

Semantic Execution Environments for Semantics-Enabled SOA

Ausführungsumgebungen für eine um Semantik erweiterte SOA

Barry Norton, Carlos Pedrinaci, John Domingue, The Open University (United Kingdom),
Michal Zaremba, Innsbruck (Austria)

Summary Much recent work in SOA has concentrated on documenting reference architectures. In this paper we present on-going work on the establishment of such an architecture within the OASIS SEE TC, detailing how the benefits of semantic Web services can be applied to SOA and the role of semantic execution environments, such as the IRS and WSMX.

▶▶▶ **Zusammenfassung** Viel Arbeit im Bereich SOA hat sich in letzter Zeit darauf konzentriert, Referenzarchitekturen zu dokumentieren. Hier zeigen wir die momentane Arbeit an der Aufstellung einer Referenzarchitektur innerhalb des OASIS SEE TC. Detailliert stellen wir den Nutzen von semantischen Web Services für SOA sowie die Rolle von semantischen Ausführungsumgebungen wie das IRS und WSMX dar.

KEYWORDS D [Software]; Semantic Web services, Service-Oriented Architecture, reference architecture

1 Introduction

Recently, a number of initiatives have explored how the application of Semantic Web technologies can support the development of applications from Web services.

These initiatives rely upon semantic descriptions of Web services, forming semantic Web services, founded upon generic ontologies. As a key layer within the Semantic Web stack, ontologies facilitate interoperability, for both humans and machines, by formally capturing a shared domain discourse. The majority of work to date associated with semantic Web services has focused on defining top-level generic ontologies for Web services and on components supporting automated service discovery and/or composition.

In this paper we describe ongoing work in a complementary area. In particular, we focus on how we

are defining a coherent architecture supporting the application of semantic techniques to SOA, building on an existing SOA reference architecture. Our work takes place within the context of the OASIS technical committee on Semantic Execution Environment¹ which has the overall goal of establishing a Semantic SOA Reference Ontology and Architecture.

2 Service-Oriented Architectures

SOA is commonly lauded as a silver bullet for Enterprise Application Integration, inter-organizational business processes implementation, and even as a general solution for the development of all complex applications. Although technology independent, SOA is typically implemented using Web services related

¹ <http://www.oasis-open.org/committees/semantic-ex>

technologies such as WSDL, SOAP and WS-BPEL. In order to define the essence of SOA, transparently from the technologies applied, the OASIS SOA Reference Model (SOA-RM) was defined [1].

SOA-RM is centered around the concept of *service* which is defined as a mechanism for *service providers* to offer certain *capabilities* to *service consumers*. Services are accessed using a prescribed *interface*. A service's implementation is hidden from its consumers except for the *interface* definition and additional information that consumers require for determining the services suitability for fulfilling their needs. The interface is in turn defined in terms of an *information model* and a *behaviour model* which together provide the required information for consumers to invoke services. The RM also includes concepts such as the *real world effect* carried by the

service execution and *execution context*.

SOA-RM defines an abstract framework that captures the main ideas and concepts present in the overwhelming number of definitions, reference models and reference architectures provided by different vendors and organisations so far. As such it attempts to provide a homogenizing model that supports an effective communication among different stakeholders.

3 Semantic SOA

Although useful, SOA-RM is too abstract to effectively support SOA implementations. Some argue for the need for a guideline or methodology for driving the development process, which would indeed be desirable. However, we here want to put the emphasis on the pressing need for machine processable semantics, to support the automation of what actually takes to support an SOA-based system.

For instance, let us imagine the typical travel-booking service which requires the arrival and departure dates as well as the origin and the destination locations. An effective interaction requires service provider and consumer to interpret consistently the dates and locations, which in SOA-RM terms comes down to having a shared understanding of the information model. The need for a shared understanding can quite easily be extrapolated for aspects such as constraints and policies or for the actual invocation of non-trivial services, i.e., the behaviour model.

Typically mismatches are solved by human intervention either aligning data models or developing ad-hoc transformation modules, which limits to an important extent the degree of automation that can be supported. SOA aims to provide a framework for matching needs and capabilities by combining services in a modular and extensible way but automating this process inevitably requires machine processable semantics.

We refer to Semantic SOA as an extension of SOA which makes use of semantic Web services ideas in order to solve the aforementioned limitations, informed by existing approaches such as the Web Service Modelling Ontology (WSMO) [2] and the OWL Services Ontology (OWL-S) [3]. We take an inclusive approach to these existing Web service standards, attempting a union of features, rather than including only their intersection.

Semantic SOA is therefore based on the use of ontologies as a means to support formally defining services in a way that is amenable to automated reasoning. Doing so paves the way for automating the discovery of services that can fulfill certain needs, for automating the integration of heterogeneous data models, for better supporting the combination of services or for the creation of smarter systems, as well as invocation and monitoring of combined services. Finally, using ontologies enables the development of more adaptable and more extensible systems which are in fact some of the main benefits attributed to SOA.

4 Semantic Execution Environment

The Semantic Execution Environment Technical Committee at OASIS is defining a Semantic SOA Reference Ontology as an extension and formalisation of SOA-RM enhanced with additional semantic features. In terms of this ontology it is defining a Reference Architecture (RA); an implementation of this RA is termed a *Semantic Execution Environment* (SEE).

4.1 Reference Ontology

The Semantic SOA Reference Ontology (RO) formalises the notion of service description via an ontological definition. This model includes the concept of Web Service described in terms of its *interfaces* and its *capability*, as in SOA-RM. Also as in SOA-RM the model includes an information aspect, con-

veyed via an *action model*, and a behavioural aspect, conveyed via a *process model*. Unlike SOA-RM, however, the Semantic SOA RM explicitly models the description of contents of the action model via ontological concepts, and their grounding to XML. Similarly there is an explicit semantic model of the capability of the service, including its real world effects, via logical expressions over the information model. The Semantic SOA RO also models two specific uses for process models, as in WSMO [2]: a *choreography* describes the permitted interactions upon a service provider via its client interface; an *orchestration* describes the interactions that the service provider may engage in with other services in order to achieve the service capability.

A significant enhancement of the Semantic SOA RO over the SOA-RM is its explicit model of the information model, required capability and possible interactions of the service consumer. The concept used to contain these descriptions is called a *goal*. By explicitly, and separately, modelling the viewpoint of the prospective service consumer the Semantic SOA RO enables the notion of *mediation*, acknowledged as an important functionality of an Enterprise Application Integration solution and an important part of the Semantic SOA RA. By including mediators also as a first-class member of the RO we allow both *a priori* modelling of the mediation between goals and Web services, and between ontologies as explained later, and also model the communications in which components of the RA engage when these are derived automatically.

4.2 Reference Architecture

The Semantic SOA RA formalises the components that are required to support the execution of applications structured according to a semantics-based Service-Oriented Architecture. The SEE TC has also

approached this description based on existing SWS technologies. Specifically the required functionality for each component is described as a goal in order that this can be met by different services; in some cases a direct implementation, in others an orchestration over the goals associated with other components.

Knowledge Bases: A knowledge base component must be available in a semantics-based SOA to allow storage, retrieval and update of ontological definitions. Goals associated with this component express 'CRUD' operations (create, update, read and delete) over the contents of ontologies.

Reasoning: Reasoning consists of the evaluation of *logical expressions* over ontology definitions. Goals involve checking the satisfiability of the axioms in an ontology, of new expressions, and retrieving the instances that can be bound to an expression with free variables, in order to satisfy it, as a basic form of querying.

Service Registry: A service registry is an essential component in any SOA and provides CRUD goals on service descriptions. In a Semantic SOA service descriptions may be simply represented as ontological instances in the knowledge base, in which case the service registry goals may be met by services that are orchestrations over the knowledge base goals. Alternatively the service descriptions may be specially treated, in which case services may meet these goals with dedicated implementations.

Discovery: Discovery is effectively the query interface over the service repository. One of the strengths of the Semantic SOA approach however is that services are retrieved not merely by syntactic interfaces and unstructured meta-data but via ontological reasoning. This may also involve indirect matches that necessitate mediation, at the data or

process level. The main discovery goal involves submitting a user goal, and optionally the data to be passed in the intended invocation, and receiving back a list of services and/or mediators linking the goal to services via mediation.

Selection: Having obtained a list of matches from the discovery component, the selection component aids in the selection of a service for invocation. The main goal involves submitting the list of services and mediators, together with an ontology that will distinguish between them in terms of degree of match, quality of service profiles, trust/policy matters etc.

Service Execution: Service execution is itself a goal that can be met by one of two services. The first is selected for *atomic* services and composes choreography execution, process mediation, data mediation, grounding and transport. The second alternative for service execution is selected in the case of *composite* services and defers to orchestration.

Choreography Execution: Choreography execution can take two forms. If both the goal and service express separate choreographies then they can each be simulated and process mediation is invoked to find a shared path between them. In case a goal and service are already matched, there may exist a *client choreography*, expressing the desired shared path between (possibly implicit) goal and service choreographies, which is directly executed needing no dynamic process mediation.

Process and Data Mediation: Process mediation is the goal of finding a shared path between goal and service choreographies until a mutual successful end state. Data mediation is the goal of mediating between ontology instances, which is a more general task than transformation services based on syntactic schemas. The input to the goal is a mediator,

a first class concept in the reference model, whose attributes document the entities being mediated between. Data mediation may be described in terms of logical expressions, an abstract mapping language or in terms of the capability of a mediation goal or service.

Communication: Many of the services brokered by a SEE will be built on 'traditional' web service standards and will communicate in terms of SOAP or REST-style encodings of XML-based messages. The mapping between ontological representation of data and its communicable form is called grounding, and the Grounding component has goals for lifting and lowering data, respectively translating into, and out of, ontological form. A SEE also provides an abstraction over different transport mechanisms in the form of Communication goals.

Orchestration Execution: A composite service definition is, as in traditional SOA, executed by an orchestration engine. In the Semantic SOA RA this is abstracted into a goal that can be met by varied services, according to the *process model* chosen, from simple SWS orchestration languages, like the OWL-S process model [3], to a fully fledged semantics-enabled orchestration language like BPEL4SWS [4]. The extensions to such an orchestration language involve taking advantage of the goal-based execution and mediation capabilities of a Semantic SOA.

Composition: Where a goal cannot be met by an existing atomic or composite service, a composition component may be able to generate a new composite service. The goal by which this task is expressed is similar to the discovery goal, but will return a single service if possible.

4.3 Concrete Implementations

Two SEE-compliant implementations currently exist, each with dif-

ferent histories and different underlying implementation technologies but both based on WSMO and sharing a common API based on an encoding of the Semantic SOA RA entry points in WSMO4J. IRS-III [5] is a semantic broker for Web services supporting capability-based invocation. In particular, the IRS allows clients to specify a goal to be achieved and uses ontology based reasoning to locate and invoke relevant services. The IRS also incorporates a number of publishing platforms which are able to automatically generate wrappers for standalone Java and Lisp code and for Web applications (structured via HTTP GET requests).

WSMX² is an open source project aimed at providing an open reference implementation of SEE, developed in close cooperation with the WSMO and WSML working groups. WSMX started being driven by the need to have a semantic execution environment capable of consuming semantic messages, discover semantically described Web services, invoke and compose them for the end-user benefit. On the way, many components have been added, to provide support for SEE tasks.

5 Conclusions

In this paper we have discussed the advantages of the use of Semantic Web technologies in Service-Oriented Architecture and motivated the need for a Semantic SOA Reference Architecture. We have discussed on-going work towards the establishment of such a architecture within the OASIS SEE TC and dis-

cussed two implementations which are compliant with this developing standard.

References

- [1] OASIS SOA Reference Model TC, Official Committee Specification, August 2006. <http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf>
- [2] H. Lausen et al., 'Web Service Modeling Ontology (WSMO)', W3C Member Submission, June, 2005. <http://www.w3.org/Submission/WSMO/>
- [3] D. Martin et al., 'OWL-S: Semantic Markup for Web Services', W3C Member Submission, Nov. 2004. <http://www.w3.org/Submission/OWL-S/>
- [4] J. Nitzsche, T. van Lessen, D. Karastoyanova, and F. Leymann, 'BPEL for Semantic Web Services (BPEL4SWS)', Proc. of the 3rd Int'l Workshop on Agents and Web Services in Distributed Environments (AWeSome'07), 2007.
- [5] J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton and C. Pedrinaci. IRS-III: A Broker-based Approach to Semantic Web Services, Journal of Web Semantics (to appear), Elsevier, 2008.

1 Barry Norton is a Research Fellow in Semantic Web Services at the Knowledge Media Institute (KMi) of The Open University. He is an OASIS SEE TC member, editor of the Semantic SOA Reference Model and an active member of the WSMO Working Group. Address: Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom, E-Mail: b.j.norton@open.ac.uk

2 Dr. Carlos Pedrinaci is a Research Fellow at the Knowledge Media Institute at The Open University. He is actively involved in the standardization of Semantic Web Services technologies as a member of the W3C SAWSDL Working Group, the OASIS SEE TC and the WSMO Working Group. Address: Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom, E-Mail: c.pedrinaci@open.ac.uk

3 Dr. John Domingue has published over 100 refereed articles in the areas of semantic Web services, semantic Web, ontologies and human computer interaction. He is Scientific Director of the SUPER project, and co-chairs the WSMO Working Group and OASIS SEE TC. Address: Knowledge Media Institute, The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom, E-Mail: j.b.domingue@open.ac.uk

4 Dr. Michal Zaremba is a post-doctoral researcher at DERI Innsbruck, leading the Semantic Execution Environment cluster, driving research on the system architecture for Semantic Web Services. He is a contributor to the WSMO and WSMX working groups and co-chair of OASIS SEE TC. Address: University of Innsbruck, STI Innsbruck, ICT – Technologie Park Innsbruck, Technikerstraße 21a, 6020 Innsbruck, Austria, E-Mail: michal.zaremba@deri.org



1



2



3



4

² <http://www.wsmx.org>