Service Description Ontologies

Julia Kantorovitch
VTT Technical Research Centre of Finland, Finland

Eila Niemelä
VTT Technical Research Centre of Finland, Finland

INTRODUCTION

Services can be Internet-based e-commerce services, business services that abstract company-level interactions, or any other software services that are provided by surrounding devices that are mobile or embedded in nearly any type of physical environment (e.g., home, office, or cars). In brief, services are ubiquitous and executed in heterogeneous environments.

Surrounding the definitions and technologies that describe services, there are some important features that are in common. First, services always have some actions that are performed by an entity, possibly on behalf of another. Second, there always exists service interaction, including a service provider, service requestor, and service registry. Finally, services have inherent value that is transferred from the service provider to the service requestor as a result of the service's execution.

To invoke and operate a service in the most efficient way, the service is to be described via essential types of knowledge: a) what the service requires from the user/agent(s) and then provides for them; b) where and when the service is available; c) what quality level is to be guaranteed; d) how to access and interact with the service; and e) what access rights are granted over the service.

An accurate service description, including the specifications of functional and nonfunctional properties, benefits and facilitates several important activities, such as service discovery, service composition, and service administration, including the monitoring and controlling of the service's execution. However, due to the diversity of service contexts, service technologies shall be generic and adaptable to different domains and heterogeneous environments. Service description ontologies solve this problem by enabling a rich representation of services and a common understanding about their respective features. The use of ontologies enables computational entities and services to have a common set of concepts and properties for representing knowledge about a domain of interest. The deployment and customization of existing and emerging service systems can also be considerably facilitated by a common set of ontologies that is developed in order to describe service semantics.

BACKGROUND

The studies on the data schema in XML, RDFs, OWL, and service description languages all form the basis for defining the service semantics. The eXtensible Markup Language (XML), as defined by the World Wide Web Consortium (W3C), is a well known, and industry accepted, way for representing flexible information. It is used to create information objects consisting of elements encoded by tags and attributes. XML schemas express shared vocabularies and allow machines to carry out those rules that are established by people.

The resource description framework (RDF) and Web ontology language (OWL) are built on XML and facilitate greater machine interpretability of content by providing additional vocabulary along with formal semantics.

XML, SOAP, WSDL, and UDDI form the core of Web service standards. Simple object access protocol (SOAP) is a lightweight protocol for exchanging XML-based information in a distributed environment. Web service description language (WSDL) is an XML format for describing services as a set of endpoints operating on messages. The operations and messages are described abstractly, and then bound to a concrete protocol and message format in order to define an endpoint. UDDI (universal description, discovery, and integration) is concerned with the publishing and discovery of Web services.

Service Level Agreements (SLAs) related quality information is described by:

- Web service level agreement (WSLA) language (Ludwig et al., 2002), which is based on an XML schema and defines the SLAs in three parts; i) contractual parties, ii) the characteristics of the service and its observable parameters, and ii) obligations to various guarantees and constraints that may be imposed on the SLA parameters, or
- Web services offerings language (WSOL) (Tosic, Paguredk, Patel, 2003), which is a formal specification language for defining the functional and QoS constraints and access rights for Web services. WSOL is XML-based and compatible with WSDL.

SERVICE ONTOLOGIES

In the following, the existing semantic approaches for describing the services, including their functional capabilities, QoS, and context are compared in Tables 1-3. This comparison aims at identifying the main benefits and shortcomings and the missing aspects in the service description ontologies that serve as a basis for the conclusion and future research trends that are to be identified in the remainder of the paper.

Service Functionality Descriptions

Several service ontologies contribute to the service creation, provision, and execution, to a varying extent, by using different description languages. Table 1 compares the main properties of four existing service description ontologies from the viewpoint of their completeness to describe the service related aspects. Web ontology language for services (OWL-S), Web service modeling ontology (WSMO) (Roman, Keller, Lausen, Lara, Bruijn, Stollberg, et al., 2005) and Internet reasoning service (IRS) (Domingue, Cabral, Hakimpour, Sell, & Motta 2004; Motta, Domingue, Cabral, & Gaspari, 2003;) provide specific ontology building blocks for particular purposes of use. Conversely, METEOR-S (LSDIS Lab, 2005) is an approach that targets the extension and integration of the existent Web services and semantic Web technologies. OWL-S is the most widely used approach concerning service semantic modeling. OWL-S combines the expressiveness of description logics, as it builds on OWL. The WSMO is a relatively new effort and is based on the Web service modeling framework (WSMF) (Fensel, Bussler, Ding, & Omelayenko, 2002). All of the approaches provide for specific advantages that are missing from another approach. However, the approaches do not individually provide complete description support for service semantics (see Table 1, Difference row).

Service Quality Descriptions

Table 2 compares a set of QoS ontologies that address their benefits and shortcomings. The difference that is in focus results in different ontology layers. A lack of completeness is common for all the approaches; only one or a few qualities are considered, and the vocabulary or/and metrics are missing. Moreover, there is no support for making tradeoffs between quality attributes or managing QoS at run-time.

In Zhou et al. (Zhou, Chia, & Lee, 2004), the QoS ontology with the three layers covers the matchmaking, QoS property definition layer with domain and range constraints, and metrics with the measurement details. A drawback of this approach is that the proposed ontology is rather limited, while the QoS ontology vocabulary is absent. The framework presented in Maximilien and Singh (2004) is based on agents that enable dynamic Web services selection. On the other hand, work in Tosic et al. (Tosic, Esfandiari, Pagurek, & Patel, 2002) has focused on metrics, measurement units, and currencies to support QoS semantic management. An extended matchmaking mechanism with the concept of the service broker is addressed in Tian et al. (Tian, Gramm, Naumowicz, Ritter, & Schiller, 2003). It also classifies the QoS parameters into network-related and server/client-related parameters. The MOQ (mid-level ontologies for quality) framework (Kim, Sengupta, & Evermann, 2005) aims to minimize the ambiguities in QoS evaluations by defining the ontologies for the requirements, measurement, traceability, and quality management.

Service Context Descriptions

Concerning the contextual characteristics of services, several ontologies have been designed, some of which are more elaborate and others more succinct, depending on their scope. The most popular of these are context ontology language (CoOL) (Strang, Linnhoff-Popien, & Frank, 2003), context broker architecture (CoBrA) (Chen, Finin, & Joshi, 2003), service-oriented context-aware middleware (SOCAM) (Gu, Wang, Pung, & Zhang, 2004), COntext MAnagement oNTOlogy (COMANTO) (Strimpakou, Roussaki, Pils, & Anagnostou, 2006), and the standard ontology for ubiquitous and pervasive applications (SOUPA) (Chen, Finin, & Joshi, 2005). Their main characteristics are shown in Table 3.

Most of the approaches presented address the vocabulary ontology needs in the domain of pervasive computing. The context ontologies that are designed include a set of vocabularies for describing people, agents, and places, as well as a set of properties and relationships that are associated with these basic concepts. However, rather little emphasis is placed on services, including their functional properties and related aspects, such as user interfaces and devices on which these services are deployed, along with temporal contextual information. No attempts have been made to align service and context ontologies.

Tools for Describing Service Semantics

Numerous freeware and commercial tools to support the development and use of ontologies are currently available: SWOOP is a hypermedia based OWL ontology editor; Protégé is a free, open source ontology editor and knowledgebase framework; TopBraid Composer™ is an enterprise class platform. The advancement in these tools has greatly improved the ability to test and build ontologies from scratch or to reuse existing ontologies.

Application programming interfaces (APIs) for ontology languages provide programming language dependent means to load ontologies, manipulate the ontology classes and relations, perform reasoning, and provide persistent storage for

Related Content