

Semantic Web Services: Past, Present, and Future
Technical Panel, Dynamic Web Processes (DWP) 2005 Workshop
In conjunction with International Conf. on Service-Oriented Computing (ICSOC) 2005
Amsterdam, The Netherlands, December 12th, 2005 より抜粋

川村 隆浩

(株) 東芝 研究開発センター 知識メディアラボラトリー

e-mail: takahiro@eel.rdc.toshiba.co.jp

HP: www.maekawa.is.uec.ac.jp/~kawamura/

Motivations

Semantic Web Services (SWS) promise to facilitate all aspects of Web services usages. **The overarching goals of the various SWS efforts are to provide some level of automation of Web services discovery, selection, invocation, and composition.** Indeed, early efforts, e.g., OWL-S, have demonstrated results such as dynamically matchmaking and composing Web services using a service ontology that includes descriptions of the services capabilities and needs of the service consumer. Other early efforts, e.g., METEOR-S, developed a broad framework of semantics that can be added to Web services to facilitate their use.

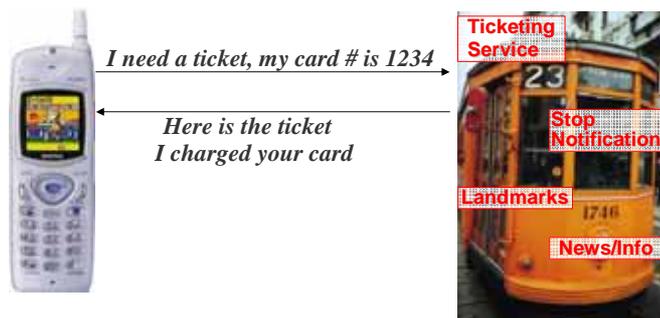
Though decent progress has been achieved, **we are far from completely realizing all of the aforementioned goals and gaining wide acceptance.** There remain various challenges and questions to be addressed. Recently, newer efforts, i.e., WSMO, SWSF/FLOWS, and WSDL-S, have emerged taking at times a different approach to adding semantics to Web services. **Since none of the efforts have been widely adopted (so far) and Web services themselves are facing some challenges from simpler alternatives such as REST and some of the Web 2.0 initiatives;** we maybe at a juncture where some discussions and sharing of lessons learned may be needed to galvanize the SWS efforts and community, as well as to give some directions for future research and approaches.

Building on a successful ONTOLOG-forum SWS panel on October 10th, 2005, this panel brings together various leaders from the major SWS initiatives and community, from both academia and industry. After a short introduction and presentation from each initiative, we intend to engage in a debate and discussion that will include the following

Questions

1. How can the complexity of the current SWS approaches be reduced to help gain some wider adoption? After all, the primary goal of the SWS efforts is to facilitate the usage of Web services.
2. Is a folksonomy-type approach a better, more realistic, alternative to adding semantics to Web services? (similar to efforts like <http://del.icio.us> and Yahoo! Flickr)
3. Should the SWS community take a pragmatic approach to adding semantics to Web services by heavily leveraging and extending the existing Web services stack as was done with WSDL-S? Or, is that a flawed approach since it inherits any limitations of the stack?
4. Should the SWS community agree on some basic standards and help extend and improve the current Web services stack? And what are advantages and disadvantages?
5. What are some of the low-hanging fruits that the SWS community should strive for first and progressively address the vision questions? What are some basic use-cases (e.g., semi-automated Web services usages with human in the loop and automated Web services usages via software agents)?
6. Should SWS ontology annotation be limited to OWL-type, FOL DL- type languages? Or, should we look into adopting other languages for ontology/taxonomy constructions, e.g., UML?
7. Can formal approaches like FLOWS, which provides complete semantics of processes, help the implementation of use cases and achieve results that demonstrate clear advantages for businesses over well accepted languages like BPEL? What are some example use cases that show these advantages? Or should such formal approaches instead leverage and extend languages like BPEL?
8. What have we learned from current efforts that should drive the SWS roadmap?

Services on the go



Services are ubiquitous in the environment
 Simple services
 Easy to model: provide a good test case for (S)WS technology

Problems

- User should be in the loop
 - Users want to be in charge of decisions
 - Mobile should not take decisions they do not like
- Users are not programmers
 - Mobile should resolve interoperation issues
 - Are semi-automatic solutions a possibility
- Services have a limited range
 - No network, no discovery ahead of time and of course no composition ahead of time
- Services may require interaction with the user
 - Unclear how to acquire WSDL descriptions

Questions

1- Complexity of the approach

The complexity is not in the SWS side, the complexity is in the ontological reasoning.

Complexity in the SWS can be taken care of by tools

2+6 Is *folksonomy* better (more realistic)? Should we use other languages on top of OWL? For example UML?

Sure! But mind the task (and the price) If facilitating annotation is the goal, then the more the better.

But if interoperation is the goal, the more is not the better. Somebody will still have to integrate Flickr and UML

Would a flicker user do with UML?

Questions

3+4 Should the SWS community use (and contribute to) the WS-Stack? Should we converge to standards?

No brainer! Every body does it. OWL-S did it since 2002 or there about.

The goal of SWS (at least of OWL-S) has always been to contribute to the WS stack.

(We would have used FIPA otherwise ;-)

As for converging to standards... standards are good, and converging to WSDL-S is in principle good, but we should not use them to replace research

The question I hear is "what is the business model" not "where is the standard"

Questions

5- Low hanging fruits

There are no low hanging fruits, there are problems to solve. If SWS technology takes you there, then use it!

8- What we have learned, what should we do to move forward

Learned: what WS are about, what annotate, what are the issues

Move toward a synthesis

Integration challenge is good step

Comparative work on OWL-S/WSMO Meteor-s/WSMO are good steps