

CS565

Simulation, Verification and Composition of Web Services

Michalis Athanasakis
michath@csd.uoc.gr

Munasic Priom
priom_pires@hotmail.com

Kostas Plelis
plelis@csd.uoc.gr

Semantic Web

Introduction

Semantic Web currently is contained in more than four million Webdomains based on a survey done in 2013. But what is exactly the Semantic Web? The Semantic Web is an extension of the web though standards defined by the World Wide Web Consortium (W3C). These standards promote common data formats and exchange protocols on the Web, most fundamentally the Resource Description Framework (RDF). According to the W3C, "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries". The term was coined by Tim Berners-Lee for a web of data that can be processed by machines. While its critics have questioned its feasibility, proponents argue that applications in industry, biology and human sciences research have already proven the validity of the original concept.

Standards

As we said before, the standardization of Semantic Web in the context of Web 3.0 that was defined by Tim Berners-Lee is under the care of W3C. When we talk about standardization we mean the standardization of the formats and the technologies that enable the Semantic Web. These technologies include:

- Resource Description Framework (RDF), a general method for describing information
- RDF Schema
- Simple Knowledge Organization System (SKOS)
- SPARQL, an RDF query language

- Notation3 (N3), designed with human-readability in mind
- Turtle (Terse RDF Triple Language)
- Web Ontology Language (OWL), a family of knowledge representation languages
- Rule Interchange Format (RIF), a framework of web rule language dialects supporting rule interchange on the Web

Applications

All these standards enabled us to enhance the usability and the usefulness of the Web and its interconnected resources through.

- Servers which expose existing data systems using the RDF and SPARQL standards. Many converters to RDF exist from different applications. Relational databases are an important source. The semantic web server attaches to the existing system without affecting its operation.
- Documents "marked up" with semantic information (an extension of the HTML <meta> tags used in today's Web pages to supply information for Web search engines using web crawlers). This could be machine-understandable information about the human-understandable content of the document (such as the creator, title, description, etc.) or it could be purely metadata representing a set of facts (such as resources and services elsewhere on the site).
- Common metadata vocabularies (ontologies) and maps between vocabularies that allow document creators to know how to mark up their documents so that agents can use the information in the supplied meta-data.
- Automated agents to perform tasks for users of the semantic web using this data
- Web-based services (often with agents of their own) to supply information specifically to agents, for example, a Trust service that an agent could ask if some online store has a history of poor service or spamming.

Take notice of the last bullet because we are going to analyze Web-based services much more and all the challenges its simulation, verification and composition of these services might bring. But before that a quick introduction about what exactly a web service is.

Web Services

Introduction

A Web service is a method of communication between two electronic devices over a network. It is a software function provided at a network address over

the Web with the service always on as in the concept of utility computing. The W3C defines a Web service generally as:-

a software system designed to support inter operable
machine-to-machine interaction over a network.

Web services have automated tools to help generate them. There are approached both bottom-up and top-down oriented. Web services can exist without the necessary presence of the semantic web. Though, combining both the technologies provided by the web services and the semantic web it can give a major advantage and enrich the experience of the users.

Semantic Web Services

Semantic Web Services, like conventional web services, are the server end of a clientserver system for machine-to-machine interaction via the World Wide Web. Semantic services are a component of the semantic web because they use markup which makes data machine-readable in a detailed and sophisticated way (as compared with human-readable HTML which is usually not easily "understood" by computer programs).

The reasons that we were not satisfied with the already existing services is that we have yet problems to address that could be solved using the principles of the semantic web.

The mainstream XML standards for inter operation of web services specify only syntactic interoperability, not the semantic meaning of messages. For example, Web Services Description Language (WSDL) can specify the operations available through a web service and the structure of data sent and received but cannot specify semantic meaning of the data or semantic constraints on the data. This requires programmers to reach specific agreements on the interaction of web services and makes automatic web service composition difficult.

Semantic web services are built around universal standards for the interchange of semantic data, which makes it easy for programmers to combine data from different sources and services without losing meaning.

Currently, Semantic Web services platform use Web Ontology Language(OWL) to allow data and service providers to emantically describe their resources using third-party ontologies like Simple Semantic Web Architecture and Protocol (SSWAP) But the creation and usage of OWL is based on other languages like DARPA Agent Markup Language (DAML) or Ontology Inference Layer (OIL). DAML and OIL were combined and DAML+OIL was created.

DAML-S

Introduction

The target is to find ways and technologies that will enable us to describe, simulate, automatically compose, test and verify Web service compositions. For this

to work it is necessary to have a way to describe these Semantic Web services. We come to the conclusion that we are in need of a semantic markup language able to describe the content and the capabilities of these web services. The state of the art include languages like OWL we mentioned before. We are going to focus on the DARPA agent markup language (DAML). After the creation of DAML+OIL we improved and created DAML-S. DAML-S is markup language based on DAML+OIL that was created to specifically describe services. Correspondingly, as DAML was replaced with OWL, so DAML-S was also superseded by OWL-S.

DAML+OIL

DAML+OIL is an AI-inspired description logic-based language for describing taxonomic information. The DAML+OIL is based on top of XML and RDF(S), both defined in the Semantic Web standards. Essentially it is a language with well-defined semantics and a set of language constructs. These include classes, subclasses and properties with domains and ranges. All these can be used to describe a Web domain.

DAML-S

DAML-S is DAML+OIL ontology for Web services. It is also developed under the auspices of the DAML program. There are few different releases as expected. The DAML-S ontology is able to describe a set of classes and properties, specific to the description of Web services. It is separated in parts, the upper ontology of DAML-S is created by a service profiler used for describing service advertisements, another process model to enable the description of actual programs that execute each service and a service grounding for describing the transport-level messaging information associated with the actual execution of the program.

DAML-S Examples

DAML-S process model is able to describe two different types of processes, *atomic* and *composite*. Also it allows the creation a type of *simple* process, which is a description of a view or abstraction of a atomic or composite process to which expands.

```
<daml:Classrdf:ID="Process">
  <daml:unionOfrdf:parseType="daml:collection">
    <daml:Classrdf:about="#AtomicProcess"/>
    <daml:Classrdf:about="#SimpleProcess"/>
    <daml:Classrdf:about="#CompositeProcess"/>
  </daml:unionOf>
</daml:Class>
```

An atomic process is a non-decomposable Web-accessible program. It is executed in a single http call and returns a response. An example of a atomic

process is the **LocateBook** that either returns information about the book or nothing.

```
<daml:Classrdf:ID="LocateBook">
  <rdfs:subClassOf rdf:resource="&process;#AtomicProcess"/>
</daml:Class>
```

On the other hand a composite process is composed of other composite or atomic processes. These composing processes are connected using control constructs, such as *sequence*, *if-then-else*, *while*, *fork etc.* An example of a composite process is the **Find-n-Buy** service that contains the **LocateBook** along with some other services.

```
<daml:Classrdf:ID="CompositeProcess">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Classrdf:about="#Process"/>
    <daml:Restrictiondaml:minCardinality="1">
      <daml:onPropertyrdf:resource="#composedOf"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>
```

Also, each process is able to has a set of properties, called parameters and can be either be input or output. In the case of the **LocateBook** an input parameter might be something like the name of the book.

```
<rdf:Propertyrdf:ID="bookName">
  <rdfs:subPropertyOf rdf:resource="&process;#input"/>
  <rdfs:domainrdf:resource="#LocateBook"/>
  <rdfs:rangerdf:resource="&xsd;#string"/>2
</rdf:Property>
```