

SWS CHALLENGE

Status, Perspectives and Lessons Learned So Far

Charles Petrie*, Tiziana Margaria†, Ulrich Küster‡, Holger Lausen§ and Michal Zaremba§

*Computer Science Dept., Gates Building, Stanford, CA 94305-9020, USA

†Chair of Service and Software Engineering, Institute for Informatics, University of Potsdam, 14482 Potsdam, Germany

‡Institute for Computer Science, Friedrich-Schiller-University Jena, 07743 Jena, Germany

§DERI Innsbruck, University of Innsbruck, Austria

Keywords: SWS, Semantic, Web, Service, jABC, jETI, Application, Building, Center, Electronic, Tool, Integration, Milano, Dortmund, Potsdam.

Abstract: In this final contribution we summarize the current status and achievements of the SWS Challenge, present the perspectives and future steps, and summarize the lessons learned so far - both concerning this Challenge and the non-competitive problem driven comparison of approaches in general, as for instance in the context of other initiatives with a similar spirit (like the jETI-FMICS and Bio-jETI initiatives).

1 INTRODUCTION

The SWS Challenge has held three workshops in 2006, the third evaluating six (6) teams. A fourth workshop is scheduled to be co-located with the European Semantic Web Conference in Innsbruck in June 2007. A fifth is scheduled for Stanford University in November of 2007.

We briefly summarize here the reflections and lessons learned over this first year of activity. They concern our *methodology*, the *setup*, the *evolution* of the scenarios, and our *future activities*.

2 METHODOLOGY

For the most part, our experience has validated the methodology though we have learned much during the year: i.e., we have had to refine the methodology but slightly over the course of a year.

Claims. For each team submission, we evaluate the claims by having the workshop participants mutually examine the code changes of the submission. Initially, we thought that we would need to divide up

into teams to examine the submissions but we found that the whole workshop could collectively examine each submission and that everyone wanted to do so. We suspect that since the results are developed by the collective consensus of the whole workshop, they are better than they would have been had they been developed by smaller groups.

Comparison Criteria. We also initially tried to rank the submissions in difficulty of moving from one problem level or sub-level to another by trying to determine whether code was changed that would necessitate a re-compilation and linking, or whether there was only a change to the declaration of objects upon which the code acted. Further, we wanted to distinguish between whether the current declarations had to be altered, or whether new declarations were simply added. We found that these distinctions could not be made objectively. For example, if someone is writing in Lisp, there is no objective difference between declarations and code. XML schemas and Java present similar though less extreme problems.

We have resorted to a collective consensus on simply whether code or declarations have been changed as a measure of difficulty in moving from one level solution to another. This has been particularly challeng-

Petrie C., Margaria T., Küster U., Lausen H. and Zaremba M. (2007).

SWS CHALLENGE - Status, Perspectives and Lessons Learned So Far.

In *Proceedings of the Ninth International Conference on Enterprise Information Systems*, pages 447-452

Copyright © SciTePress

ing especially in approaches where solutions are synthesized by arranging software components in a graph with a GUI. One consideration has been whether changing the graph requires a re-compilation and linking, producing new code or whether this is essentially a declarative input to an engine, the code of which never changes: only its behavior.

Open Approach. One of the major successes of our methodology has been the open approach. First, participants are asked to submit new scenarios (including web services) and these are constantly being evaluated and added to our problem suite. Second, all solutions are documented and participants are encouraged to "steal" from each other. One of the teams that has solved the most problems uses one approach to solve the mediation problem and another to solve the discovery problem. This team is composed of people from two different institutions who have developed a successful synthesis of technologies.

This is exactly the sort of outcome we hoped for: understanding of which approaches worked best for what kind of problems and cooperation among researchers at different institutions.

3 INFRASTRUCTURE AND SUPPORT

Already the first time when we discussed to build up a challenge for semantic Web service systems we agreed on one fundamental principle: "No Participation without Invocation". However this principle brought some well underestimated effort for both the organizers as well as the participants. On the other hand the challenge greatly profited by enforcing to have real Web Service available, documented and running at all times: it enforced everybody not to hush up a problem that occurred, but to solve it.

Web Service Infrastructure. We have started with three Web Services simulating a client trying to purchase goods using the RosettaNet protocol and its counter part, the Moon legacy system. Taking into account different versions of services and the mediation systems that have been implemented to test the system we are operating at present around 20 different Web Services. Over time five different developers have been involved for different aspects of the execution platform. All services have now been migrated to the axis2 engine for Web Services. Unfortunately, the complexity of the messages used has revealed several bugs in the implementation of the axis2 engine, which

caused spending major resources just on the underlying technologies and not purely on the "business problem."

In fact it turns out that a variety of skills is required to master such a testbed. First, in-depth knowledge of WSDL and XML Schema to design proper service description utilizing the maximum of the descriptive power of the standards. Most obviously some knowledge on a web service engine (such as axis2) and the underlying application server (such as tomcat) is required as well as a fair amount of database design and web application programming skills. It also turned out to be necessary to understand a good deal about the Internet Protocol and firewalls in order to help participants to manage their invocations. And, last but not least, such an infrastructure requires some monitoring facilities that guarantee a 24/7 live system, which is not the usual approach in a university respectively research environment.

Effectively it demonstrated that in spite of the fact that Web Services are an established technology, current tools are only able to hide a small degree of the underlying complexity. As soon as we reached some border case, understanding of underlying protocols and standards was essential.

Besides the technical challenge we realized another important point: We decided to not formalize the problems using a logical formalism, but rather to describe them using natural language documentation. Having to communicate with developers as well as participants, we conclude that only having text based documentation as a common model is suboptimal. We realized that a fair amount of the solution to the problems is its formal description. In fact, had we had such descriptions from the start we could have saved several iterations of discussion with developers.

Collaboration Infrastructure. Having effective means to share information between the organizers and the participants is another important aspect for a successful challenge. We have started with a set of static web pages, however it was soon clear that this is suboptimal. A Wiki that enables corrections and improvements on the documentation in a collaborative fashion turned out to be much more adequate. While this improved the efficiency of the discussions around the different problems sets, it turned out not to be enough to share descriptions of the solutions between participants.

Similar to the problems, also the solutions come with a fair amount of complexity. In order for a team to participate, we required to publish the declarative parts of the teams solution on the Semantic Web Challenge Portal. A Wiki did not provide sufficient means to share such complex structures, so in addition we

created FTP accounts. However this turned out to be suboptimal: while it enabled to understand and verify a particular solution, the link between a solution's description in the papers submitted to the workshops, to the related discussion on the Wiki, and finally to the relevant parts of a solution's declarative description is too little integrated. We assume that this is one of the reasons why so far participants only share to a very limited amount of their formalizations. We hope to improve this in the future.

Evaluation and Debug Infrastructure. Another aspect of involving real Web services is the possibility to automatically verify a solution by issuing a set of different messages and monitor the subsequent message exchanges. This is a useful feature, since it makes the challenge more scalable with respect to the number of participants - it essentially enables to automatically verify solutions. Moreover it allows for teams to participate not only during workshops, but also at any other time by just exposing their Web Services. Other people interested in the claims of a team can just use the online portal to start a test set against a particular solution and verify its coverage.

Another aspect is to offer some form of debugging support. Already with six teams it was quite often necessary to examine the application server's log, be it to determine a typo in the endpoint addresses used in a mediator implementation, or to identify an invalid message. Over time we added different views to the online portal that allows to examine parts of the message exchange and in particular the status of the systems involved.

4 STATUS AND FUTURE OF THE MEDIATION SCENARIO

As of now there are three levels related to the data and process mediation scenario. The first, original scenario involves the mediation between Blue and Moon, within a stable (static) scenario: the protocols, the messages, and the data formats are known and fixed.

Data and process mediation scenarios have been based completely on the RosettaNet protocol (RosettaNet Website, 2007). RosettaNet Partner Interface Processes (PIPs) allow trading partners to connect electronically to process transactions and move information within their extended supply chains. The first impression of the RosettaNet specification is its completeness, but once we started to work on scenario definition and implementation, we realized that several aspects of the specification should be improved

to allow for automation of the RosettaNet processes. We can give a couple of examples: The same fields in the schema of one message are defined differently in the schema of another message (even within the same PIP). There are various possible interpretation for particular fields in the messages, causing ambiguities: two teams working on the integration solution might actually use the same field differently. Various cases allow for free interpretation, e.g. having an address defined on the order level and on the line item level caused a confusion about which one should be used. Regarding the practical problems, potential RosettaNet messages are extremely large (e.g. even to confirm a message, the whole initial message must be included with it), but the schema requires that at the same time the whole message with many empty fields is sent. Due to lack of formal semantics, processes defined by UML specification can be interpreted differently by various teams.

Since Web services address also dynamically changing scenarios, even when knowing the partner (i.e. without discovery) it is already possible and likely that

1. WSDL descriptions may change, leading to different message structures being exchanged, and a need for all the conversation partner to adapt
2. protocols may change, for example when adding complexity in the structure of an operation. In this scenario, instead of one line item per order multiple line items are allowed. This requires adapting the business logic of the mediator.

No new levels are currently foreseen.

At the time of the last SWS-Challenge workshop in Athens, GA, six groups had presented their solutions to the mediation approach. Five of them were ranked according to the evaluation criteria, and indeed they showed very different approaches. From the most to the least declarative, we range

- from a fully declarative approach based on METEOR-S (Wu et al., 2007), where nearly full automation was achieved, to
- three approaches that combine partially automatic generation and partially automatic adaptation, but in different subproblems:
 - the WSMO/WSMX approach of (Zaremba et al., 2007) uses a generic (abstract) state machine for the flow, thus it has advantages on the process adaptation level
 - the WebML/Webratio (Zaremba et al., 2007; Margaria et al., 2007) uses generic import/export mechanisms from the WSDL and a partial generation of the processes, that ease the adaptation

- the jABC/jETI approach (Margaria et al., 2007) provides automatic generation of ad hoc components from the published WSDL descriptions into its own service components and (at this stage) manual graphical construction of the service logic, thus supporting automatic adaption for level 1a and requiring manual intervention for level 1b.

These three approaches have been pairwise compared in two contributions to this special Session: (Zaremba et al., 2007) and (Margaria et al., 2007). Additionally,
- two approaches resort to software engineering for the mediation solution:
 - the DIANE approach is actually geared primarily (fast exclusively) towards discovery, thus the mediation solution falls outside the specific profile. The mediation problem was solved traditionally (Ulrich Küster, 2007), by providing specific adapters to the RosettaNet messages and to the Moon system, and a process logic written in BPEL. The adaptation required for level 1a was achieved automatically and for level 1b manually, by editing the BPEL.
 - the Swashup solution (Michael Maximilien, 2007) is a pure software engineering approach based on agile programming by means of Ruby on Rails service mashups. It has been designed for compactness of the code, and it is the only one that for the moment does not provide graphical support to the programmer.

The level of declarativeness was considered initially as an indicator of merit of a solution, in that it indirectly expresses its abstraction (from the programming level) and robustness (wrt. changes and evolution). At least one group (jABC/jETI) is currently working on achieving automation of service logic composition from declarative specifications.

5 STATUS AND FUTURE OF THE DISCOVERY SCENARIO

As of now there are two comprehensive scenarios related to service discovery and matchmaking. The first, original scenario involves the discovery of an appropriate shipment service out of five offers, each with different peculiarities regarding price, supported locations, maximum package weight, constraints on the pickup time and the speed of delivery. One offer required to call the service to check for the actual price of a particular shipment.

Based on a hierarchy of increasingly difficult given goals (i.e. shipping requests), submitted solutions are evaluated. At the time of the last SWS-Challenge workshop in Athens, GA, four success levels have been evaluated and two more were planned but not ready for evaluation, since the corresponding goals had not been released:

1. discovery based on location
2. discovery with arithmetic price and weight computations
3. discovery including request for quote
4. discovery including sending multiple packages (which had to be resolved to multiple service invocations).
5. discovery with temporal semantics, i.e. pickup times and required speed of delivery (not ready for evaluation)
6. discovery with conversion of measurement units (not ready for evaluation).

Meanwhile goals for level 5 as well as a completely new scenario have been released. Submissions for the new goal and the new scenario will be evaluated beginning at the upcoming Fourth SWS-Challenge Workshop co-located with the ESWC in June 2007.

The new second discovery scenario deals with the discovery of a vendor for electronic products. It includes three sources of difficulty:

- Currently only few products which are available for purchase have been modelled in the scenario. However, it is planned to extend this to a much more realistic setting. Participants should indicate how they are going to cope with semantic descriptions of vendors offering tens of thousands of different products.
- A limited notion of preference is introduced to the requests which require some notion of ranking among matching offers.
- Most requests cannot be serviced by a single invocation of a single offer, instead some means of automated service composition are required to solve the more advanced goals.

Success levels to evaluate submissions against this scenario will be developed at the Fourth SWS-Challenge Workshop, but the following levels are envisioned and checked for by the released goals:

1. discovery based on clear product specifications
2. discovery including preferences (like as cheap as possible)

3. discovery for multiple products that must be resolved to multiple service invocations
4. discovery for multiple products with a global optimization goal (e.g. overall minimal price)
5. discovery for multiple correlated products (like a notebook and a compatible docking station)
6. discovery for multiple correlated products and a global optimization goal.

For the near future two extensions to the scenarios are planned.

On the one hand we will add goals that require automated unit conversion to either of the discovery scenarios. This might e.g. be done by mixing products with a price stated in Dollars with products with a price stated in Euro in the scenario. Participants will have to detect that prices are given in different currencies and develop means to deal with this, e.g. by automatically invoking a currency conversion service during service matchmaking. This will be one further step towards really adaptable systems.

On the other hand we are currently working to include a realistic number of products into the supplier scenario. We are investigating whether it is possible to exploit the Amazon E-Commerce service to gather the necessary amount of realistic product data. Including a large number of products into the scenario will have major implications on the solutions. First, creating meaningful descriptions will become much more difficult. On the one hand a broad generic description in the sense of “this service sells electronic products” will be of little use during discovery. On the other hand it might not be feasible to explicitly list all available products within a description for various reasons (privacy, dynamicity, ...). Thus participants will have to balance their solution somewhere between these extremes and decide on the amount of statically encoded information versus the amount of information being dynamically gathered.

The first of the two planned extensions is targeted at increasing the complexity of the discovery problems at the process and reasoning level. Solutions being able to still tackle the problems will have proven an even higher level of adaptability to homogenous environments. The second extension is complementary and increases the complexity with regard to the amount of information that needs to be processed and – finally – taken advantage of during discovery. Both extensions combined are aiming at making the discovery scenarios even more realistic than they already are, thereby underlining the goal of the SWS-Challenge to provide industrial level application scenarios.

6 EVOLUTION AND FUTURE PLANS

We will continue the organization of the SWS Challenge workshop in 2007 and hopefully beyond it. The initiative is now going beyond its initial boundaries and we hope to target much wider audience. Just when finalizing this paper, the W3C Semantic Web Service Testbed Incubator Group initiated by Challenge organizers has been approved by the W2C. The mission of the group is to develop a standard methodology for evaluating semantic web services based upon a standard set of problems and develop a public repository of such problems. There is a new Coordinated Action proposals in preparation for European Commission, which includes Challenge as one of its core activities. And we have a book planned to report on the first year’s results.

The Challenge is now quite a growing and still naturally mutating “organism”. Many of the initial assumptions about how the challenge should be run and structured have been verified and/or modified during its execution. In this last section we would like to mention just a few new ideas for the future Challenge evolution.

The Challenge needs more new interesting scenarios. While the initial scenarios have been provided by the authors of this paper, this is not scalable, and currently we have already new scenario problems created by the larger SWS Challenge community. We are open for new proposals of interesting use cases, which could be hosted by the Challenge testbed system and against which participants could test their execution engines. This plans are also related with providing an easier process for submitting new problems. Currently we maintain WIKI infrastructure, where all the scenarios are stored. Together with the grow of the community, we should have some more formal process how we incorporate new use cases, how we make sure that their fit the interest of participants (so some formal approval process), as well as how makes the hard job and takes care of implementation of the problems. This will be part of the outcome of the W3C Incubator.

During previous workshops we used the whole workshop to evaluate solutions of all the teams. This cannot scale as the number of teams participating in the challenge is growing. What is even more important is the lesson we learned during Athens meeting, that teams might have different understanding of passing/not passing the same tests. The Challenge would require an improved integrated testbed allowing for automation of the process. The set of the automated tests would be deciding on behalf of organizers if the

team accomplished the given level of problems, as the automated script would be run against proposed solution (e.g. the message unknown to participants would be send to their mediators to make sure that the solution is not hardcoded and can actually handle any message).

Last, but not least is the idea of integrating different problems to allow "mashups" - combining content from more than one source into an integrated experience. Currently the scenarios are pretty separated and we proved during our past meeting the teams can accomplish one problem without even touching the other one. Given this independence, it would be interesting, to split the existing problems into micro-problems (and to host only micro-problems on the Challenge server), but to allow to mashup them freely to create even new scenarios, not envisioned by the creators of the mashups.

REFERENCES

- Margaria, T., Winkler, C., Kubczak, C., Steffen, B., Brambilla, M., Ceri, S., Cerizza, D., Valle, E. D., Facca, F. M., and Tziviskou, C. (2007). The sws mediator with webml/webratio and jabc/jeti: A comparison. In *Comparative Evaluation of Semantic Web Service Frameworks Special Session at ICEIS 2007*.
- Michael Maximilien, E. (2007). A partial solutio to the semantic web service challenge problem using swashup: The ruby on rails services mashup approach. In *Comparative Evaluation of Semantic Web Service Frameworks Special Session at ICEIS 2007*.
- RosettaNet Website (2007). *RosettaNet Website*. <http://portal.rosettanet.org>.
- Ulrich Küster, Birgitta König-Ries, M. K. (2007). *Discovery and Mediation using DIANE Service Descriptions*. available at <http://sws-challenge.org/2006/paperathens/SWS-Challenge-2006-Athens-Full.pdf>.
- Wu, Z., Ranabahu, A., Gomadam, K., Sheth, A. P., and Miller, J. A. (2007). *Automatic Composition of Semantic Web Services using Process and Data Mediation*. Techn. Rep. Kno.e.sis Center, Wright State University, Dayton, Ohio, February 28, 2007.
- Zaremba, M., Vitvar, T., Moran, M., Brambilla, M., Ceri, S., Cerizza, D., Della Valle, E., Facca, F. M., and Tziviskou, C. (2007). Towards semantic interoperability: In-depth comparison of two approaches to solving semantic web service challenge mediation tasks. In *Comparative Evaluation of Semantic Web Service Frameworks Special Session at ICEIS 2007*.