Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 127 (2018) 398–407

Procedia Computer Science

www.elsevier.com/locate/procedia

The First International Conference On Intelligent Computing in Data Sciences

Exploiting Semantic Web Services in the Development of Context-Aware Systems

Hatim Guermah[a,*], Tarik Fissaa[a], Hatim Hafiddi[a,b], Mahmoud Nassar[a]

[a]*IMS Team, ADMIR Laboratory, ENSIAS, Rabat IT Center, Mohammed V University, Rabat, Morocco*
[b]*ISL Team, STRS Lab, INPT, Rabat, Morocco*

Abstract

Nowadays, web services have become increasingly used and deployed in view of the benefits that they offer. Thus, the increasing of providers and service offerings as well as the features of mobile devices has made the selection of relevant services a difficult operation. In addition, the information related to the user, called context / situation, became very useful for the implementation of the eloquent choice of the sought-after functionalities. In this paper, we present a comparative study of semantic web services and service discovery approaches. Then we propose the model of a context-aware semantic service with their process of implementation. We also highlight the discovery algorithm based on the concept of context / situation.

Keywords: Semantic Web service, Discovery and selection, Context, Situation, Qos Properties ;

1. Introduction

To describe a futuristic vision of computing, Weiser [1] presented the premises of a new paradigm of interaction, which focuses on the purpose of interaction rather than the interaction itself. He envisioned a largely populated world of IT equipment that would be connected in large-scale networks and interfered autonomously and transparently to perform various tasks dedicated to the user. This concept is called ubiquitous computing. Ubiquitous computing creates an environment characterized by a set of natural elements and materials related to the

* Corresponding author. Tel.: +212 537 778 579 ; fax: + 212 537 777 230.
E-mail address: Hatim.Guermah@um5s.net.ma.

1877-0509
10.1016/j.procs.2018.01.137

user through which systems interact and offer relevant services with computing, storage and communication capabilities at any time. In such a perspective, Application developers can make services ubiquitously available to users by moving their interfaces to the nearest appropriate input or output device [2]. These services must be context-aware, dynamically adaptable to the user's profile and environment, promise networking development while defining the roles, responsibilities and characteristics of the work environment. In addition, the use of such services should allow better opening of vital sectors (health [19], education, administration and tourism) to the public even the most remote or poor.

In such a perspective, Web services have the advantage of being based on a set of standards and norms that allow many benefits over other types of distributed computing architectures such as loosely coupled systems can be deployed and virtually platform-independent. However, the use of the WSDL language, as well as service orchestration languages such as Business Process Execution Language (BPEL) and WSFL (Web Services Flow Language) or service choreography languages as The Web Service Choreography Interface (WSCI) and Web Services Choreography Description Language (WS-CDL) do not describe in detail the capabilities of a service. Indeed, we can find two WSDL interfaces that have the same input / output parameters that offer different functionality or vice versa. As a result, the researchers have thought of introducing various semantic specifications, called semantic Web services, which offer formal descriptions of the capabilities of a web service based on its IOPE (Input, Output, Precondition and effect) functional properties and which allow the automation of various activities related to the lifecycle of a web service such as selection or composition.

This paper focuses on how semantic Web services are modeled and managed and how they could satisfy the requests based on context's / situation's user. The outline of this paper is as follows. In the next section we compares some related work about semantic web Services and discovery approach. We present, in section 3, our extension of the semantic context-aware service and its implementation methods. Sect. 4 presents our proposal algorithms for context-based discovery and selection services. Finally, we conclude the paper in Sect. 5, with plans for future work.

2. Related works

2.1. Semantic Web services

Semantic Web services [3] are at the convergence of two domains: the Semantic Web and Web Services. The Semantic Web is interested in providing semantic interoperability of content while Web services are primarily concerned with the syntactic interoperability of data exchanges to make the Web more dynamic. Semantic Web services are therefore interested in procreating a semantic Web of services whose interfaces and properties are described in a machine-readable way. We will expose a semantic annotation of the WSDL standard, called SAWSDL, and two service ontologies called WSMO and OWL-S.

2.1.1. SAWSDL

SAWSDL (Semantic Annotations for WSDL and XML Schema) [4] is a W3C recommendation, which facilitates the annotation of WSDL 2.0 descriptions while supporting WSDL 1.1. It aims to support the use of semantic concepts while ignoring the semantic representation language. The goal of SAWSDL is to define how an annotation should be performed, while leaving the choice of the language used for the semantic description and providing the mechanisms for attaching concepts described in ontologies to the annotations of the WSDL descriptions. These annotations are defined using three SAWSDL attributes:

- ModelReference: used to link WSDL elements
- LiftingSchemaMapping and LoweringSchemaMapping: specify the correspondences between the semantic data and the XML structures.

2.1.2. OWL-S

OWL-S [5] known as DAML-S, is an ontology, based on the OWL language, used for the construction of semantic web services and the description of Web services in an understandable, unambiguous and easily