ABSTRACT

The realization of ubiquitous networks brings new challenges to application development. In this kind of network, services and, more specifically web services, have been used to provide the functionality required by its users and applications. In such environments features like automatic service discovery and composition are of need. This paper reports the ongoing work towards an approach for dynamic service discovery and composition. This approach is based on modeling services as entities independent of concrete implementations. This abstraction allows applications and users to subscribe to a desired functionality and later, the supporting platform discovers the service or a composition of services to provide that functionality.

1. INTRODUCTION

With the move towards the widespread deployment of ubiquitous networks new challenges have emerged. Among then we can mention the necessity to tackle with the increasing number of services being offered in the context of those networks. For instance, in an environment with a high number of services, the ability to manually select which one is (best) suitable for one’s needs diminishes. Moreover, the growing complexity, amount of distribution and number of users of these services reinforces the necessity for an underlying support which can provide means to a rapid deployment, discovery, selection, invocation and composition of services. This support can be provided by service platforms [1].

Envisioning a large-scale interoperability of the services, a service platform can make use of an unambiguous, machine-understandable representation of the properties, capabilities, effects and interfaces of the services. One approach to provide this kind of representation is by means of semantic annotations of the concepts present in the service’s descriptions.

Another aspect that emerged with the advent of the ubiquitous networks is the context-awareness. In our daily life we use implicit information to guide our interactions and to better understand the surrounding environment. This kind of situation is very helpful in human-human conversations. Contrarily, when we get into traditional human-computer interaction, human users have to, explicitly, input the required data for computation to take place. Improvement in computer’s ability to gather and “understand” the surrounding context increases the richness of human-machine communication and the possibility for more useful computational services [5]. With the dissemination of ubiquitous computing, where user’s context changes rapidly, the use of context is increasingly important and a new class of applications, referred as context-aware applications, is emerging.

Context-aware applications also bring to light several architectural design challenges. Issues such as context gathering, context storage and monitoring, context modeling and interpretation, and context-aware service discovery are examples of the new kind of requirements that have to be treated by the service platform.

This paper aims at presenting the current status of the work towards the proposal of an approach for dynamic discovery and composition of services in a scenario of a context-aware services platform. This paper is organized as follows. Section 2 presents the service platform used as reference for testing the proposed approach and its running context. Section 3 explains the approach for dynamic service discovery and composition and section 4 concludes this paper and identifies the direction of the remaining work.

2. REFERENCE SERVICE PLATFORM

The reference platform used in this work has been presented in [3]. The platform is a context-aware service platform meaning that the user’s context, gathered by context providers is taken into account for several tasks of the platform such as service discovery, selection and composition, among others. For the understanding of this paper, a simplified version of the platform’s architecture is depicted in Figure 1.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CoNEXT’06, December 4–7, 2006, Lisbon, Portugal.
© 2006 ACM 1-59593-456-1/06/0012 5.00.
Following the architectural components presented on the figure, the client applications interact with the platform by subscribing to the services made available by the service providers. The service descriptions are stored in the service description registry. The platform receives, from the context providers, contextual information regarding the client applications’ users. The platform is responsible for monitoring the received contextual information and, when the context specified in the subscription is reached, the platform triggers the execution of the services, receives their outcome, perform the transformation (if necessary) and return the final result to the client application.

3. APPROACH FOR SERVICE DISCOVERY AND COMPOSITION

Current approaches for service composition compare inputs, outputs, pre-conditions and post-conditions to match the appropriate component services. This comparison is commonly made at syntactical level, therefore raising semantic incompatibility. Our approach for service discovery and composition starts with defining a high level description of the service. Hereafter we name this high level description of a service as Abstract Service. The motivation for the use of abstract services is based on the assumption that a user of a service, being the user a human or an application, is interested in a given functionality provided by this service and not on how this functionality is implemented. For instance, a user is interested on receiving the weather conditions of the route he/she will take in a trip and not on whether it will be necessary to use more than one weather forecast service because the route includes several countries.

In the current status of this work, the representation of the abstract service is made based on OWL-S service ontology [4]. The ServiceProfile class is used to describe the abstract service capability, i.e., what it does. The ServiceModel class describes how the abstract service works, i.e., the involved subtasks and their relations. Since the abstract service does not directly bind to any actual concrete service, the OWL-S ServiceGrounding class that specifies the details of how to access a service (communication protocol, message format, URI, among other details) is not necessary.

After the abstract service definition, client applications can subscribe to these abstract services. In this subscription, a specific user context is defined. When this defined context is reached, the subscribed service is then triggered. For this, the platform requires contextual information supplied by the context providers.

A strategy has been defined that, when a client application subscribes to an abstract service, the platform immediately searches for concrete services that can provide the functionality described by the abstract service. When the service (or a composition of services) is discovered, the platform “pre-selects” it as the concrete realization of the abstract service. When the triggering context is reached, the platform checks whether the pre-selected service is still suitable. If the check is successful, the platform evokes the service. Contrary, if the platform verifies that the pre-selected service is no longer suitable it searches for another service(s).

4. CONCLUSION AND FUTURE WORK

In this paper we presented the current position of our work of a proposal for dynamic service discovery and composition for context-aware services platforms. This proposal makes use of task ontologies to provide an abstract definition of a service. This abstract definition is used to supply the platform with the necessary information to allow the dynamic discovery and composition of concrete services.

The current status of the research is the definition of the underlying context-aware services platform. Moreover, the preliminary examples of the abstract services have been created based on OWL-S. Since the abstract services do not bind directly to any actual service, such as a Web Service, the OWL-S Service Grounding class is not necessary.

The next steps of the work are, among others, (i) to create a complete prototype of the platform to test the approach; (ii) to investigate other languages and evaluate the suitability of them to represent the task ontologies; (iii) to evaluate the inference possibilities of the semantic annotations; define a methodology for the design and deployment of context-aware services; (iv) to generalize the approach and evaluate against other platforms/scenarios.

5. ACKNOWLEDGMENTS

The present work has been done in the Freeband Communication project A-Muse (http://a-muse.freeband.nl). Freeband Communication is sponsored by the Dutch government under contract BSIK 03025. This work is partially funded by the Amigo project in Ambient Intelligence. The Amigo project is funded by the European Commission as an integrated project (IP) in the Sixth Framework Programme under the contract number IST 004182.

6. REFERENCES


