matcher service, the class implementing the required interface and the name of the matching system to be evaluated (Figure 2). Three of the OAEI data sets have been selected, namely Anatomy, Benchmark and Conference. In this example, we have used the conference test case.

Submitting the form data, the BPEL workflow is invoked. It first validates the specified web service as well as its output format. In case of a problem, the concrete validation error is displayed to the user as direct feedback. In case of a successfully completed validation, the system returns a confirmation message and continues with the evaluation process. Every time an evaluation is conducted, results are stored under the enpoint address of the deployed matcher (Figure 3).

The results are displayed as a table (Figure 4), when clicking on one of the three evaluation IDs in Figure 3. The results table is (partially) available while the evaluation itself is still running. By reloading the page from time to time, users can see the progress of an evaluation that is still running. In the results table, for each test case, precision and recall are listed. Moreover, a detailed view on the alignment results is available (Figure 5), when clicking on the alignment icon in Figure 4. This detailed view lists those correspondences, that (a) have been generated and are in the reference alignment (true positives), that (b) have been generated but are not in the reference alignment (false positives), and that (c) have not been generated but are in the reference alignment (false negatives).

[8] Available at `http://mindblast.informatik.uni-mannheim.de:8080/sealstools/aflood/matcherWS?wsdl`

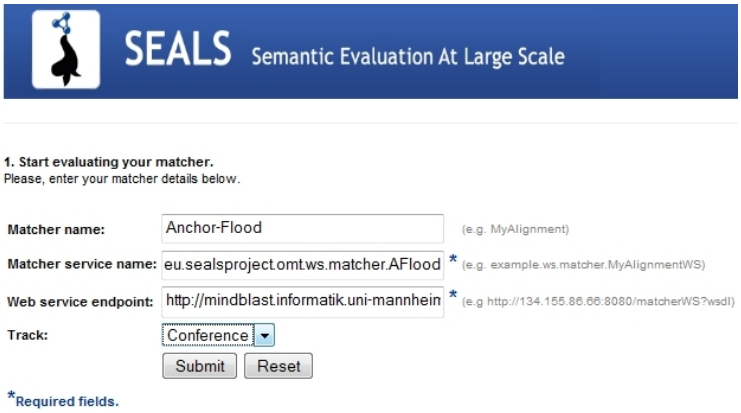

Fig. 2. Specifying a matcher endpoint as evaluation target.

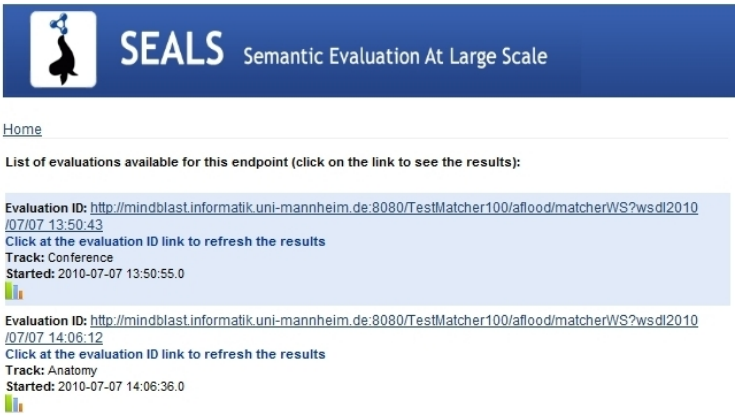

Fig. 3. Listing of available evaluation results.

The user can visualize the results in an OLAP application by clicking on the plot figure in Figure 3, in a similar way from what is shown in Figure 6, but in a setting where only the results of his/her system are shown. Furthermore, organizers have a similar tool for accessing the results registered for the campaign as well as all evaluations being carried out in the evaluation service (even the evaluation executed for testing purposes).

6 Preliminary Results

The OAEI 2010 campaign has counting with 15 participants [5] (16 participants in 2009 [4]). Regarding the SEALS tracks, 11 participants have registered their results for Benchmark, 9 for Anatomy and 8 for Conference. Some participants in Benchmark have not participated in Anatomy or Conference and vice-versa.

Test	Precision	Recall	Status	Alignment
cmt-conference	0.30	0.38	completed	
cmt-confOf	0.45	0.31	completed	
cmt-edas			not completed	
cmt-ekaw			not completed	
cmt-iasted			not completed	
cmt-sigkdd			not completed	
conference-confOf			not completed	
conference-edas			not completed	
conference-ekaw			not completed	

Fig. 4. Display results of an evaluation.

Correct correspondences

Paper = Paper
ConferenceChair = ConferenceChair
Review = Review
Conference = Conference
Person = Person

Back to top

Incorrect correspondences

writtenBy = isWrittenBy
reviewsPerPaper = relatedToPaper
date = endDate
email = hasEmail

Back to top

Missing correspondences

assignedTo = isReviewedBy
hasAuthor = isWrittenBy

Fig. 5. Detailed view on an alignment.

Figure 6 shows some preliminary results. The values of precision, recall and f-measure are the average of the results for all test cases in each track.

For the *benchmark* track, two systems are ahead: ASMOV and RiMOM, with AgrMaker as close follower, while SOBOM, GeRMeSMB and Ef2Match, respectively, achieve intermediary values of precision and recall. For *anatomy*, AgrMaker has generated the best alignments with respect to f-measure. This system is followed by three participants (Ef2Match, NBJLM, SOBOM) that share a very similar characteristic regarding precision and recall. Finally, for *conference*, the matcher with the highest average f-measure was CODI, with Falcon, Ef2Match and ASMOV as followers.

There is no unique set of systems ahead for all three tracks, what clearly demonstrates that systems exploiting different features of ontologies perform accordingly to the features of each test cases. These preliminary results will be discussed in the fifth Ontology Matching Workshop collocated with ISWC, in Shanghai, China[9], and they can be found at `http://oaei.ontologymatching.org/2010/results/`. The complete analysis of the results will be available soon after the workshop in its web site.

7 Lessons Learned

The new technology introduced in the OAEI affected both tool developers and organizers to a large degree. In the following we highlight some of the outcomes and describe the lessons we learned from the experiences made so far.

As already argued, implementing the web service interfaced requires some effort on side of the tool developer. We stayed in contact with some of the tool developers during this process and observed that the time required for implementing the interface varied between several hours and several days depending

[9] `http://om2010.ontologymatching.org`

Evaluation campaign (OAEI) results

				Measures		
Track	Tool	EvaluationID	Test	Precision	Recall	FMeasure
−All Tracks	+All Tools	+All IDs	+All Tests	.839	.627	.718
Anatomy Testsuite	−All Tools	+All IDs	+All Tests	.907	.722	.804
	AgrMaker	+All IDs	+All Tests	.903	.853	.877
	AROMA	+All IDs	+All Tests	.769	.683	.724
	ASMOV	+All IDs	+All Tests	.799	.772	.785
	BLOOMS	+All IDs	+All Tests	.954	.731	.828
	CODI	+All IDs	+All Tests	.968	.651	.779
	Ef2Match	+All IDs	+All Tests	.952	.781	.858
	GeRMeSMB	+All IDs	+All Tests	.884	.307	.456
	NBJLM	+All IDs	+All Tests	.919	.791	.850
	SOBOM	+All IDs	+All Tests	.944	.778	.853
	TaxoMap	+All IDs	+All Tests	.924	.743	.824
Benchmark Testsuite	−All Tools	+All IDs	+All Tests	.887	.629	.736
	AgrMaker	+All IDs	+All Tests	.956	.804	.873
	AROMA	+All IDs	+All Tests	.865	.460	.601
	ASMOV	+All IDs	+All Tests	.977	.855	.912
	CODI	+All IDs	+All Tests	.745	.445	.557
	Ef2Match	+All IDs	+All Tests	.966	.637	.768
	Falcon	+All IDs	+All Tests	.752	.624	.682
	GeRMeSMB	+All IDs	+All Tests	.935	.636	.757
	MapPSO	+All IDs	+All Tests	.641	.573	.605
	RiMOM	+All IDs	+All Tests	.981	.820	.893
	SOBOM	+All IDs	+All Tests	.927	.701	.799
	TaxoMap	+All IDs	+All Tests	.853	.351	.497
Conference Testsuite	−All Tools	+All IDs	+All Tests	.483	.608	.538
	AgrMaker	+All IDs	+All Tests	.506	.672	.578
	AROMA	+All IDs	+All Tests	.360	.495	.417
	ASMOV	+All IDs	+All Tests	.349	.732	.473
	CODI	+All IDs	+All Tests	.874	.507	.642
	Ef2Match	+All IDs	+All Tests	.506	.653	.571
	Falcon	+All IDs	+All Tests	.605	.595	.600
	GeRMeSMB	+All IDs	+All Tests	.351	.523	.420
	SOBOM	+All IDs	+All Tests	.291	.640	.400

Fig. 6. Using OLAP for results visualization.

on the technical skills of each developer. We also observed that the first version of the provided tutorial contained some unclear information resulting in problems for some participants. From the feedback of the developers, we have improved the tutorial. Another typical problem is related to the fact that some tool developer had only restricted access to a machine that is available from the Internet. These problems could finally be solved, however, system administrators of the particular company or research institute should be contacted early.

Once technical problems had been solved, the evaluation service has been used by some of the participants in the phase of preliminary testing extensively.

Obviously, the direct feedback of the evaluation service has supported the process of a formative evaluation well. Other participants used the service only for submitting their final results.

Regarding the evaluation service performance, during the first weeks the runtime performance was suboptimal. We solved the underlying problems finally. These problems might have been the reason for some participants to abandon from the use during the first weeks. Once the problems have been solved, we contacted each participant in order to explain the problems and they started to use the system again.

On side of the organizers, the evaluation service reduced the effort of checking the formal correctness of the results to a large degree. In the past, it was required to communicate many of the problems in a time-consuming multilevel process. Typical examples are invalid xml, missing or incorrect namespace information, unsupported types of relations in generated alignments, incorrect directory structure and an incorrect naming style used for the submissions. All of these problems are now directly forwarded to the tool developer in an error message or in a preliminary result interpretation that does not fit with the expectations.

Moreover, the organizers could analyse the results so far submitted at any time and had an overview on the participants using the systems. However, while some analysis methods are already available, a number of specific services and operations is still missing. The graphical support of the OLAP visualisation does, for example, not support the generation of precision and recall graphs frequently used by OAEI organizers. In particular, evaluation and visualisation methods specific for ontology matching are not supported. However, most of these operations are already implemented in the Alignment API and will be made available in the future.

8 Final Remarks

This paper has reported the first efforts in integrating the SEALS evaluation service in OAEI evaluation campaigns. A preliminary version of this service has been exposed via a web service interface. For that reason, participants are asked to make available their tools as web services, which will be accessed in the evaluation experiment. The resulting approach offers the minimal requirements needed to execute a complete cycle of evaluation. The major benefit of this approach is to allow developers to debug their systems, run their own evaluations, and manipulate the results immediately in a direct feedback cycle. As a limitation, runtime and memory consumption cannot be correctly measured because there is no a controlled execution environment. Another important drawback is related to the missing reproducibility of the generated results.

In a second development iteration matching tools will be deployed and executed on the runtime environment of the SEALS infrastructure. This allows organizers to compare systems on the same basis, in particular in terms of runtime. It also solves the problem of reproducibility. This is also a test of the

deployability of tools. The successful deployment relies on the Alignment API and requires additional information about how the tool can be executed in the platform and its dependencies in terms of resources, e.g., installed databases or resources like WordNet, etc. For that reason the challenging goal of the SEALS project can only be reached with support of the matching community and depends highly on the acceptance of tool developers. We believe that an online available evaluation service is a key component to raise the acceptance in the community.

Acknowledgements

The authors are partially supported by the SEALS project (IST-2009-238975).

References

1. P. Bouquet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krötzsch, L. Serafini, G. Stamou, Y. Sure, and S. Tessaris. Specification of a common framework for characterizing alignment. Deliverable D2.2.1, Knowledge web NoE, 2004.
2. J. Euzenat. Towards composing and benchmarking ontology alignments. In *Proc. ISWC Workshop on Semantic Integration*, pages 165–166, Sanibel Island (FL US), 2003.
3. J. Euzenat. An API for ontology alignment. In *Proc. 3rd International Semantic Web Conference (ISWC)*, volume 3298 of *Lecture notes in computer science*, pages 698–712, Hiroshima (JP), 2004.
4. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. Trojahn dos Santos, G. Vouros, and S. Wang. Results of the ontology alignment evaluation initiative 2009. In P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, N. Noy, and A. Rosenthal, editors, *Proc. 4th ISWC workshop on ontology matching (OM), Chantilly (VA US)*, pages 73–126, 2009.
5. J. Euzenat, A. Ferrara, C. Meilicke, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, and C. Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2010. In P. Shvaiko, J. Euzenat, F. Giunchiglia, H. Stuckenschmidt, N. Noy, and A. Rosenthal, editors, *Proc. 5th ISWC workshop on ontology matching (OM), Shanghai (Chine)*, pages 1–35, 2010.
6. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer, Heidelberg (DE), 2007.
7. C. Meilicke and H. Stuckenschmidt. Incoherence as a basis for measuring the quality of ontology mappings. In *Proc. of the ISWC 2008 Workshop on Ontology Matching*, Karlsruhe, Germany, 2008.
8. H. Seddiqui and M. Aono. Anchor-flood: results for OAEI 2009. In *Proceedings of the ISWC 2009 workshop on ontology matching*, Washington DC, USA, 2009.