



Integrating a Context Model in Web Services

Bouchra Soukkarieh, Florence Sèdes

► To cite this version:

Bouchra Soukkarieh, Florence Sèdes. Integrating a Context Model in Web Services. IEEE International Conference on Web Services (Work-in-Progress) 2007, Jul 2007, Utah, United States. pp.1195-1196, <10.1109/ICWS.2007.110>. <hal-03773767>

HAL Id: hal-03773767

<https://hal.science/hal-03773767v1>

Submitted on 12 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Integrating a Context Model in Web Services

Bouchra Soukkarieh, Florence Sedes

IRIT, Université Paul Sabatier, 118 Route de Narbonne, F-31062 TOULOUSE

{sokarieh, sedes}@irit.fr

Abstract

Nowadays, with the great diffusion of mobile technology, and ubiquitous systems, the context has become the ear and the eye of information systems. These systems are more and more based on the usage of Web Services. The classical architecture of these Services that allows an interoperable interaction between service users and providers does not take in account context adaptation.

In this article, we aim to integrate context adaptation within the classical architecture of Web Services, adding to it dedicated components that would return to nomadic users a list of Web Services that are adapted not only to his profile but also to his context.

1. Introduction

The usage of *Web services* (WS) within the *Web-based Information Systems* (WIS) is increasingly frequent. A WS is a software system designed to support interoperable application to application interaction over the web. The classical architecture of WS is composed of three entities (provider, user and registry) and is based on three standards (WSDL, UDDI and SOAP). Despite the importance of context in WIS where the user isn't any more sitting at his office but is moving around different environments, WS architecture does not take into account the idea of context adaptation.

In our research works, we aim to integrate context adaptation within the classical architecture of Web Services, adding to it dedicated components that would return to nomadic users a list of WS that are adapted not only to the user characteristics (profile) but also to the context of his environment (localization, device, etc).

The rest of this paper is as follows. In section 2, we take a rapid tour at some works in the field of context adaptation within WS. In section 3, we detail our proposal which aims to extend the classical architecture of WS in order to take into account the context.

2. Related works

In this section, we take a rapid tour at some works in the field of context adaptation within WS.

1) [5] are interested in the impact of the context on a medical application. A platform (SECAS) was developed

in order to adapt the application to various context components: data, services and presentation.

2) [3] present an information system for tourism (CATIS) which enables the adaptation of mobile devices in terms of content and presentation. The application takes into account the user preferences, his localization and the type of his terminal.

We note that these researches works present specialized programmed services that are adequate to meet the needs of particular users. None of these works deliver to the user a list of services adapted to his context. It is in this direction that our work is heading.

3. Our contribution

In this section, we present our idea of extending the classical architecture of WS in order to take into account the context. Then, we introduce the context representation for both services and users using the CSCP format.

3.1. Architecture of the WS containing context

Traditionally, the architecture of WS (Figure 1) is composed of three entities: the *service provider* builds the service and publishes its description in a *registry*. The *user* needs are translated into requests that are carried on by the WS registry. Once the service is found, the user will obtain direct interaction with the service.

Our contribution aims to add to this architecture an adaptation layer containing various components dedicated to the context adaptation. This layer is centralized in the registry. We present, in Figure 1, different steps that we propose to the integration of the context in the classical architecture of WS. The different steps are :

- 1) WS Providers describe their services using metadata. These metadata describe the criteria of adapting services.
- 2) A Service Profile Manager (SPM) component allows, on one hand, to extract the metadata describing WS adaptation criteria and on the other hand, to store them in the database (Services Profile (SP)).
- 3) WSDL Description is then transmitted to the registry without the metadata as in the classical architecture.
- 4) The user launches his query to the registry (with format SOAP).
- 5) The Context Manager (CM) captures the context.
- 6) CM stores the context into the database as an XML document.
- 7) The result of the query (the list of services answering

the user request) is transferred from the registry to the CM. 8) The CM requests from the SPM the service profile that meets the user query. 9) The CM recovers the stored context and then compares two profiles (services and context profiles). 10) The CM sends to the user the list of adapted services to the context. 11) The communication between the user and the provider is done in a traditional way via SOAP.

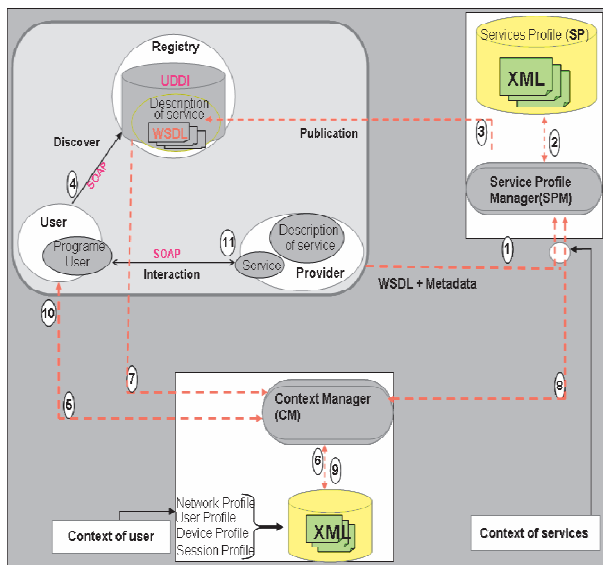


Figure 1. Architecture of WS integrating context

The adaptation layer contains two main elements:

- 1) SPM which is responsible to manage WS context.
- 2) CM which is responsible to manage user context.

So, to return the user a list of services adapted to his context, we must take into account not only user context but also the services context. The service context is expressed directly by the service provider. The user context is obtained in an explicit and implicit way.

3.2. Context profile representation

Researchers in the field of the context adaptation did not reach yet a generic definition of the context concept. The most accepted definition of context is: "context is any information that can be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application, including the user and the application themselves. Context is typically the location, identity and state of people, computational and physical objects." [1]

Researchers in this field classify contextual information according to their particular application. In our architecture, we base on work of [4] by adding parameters related to the session and to the network. Therefore, we defined four axes that aim to cover all the context parameters: the *user characteristics* which are composed of *static characteristics* (last name, first name,

etc) and of *evolutive characteristics* defined by his environment (his localization, time, etc) and his preferences, the *device characteristics* (material, software), the *network characteristics* (standard, flow) and the *session characteristics* (date, hour, state, etc).

To represent the contextual information, the majority of researchers as used the format *CC/PP (Composite Capability/Preference Profiles)* [6]. We did not use it to represent our context because of its lack of structured functionalities. Its two leveled strict hierarchy is not appropriate to capture structures of complex profiles. We use the *CSCP (Comprehensive Structured Context Profiles)* format to overcome the structure deficits of CC/PP. CSCP provide a multileveled structure which allows the representation of all the types of contextual information [2]. According to this format, our context profile is composed of a user profile, a session profile, a terminal profile and a network profile.

4. Conclusion

In this article, we have extended the classical architecture of Web Services by adding to it special components that aim to deliver a list of adapted Web Services to user preferences and contexts.

In the near future, we first aim to specify the algorithm which will help us to match the two contexts (user and services). Then, we plan to implement our architecture that integrates the context in the classical architecture of WS.

5. Reference

- [1] A. K. Dey, and G. D. Abowd, "Towards a Better Understanding of Context and Context-Awareness", CHI 2000 Workshop on the What, Who, Where, When, and How of Context-Awareness, The Netherlands, April 2000.
- [2] A. Held, S. Bouchholz, and A. Shill, "Modeling of Context for Pervasive Computing Application", In Proceedings of the 16th world Multiconference on Systemics, Cybernetics and Informatics (SCI), Orlando, FL, USA July 2002.
- [3] A. Pashtan, A. Heusser, and P. Scheuermann, "Personal Service Areas for Mobile Web Application", IEEE Internet Computing, December 2004, pp. 34 -39.
- [4] C. Lopez-Velasco, "AWSDL : une extension de WSDL pour des Services Web adaptés", INFORSID 2005, Grenoble, may, 2005, pp. 133-148.
- [5] T. Chaari, F. Laforest, and A. Flory, "Adaptation des applications au contexte en utilisant les Services Web: Le projet SECAS", Second Francophone workshop: Mobility and Ubiquity 2005, Grenoble, France 31 may 2005.
- [6] W3C, (CC/PP) Composite Capability/Preference Profiles: The last version: <http://www.w3.org/TR/CCPP-struct-vocab/> (January 2007).