

SWS CHALLENGE

First Year Overview

Charles Petrie*, Holger Lausen† and Michal Zaremba†

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

†DERI Innsbruck, University of Innsbruck, Austria

Keywords: SWS, Semantic Web, Web Services, Challenge, Methodologies.

Abstract: This paper provides an overview of the results of the first year of the Semantic Web Services Challenge. Other papers in this same conference proceedings provide detail of participant contributions as well as lessons learned. This paper is an introduction to the Challenge purpose, organization, methodology, problems, and solutions.

1 SWS CHALLENGE MISSION AND ORGANIZATION

Service-Oriented Computing is one of the most promising software engineering trends for future distributed systems. Pushed by major industry players and supported by many standardization efforts, Web services are a prominent implementation of the service-oriented paradigm. They promise to foster reuse and to ease the implementation of loosely coupled distributed applications.

Though Web services are appealing especially in the area of enterprise application integration, the idea of service-oriented computing and the envisioned availability of thousands of services can not be fully leveraged as long as service oriented applications must be created and maintained manually. Semantic technology may help here, by lifting service-oriented applications to a new level of adaptability and robustness. By using semantic annotations to describe services and resources, the tasks of service discovery, selection, negotiation, and binding could be automated.

Currently there are many different approaches to semantic Web service descriptions and many frameworks built around them, yet a common understanding, evaluation scheme, and testbed to compare and classify these frameworks in terms of their abilities and shortcomings is still missing.

The purpose of the ongoing *Semantic Web Service*

Petrie C., Lausen H. and Zaremba M. (2007).

SWS CHALLENGE - First Year Overview.

In *Proceedings of the Ninth International Conference on Enterprise Information Systems*, pages 407-412

Copyright © SciTePress

Challenge (www.sws-challenge.org) is precisely to develop this common understanding of various technologies intended to facilitate the automation of mediation, choreography and discovery for Web Services using semantic annotations. This explores trade-offs among existing approaches, reveals the strengths and weaknesses of the proposed approaches as well as which aspects of the problem space are not yet covered.

The series of three workshop in the first year of the SWS Challenge has provided a forum for discussion based on a common application. The Challenge focuses on the use of semantic annotations: participants are provided with semantics in the form of natural language text that they can formalize and use in their technologies. Being a challenge rather than a contest, workshop participants mutually evaluate and learn from each others' approaches.

The Challenge has participating groups from industry and academia developing software components and/or intelligent agents able to automate mediation, choreography and discovery processes between Web services. All approaches and participants are invited. Though "Semantic" is in the title of this Challenge, we invite non-semantic approaches and evenly evaluate all submissions in our methodology.

2 SWS CHALLENGE METHODOLOGY

At the SWS workshops, the approaches were presented and demonstrated, but also the code was jointly reviewed. The common application helped developing a profound mutual understanding of each other's technology and the collaborative discussion of the profiles of the various approaches. The participants developed an evaluation scheme that classifies the functionality and the agility offered by the various approaches, and applied it to the participating technologies.

The SWS Challenge is an evaluation of functionality rather than performance. We are not interested in how fast a particular piece of code works. Certainly algorithms for analyzing web services are relevant, but algorithms can be analyzed in terms of complexity apart from code. We are interested not in the speed of the code but in programmer productivity. For a given change in the emerging era of Service-Oriented Architectures, how hard will it be for programmers to make changes in an increasingly flexible and dynamic IT environment? This Challenge seeks to understand the advantages and tradeoffs, wrt. this question, of various programming approaches.

There is no "winner" in these challenges, though one can look at the results of each workshop and see which team has so far solved the most problems with what level of difficulty. This Challenge is intended to be an objective certification of approaches to the problems of semantic technologies, with an emphasis on industrial problems in order to make the technologies relevant.

Therefore the SWS Challenge is taking a software engineering approach to evaluating Semantic Web Services.¹ The working hypothesis of the semantic technology community is that a semantic approach will allow a given change to be made with less difficulty than with traditional coding techniques. This is essentially a software engineering claim. Thus we allow "all comers" to participate. If it develops that a particular coding technique manages the problem changes of the challenge scenarios better than a semantic approach, this will also be valuable information for the semantic community.

In the Challenge methodology, teams automatically validate their solution to problems by having their system send correct messages to the web services in the SWS Challenge infrastructure. At the workshops, teams present papers about their approach

¹"It's the Programming, Stupid", *IEEE Internet Computing*, May-June, 2006.
<http://www-cdr.stanford.edu/petrie/online/peer2peer/w306.pdf>

with claims about the ease of changing from one problem to another. Then we peer-review these claims and agree upon an evaluation of the approach, as well as certifying the technology problem level.

The problems are specified in English, other than the WSDL descriptions associated with the test services. We challenge the participating teams to develop their own semantic annotation formalisms that are sufficient to solve the problems. Additionally, we analyse the general difficulty in moving from solving one problem level or sub-level to another.

This is why we use standards such as RosettaNet and WSDL and make our scenarios at least similar to industrial problems. It is also why we insist that submissions actually solve the problems by sending correct messages to the Challenge web services for each scenario. It is easy to make claims in academic papers that such-and-such a problem has been solved. It turns out to be much more difficult in practice, as our teams have discovered, to make such approaches actually work. Our slogan is "no participation without invocation". In order to be evaluated on a problem level, the submission must have demonstrated the correct exchange of messages with the corresponding Challenge web services.

3 SWS CHALLENGE PROBLEM SCENARIOS

The problems being solved by the teams are business scenarios divided into major problem levels with sub-problem variations. The first major problem level consists of developing a mediator that allows a hypothetical company to have its legacy web services to conform to a RosettaNet purchase order (PO) standard that is being used by a customer. We then change the web services, the protocol, and the order in consecutive variations.

In particular, we have presented two broad areas of problem scenarios:

- The *mediation* scenario problems concern making a legacy order management system interoperable with external systems that use a simplified version of the RosettaNet PIP3A4 specifications².
- The *discovery* scenario problems concern the dynamical discovery, selection, binding, and invocation of the most appropriate shipment service for a set of given shipment requests.

Subsequent problem levels involve increasingly difficult web service discovery and composition sce-

²<http://www.rosettanet.org/PIP3A4>

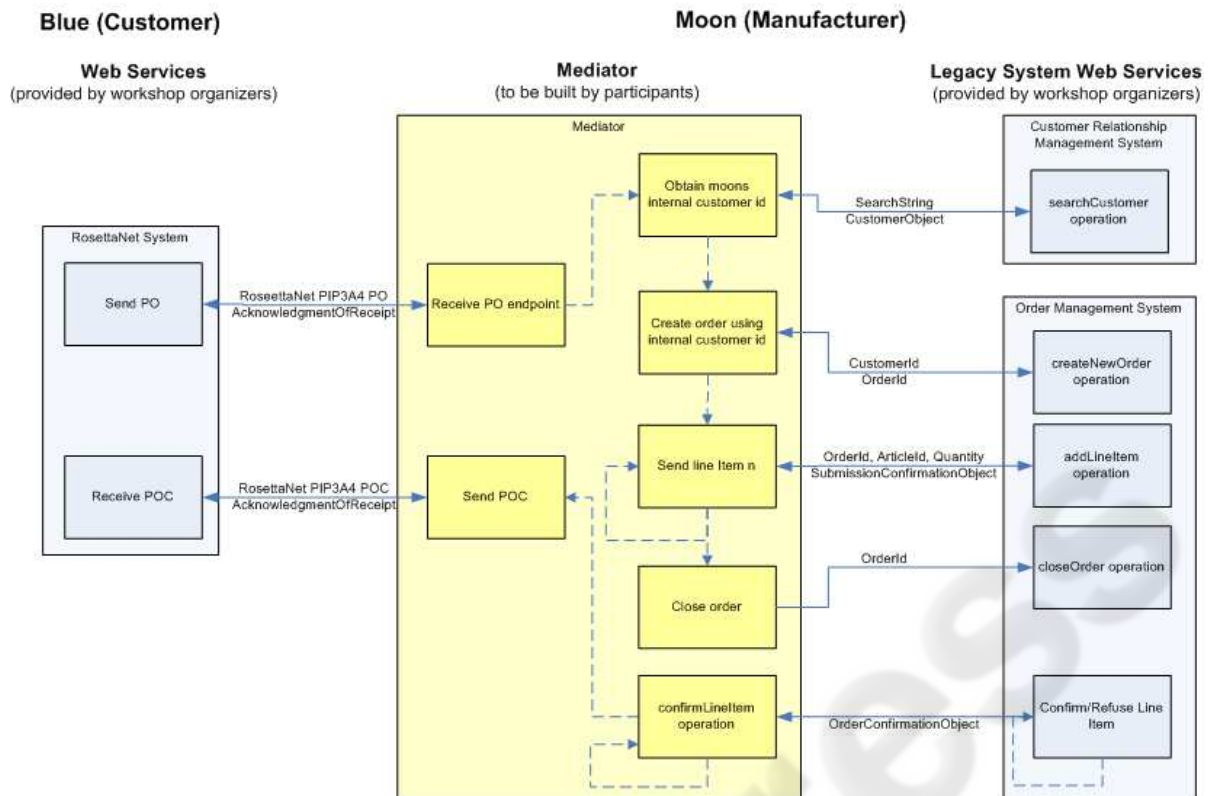


Figure 1: Mediation Scenario.

narios. We have not yet developed all of the scenarios, but in the end, the teams should process the order from the company, mediating the PO process, order the right parts from the suppliers, the suppliers should ship the parts, and the company should ship to the customer the completed order, with associated “paperwork”.

4 THE MEDIATION SCENARIO: PROCESS AND DATA MEDIATION

For this challenge we focus on the very basic scenario of purchasing goods using a simplified version of the RosettaNet PIP3A4 specifications.

There are three main components taking part in the scenario, as depicted in Figure 1:

- **Company Blue**, which is a customer (service requester) ordering products,
- **Mediator** which is a piece of technology providing automatic or semi-automatic mediation for the Moon company

- **Legacy System** of the Moon Company

While the external interfaces must follow the RosettaNet specification, internally Moon uses a propriety legacy system in which data model and message exchange patterns differ from those of RosettaNet. Participants shall basically enable Moon to “talk RosettaNet” and implement the Purchase Order receiving role part of the interaction described in the RosettaNet PIP 3A4.

Both the Moon legacy systems and the customer Web services (Blue) are provided by the challenge organizers as technical infrastructure running at DERI in Innsbruck and accessible online, and can not in principle be altered (although their description may be semantically enriched). The sketch of the mediator of Fig. 1 shall be implemented by the participants.

In the mediation scenario, Moon uses two backend systems to manage its order processing, namely a Customer Relationship Management system (CRM) and an Order Management System (OMS). The challenge provides access to both systems through public Web services described using WSDL. The scenario describes how Moon has signed agreements to exchange purchase order messages with its client com-

pany called Blue using the RosettaNet PIP 3A4 specification.

In order to address integration of Blue and Moon companies, the participating groups are encouraged to use SemanticWeb service technology to facilitate conversation between all systems, to mediate between the PIP 3A4 and the XML schema used by Moon, as well as to ensure that the message exchange between all parties is correctly choreographed. In particular,

- **Data mediation** is involved in mapping the Blue RosettaNet PIP 3A4 message to the messages of the Moon back-end systems.
- **Process mediation** is involved in mapping of message exchanges defined by the RosettaNet PIP 3A4 process to those defined in the WSDL of the Moon back-end systems.
- **Conversations** between the systems including data and process mediation operate on semantic descriptions of messages, thus requiring the transformation from messages used by existing systems to the ontological level.

Infrastructure. The existing systems are Moon's back-end applications including CRM and OMS systems as well as Blue's RosettaNet system (gateway). Each system communicates using different formats, e.g. Blues RosettaNet gateway communicates according to the RosettaNet PIP 3A4 message (Purchase Order), whereas communication with the CRM and OMS systems is more proprietary - specified in their WSDL descriptions. Detail descriptions of these WSDL interfaces can be found at SWS-Challenge web site.

Challenge Levels. The workshop organizers provide a set of challenge problems. These build upon the initial mediation problem, which we call Level 0. On top of this various levels are added, each corresponding to a general kind of problem, and each with sublevels of complexity. For the mediation scenario:

Level 0: Mediation Scenario (static)

Evaluation: is determined automatically by the sandbox system - when the solution is able to pass the basic conversation.

Level 1: Mediation Scenario (adapting to changes in systems), with the following distinction:

Level 1a: Data Mediation: signatures are changed.

Level 1b: Process Mediation: more than one item appears in the order.

Evaluation: determined by peer review at the SWS workshops.

The changes involved in the levels 1a and 1b were not published from the beginning, but made available to the participating groups only once they solved the previous levels.

5 THE DISCOVERY SCENARIO

The discovery scenario is a more visionary scenario compared to the mediation scenario in the sense that it cannot be solved with current syntactic techniques alone. This fact is also reflected by the smaller number of solutions submitted to this scenario.

The scenario defines five shipping services (described via their WSDL and a natural language documentations) and presents a set of increasingly complex shipping requests. Given a request, a suitable shipper needs to be discovered and invoked. Thus, participants have to create (semantic) descriptions for the available shippers and the given shipping requests such that the discovery and invocation task can be performed by an automated autonomous agent.

Recently the shipping scenario has been extended but here we describe the scenario in the version that has been evaluated at the Third SWS-Challenge Workshop in Athens (November 2006).³

Shipping services are characterized by the following properties:

- **Operation range:** Shippers operate worldwide or in a set of listed countries or continents.
- **Package limitations:** Shippers define maximum bounds on the dimensions and the weight of packages. Additionally the notion of a *dimensional weight* is used: Packages with a low weight, but big dimensions need to use the dimensional weight (computed from the dimensions of the package) instead of the actual weight.
- **Price:** Four shippers statically specify the price as rules how to compute the price of a package depending on shipping location and package dimensions or weight. One shipper requires to dynamically call a webservice endpoint to gather the current price providing the same information. Thus for goals specifying an upper price limit for the shipping operation, this service could not be

³A description of the extensions released since can be found in the "SWS Challenge: Status, Perspectives, and Lessons Learned so far" paper in these same conference proceedings.

discovered by exploiting static descriptions alone, but required dynamic negotiation during the discovery process.

- **Package collection:** Shippers offer collection of packages and define various constraints on the minimum or maximum advance notice for collection or the total length of the collection interval. (This aspect was not covered by the set of goals evaluated in Athens.)

The predefined requests specify a required shipping operation, characterized by concrete pickup and delivery addresses as well as concrete package dimensions and weight. The more complex goals additionally specify a maximum price for the shipping operation. During discovery, participants have to filter unsuitable shippers, automatically choose a suitable one and invoke it by calling the corresponding web service endpoint. Since the shippers do not use a common XML-Schema for their messages, participants also have to deal with issues of data mediation to create the properly formatted messages.

One advanced goal requested the sending of two packages instead of one. Since none of the shippers support multiple packages, this goal has to be mapped to multiple invocations of the same or different shippers.

6 OVERVIEW OF CONTRIBUTIONS

In the first year of the SWS Challenge, we have had three workshops with six (6) teams participating. The other papers in this conference proceedings describes the contributions of these teams in more detail. Here we give an overview that will help the reader with an overall perspective. In each case, we provide the name of the primary contract for the approach(s).

- **DIANE (Universität Jena)**
Contact Person: Ulrich Küster, Institute for Computer Science, Friedrich-Schiller-Universität Jena, 07743 Jena (Germany), Tel: +49 3641 946433, Ulrich.Kuester@uni-jena.de

DIANE is a method for automated service match-making, selection, binding and invocation and was used in the discovery scenarios.
- **WebML/Webratio (Politecnico di Milano/CeFRIEL)**
Contact Person: Federico Michele Facca, Politecnico di Milano, Dip. di Elettronica e Informazione, 20133 Milano (Italy), Tel: +39 02 2399 9623, facca@elet.polimi.it

and

- **jABC/jETI (Universität Dortmund/Universität Potsdam)**
Contact Person: Christian Kubczak, Universität Dortmund, Chair of Programming Systems, Dortmund (Germany), Tel: +49 231 755 7757, christian.kubczak@cs.uni-dortmund.de

These two teams took a software engineering approach to the mediation scenario.

- **WSMX (DERI)**
Contact Person: Maciej Zaremba, Digital Enterprise Research Institute (DERI - <http://www.deri.ie>), National University of Ireland, Galway, (maciej.zaremba@deri.org)

This team solved both mediation and discovery problems using WSMX.

- **Swashup (IBM)**
Contact Person: Max Maximilien, IBM Almaden Research Center, San Jose (CA), tel: +1 408 9272124, maxim@us.ibm.com

The “Swashup” approach uses Ruby on Rails for a purely engineering approach to solving the first level of the mediation problem.

- **METEOR-S (Wright State University)**
Contact Person: Karthik Gomadam, Kno.e.sis Center (<http://knoesis.org>), Dept. of Computer Science & Engineering, Wright State University, Dayton, OH, karthik.gomadam@gmail.com (with collaborators at LSDIS lab, University of Georgia, Athens, GA).

This team’s approach uses SAWSDL + Planning + Data Mediation Web service to solve the mediation problem.

The precise results of all of the teams as of the Athens, George Workshop can be found in the SWS Challenge Wiki⁴. It should be noted here that the three teams of Politecnico di Milano/Cefriel, DERI, and Jena currently have solved the most problems in both the mediation and discovery scenarios. The IBM approach was near to solving the first level of the mediation problem but had not done so as of February 2007. We expect a solution at the Innsbruck workshop in June of 2007. Other participants are invited to contribute to this workshop and another planned for November at Stanford University.

Finally we would emphasize that the SWS Challenge is open, both to participation and to the submission of new scenario problems. Some of the scenarios

⁴http://sws-challenge.org/wiki/index.php/Workshop_Athens

can be “stand alone” and others will refine and extend the “Blue Moon” customer/company/supplier/shipper scenario. Eventually, this scenario should include the company fulfilling the customer order using a supply chain composed of the best suppliers and shippers for the specific customer order.

Our mission is to supply not only a large useful “sandbox” for testing semantic web service approaches, but also a *de facto* standard for certifying such technologies, as well as furthering an academic understanding of the benefits and trade-offs of these approaches. The papers in this session are the first results from this understanding as they include comparisons of approaches.

SeitePress