Combining Services and Semantics on the Web

Katia Sycara, Massimo Paolucci and Naveen Srinivasan
Software Agents Lab
Carnegie Mellon University
Pittsburgh, PA

Mark Burstein
Human-Centered Systems Group
BBN Technologies
Cambridge, MA

The current Web is a collection of human readable pages that are virtually unintelligible to computer programs. In recent years two parallel efforts emerged that have the potential of overcoming this limitation: the first effort is the Semantic Web [1] which provides the tools for the explicit markup of the content of Web pages. The second effort aims to the development of Web services: self contained programs that by becoming the producers and consumers of information facilitate the automation of business transactions.

The objective of the Semantic Web is to provide languages to express the content of Web pages and to make accessible to agents and computer programs the information that those pages contain. More precisely, the Semantic Web is based on a set of languages such as RDF and OWL that can be used to markup the content of Web pages. These languages have a well-defined semantics and a proof theory that allows agents to draw inferences over the statements of the language. As an example, an agent may use the semantic markup of the NOAA page reporting the weather conditions in Pittsburgh, and learn that the current condition is Heavy Snow; furthermore, the agent may infer from the semantic markup of the Pittsburgh school board page that in days of heavy snow all the schools are closed. By combining the two pieces of information, the agent would infer that indeed today Pittsburgh schools are closed.

The second element of the Semantic Web is a set of ontologies, which provide a conceptual model to interpret the information provided. For example, an ontology of weather may contain concepts such as temperature, snow, cloudy, sunny and so on. It may also contain information on the relation between the different terms; for instance, it may say that cloudy and sunny are two types of weather conditions. The vision of the Semantic Web is the transformation of the Web into an Internet wide knowledge representation system, in which web pages provide information and ontologies provide the conceptual framework needed to interpret that information. However, while the Semantic Web provides meaning to the data represented on the Web, it still relies on static web pages, or ontologies, that always report the same information.

Web services provide a way to disseminate information dynamically and on demand. There are many definitions of Web services. Here we adopt the definition in [2]. "A Web service is a software system identified by a URI whose public interface and bindings are defined and described by XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by Internet protocols." Despite the broad coverage of the Web services infrastructure and the amount of proposed interoperability standards, the emerging Web services infrastructure suffers from its dependence on pure XML for interoperation. XML guarantees syntactic interoperability which allows Web services to parse each others' messages but it is not enough for semantic understanding of the message content. As a result, Web services are left in the awkward state of identifying the pieces of information that they exchange, but they do not know how to interpret them.

Because of its lack of explicit semantics, the Web services infrastructure is forced to assume that programmers hardcode the Web services interaction and the interpretation of the information that they exchange. Such hardcoding makes the implementation of the interaction and its maintenance very expensive. Any change in the description of a product, or the addition of a new product requires programmers to modify the code managing the interaction. Semantic interoperability is crucial for Web services. It allows Web services to (a) represent and reason about the task that a Web service performs (e.g. book selling, or credit card verification) so as to enable automated Web service discovery based on the explicit advertisement and description of service functionality, (b) explicitly express and reason about business relations and rules, (c) represent and reason about message ordering, (d) understand the meaning of exchanged messages, (e) represent and reason about preconditions that are required to use the service and effects of having invoked the service, and (f) allow composition of Web services to achieve a more complex service.

The Semantic Web has the potential to provide the Web services infrastructure with the semantic information that it needs. It could provide formal languages and ontologies to reason about service descriptions, message content, business rules and relations between these ontologies. In this way, the Semantic Web and Web services are synergistic: the Semantic Web transforms the Web into a repository of computer readable data, while Web services provide the tools for the automatic use of that data.

The vision that we pursue is the realization of *Semantic Web services*, which result from the integration of semantic metadata, ontologies, formal tools and the Web services infrastructure. A Semantic Web service is a Web Service whose description is in a language that has well-defined semantics. Therefore, it is unambiguously computer interpretable, and facilitates maximal automation and dynamism in Web service discovery, selection, composition, negotiation, invocation, monitoring, management, recovery and compensation. Specifically, Semantic Web services rely on the Semantic Web to describe (1) the content of the messages that they exchange, (2) the order of the messages exchanged and (3) the state transitions that result from such exchanges. The result of using the Semantic Web is an unambiguous description of the interface of the Web service which is machine understandable and provides the basis for a seamless interoperation among different services.

The use of the Semantic Web to describe Web services has wide ranging consequences. It allows the description of additional properties of Web services, such as the quality of service and security constraints in a coherent and uniform way that is universally understood. Furthermore, and most importantly, the description of the states produced by the execution of the Web service is the basis for the description of its capabilities as a transformation from its inputs and preconditions, to its outputs and effects.

The first ontology for describing Web services is OWL-S [3]. OWL-S attempts to close the gap between the Semantic Web and the Web services infrastructure. As ontology, OWL-S is based on OWL to define the concept of Web service within the Semantic Web. In addition it provides a language to describe actual Web services that can be discovered and then invoked using standards such as WSDL and SOAP. OWL-S uses the semantic annotations and ontologies of the Semantic Web to relate the description of a Web service, with descriptions of its domain of operation.

OWL-S adopts the view that the interaction of Web services requires three main operations: the first one is discovery of the providers on the basis of their capabilities, the second one is the management of the interaction on the basis of a partially specified protocol which describes the abstract information exchange between Web services and finally the third operation is the transformation of the abstract information exchanges into message passing or remote procedure call. OWL-S therefore requires that Web services be represented by a specification of their capabilities, a representation of their interaction protocol and a specification of how to compile the actual messages to exchange.

More precisely, a OWL-S Web service is defined as a OWL class with three properties which relate the Web service to the Service Profile, the Process Model, the Service Grounding. The Service Profile provides a representation of the capabilities of the Web service in terms of the input/output transformation that it produces and of a set of non-functional parameters that specify availability, quality and other properties of the service. The Process Model provides a detailed view of process of the Web service from which the requester can derive the interaction protocol with the provider. Finally the Grounding maps the process model into a WSDL specification of how to interact with the Web service. OWL-S reliance on WSDL provides the bridge between the Semantic Web and the Web services infrastructure.

The authors actively participated in the development of OWL-S, and developed algorithms and tools that favor interoperability across Web services. Such tools include capability-based matching on top of UDDI, automatic Web service composition, ontology translation, and developement tools such as the OWL-S IDE. Our experience shows that any Semantic Web services language should be based on the following four principles:

1. Web services should be represented by their capabilities and the results that they produce. In turn this representation should allow means-end analysis and reasoning on what is the best Web service to achieve a given goal, and how to use that Web service achieve that goal.

2. Any language to describe Web services should be independent of any architectural design or implementation. Brokers and mediators should not be defined in the language, rather their capabilities should be descrivable in the language.

3. The language should build on existing Web services standards: rather than defining a new set of languages to describe Web services.

4. The language should build on existing Semantic Web standards, rather than on the "perfect logics" for Web services. Easy interoperability

with the semantic web is paramont to the success of any Semantic Web services language.

The authors will engage in any W3C activity that would lead to standardization of any aspect of Semantic Web services.

References

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[2] David Booth, Michael Champion, Chris Ferris, Francis McCabe, Eric Newcomer, and David Orchard. Web services architecture. http://www.w3.org/TR/2003/WD-ws-arch-20030514/, 14 May 2003. W3C Working Draft.

[3] OWL-S Coalition. Owl-s specification. http://www.daml.org/services/owl-s/1.1/, 2004.