Towards Wide Area Context-Aware Environments

Yong Liu and Kay Connelly
Department of Computer Science, Indiana University
{yonliu, connelly}@cs.indiana.edu

Abstract

In the vision of ubiquitous computing, there are billions of context sources that continuously publish their contextual information, and even more user agents that search for and consume such information. These context producers and consumers form a large-scale, wide-area, context-aware network which is both dynamic and heterogeneous. Existing service discovery mechanisms do not address the unique challenges imposed by such environments. In this paper we identify these challenges and propose a large scale context service discovery infrastructure for context-aware computing. We describe an approach for context service discovery and description that addresses these challenges.

1. Introduction

Context-aware computing works towards using context of the physical space, such as location, to drive behavior in the virtual realm. From one of the first context-aware applications, the Active Badge Location System at AT&T labs [1], to the recent MyCampus project at CMU [2], most context-aware systems focus on context within in a limited geographical space, such as an office, a building, or a university campus. Typical *local-area* context-aware systems share the following characteristics:

- Context sources and consumers are within each other's spatial proximity.
- Context types and context sources are pre-defined in the system at design time.
- A central registry stores all of the information about context sources.
- They are not scaleable.

The vision of *wide-area* context-aware computing, where anyone can retrieve relevant context regardless of where they are, does not have the same characteristics. Imagine a tour guide application designed for world travelers which is supposed to automatically access a city's restaurants and sightseeing information according to the traveler's current location or itinerary. In this scenario, the sources of context may be located at a far distance from the traveler and new context sources may be added and moved dynamically. Another example is SmartContacts [3], an application that helps people determine if it is an appropriate time to make a call by providing the caller with the receiver's current context. In this scenario, individual cell phone users act as context sources. Due to the number of cell phones, a central registry is not feasible, making scalability a central issue.

Previous work in context-aware computing has largely focused on local area context-aware systems. One obstacle for developing a wide area context-aware system is the difficulty in building a large scale context service discovery infrastructure for ubiquitous computing environments. This task has several challenges, including scalability, automatic discovery, fault tolerance and the heterogeneity and dynamicity of context sources.

In this paper we propose a large scale context service discovery infrastructure leveraging Semantic Web and information retrieval technologies. This infrastructure reaches a large community of users through the well established Internet and World Wide Web, enables individual users and business entities to advertise, discover, and consume rich contextual information services, and supports rich featured context-aware applications.

In section 2, we describe the challenges for context source/service discovery in ubiquitous computing environment. We introduce our approach to support context service discovery in wide area context-aware systems in section 3 and describe the workflow of the context service discovery in section 4. Section 5 reports our current progress, and we conclude in section 6.

Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)
0-7695-2520-2/06 $20.00 © 2006 IEEE

2. Wide Area Context Service Discovery: Key Challenges

An emerging ubiquitous computing paradigm decouples devices from users and views them as agents that perform tasks on behalf of users. These agents should be aware of their context and adapt themselves to context changes without human intervention. In a local area context-aware system, all of the context sources can be pre-defined at design time and accessed by user agents via a central registry at run time. Hence, source discovery becomes a trivial problem. This simple solution does not work in wide area context-aware systems. A few of the most important challenges that need to be addressed:

- **Scalability:** The infrastructure should support a large volume of context sources, where both the context source and consumer may be geographically distributed in a wide area.
- **Fault Tolerance**: The infrastructure should be tolerant of faults.
- **Heterogeneity**: The infrastructure must be able to handle heterogeneous context sources provided by different organizations and individual users. Quality and even availability of individual context sources cannot be guaranteed.
- **Dynamicity:** Context services may join and leave the system at anytime. When a known context service is gone, user agents need to find alternative services, so as to accomplish their tasks.
- **Automated Discovery**: Context-aware applications should be able to discover and process context sources without human intervention.

3. Our Approach

In this paper, we propose a wide area context service discovery infrastructure. This infrastructure adopts Semantic Web and information retrieval technologies to answer the challenges posed by wide area context-aware systems. We decompose the wide area context service discovery problem into *service discovery* and *service description*.

3.1. Service Discovery

One paradigm of context-aware computing is that individual computing entities publish information about their status or surrounding environment. The entity may be a dedicated sensor (e.g. a thermometer), an electronic mobile device (e.g. a cell phone), or even an information portal (e.g. an Internet portal site). Any particular entity may provide a single type of context, or several hundred types. These context sources are highly distributed and heterogeneous, similar to today's Web pages on the Internet. Just as there is no central repository for Web page authors to register their pages, the organization of context services should not utilize a monolithic, centralized registry, but be similar to today's World Wide Web. Such an organization will be scaleable, fault tolerant, and allow for dynamic and heterogeneous context services. The main hurdle, thus, is to enable automatic discovery of context services.

Today's WWW community solves the information discovery issue by adopting search engines. People use search engines to find Web pages they need without having to memorize the addresses of Web pages. Since search engines have been central to the success of the WWW and have profoundly changed the way we access information, it is reasonable to believe this idea is appropriate for automatically discovering context services in wide area context-aware systems.

In a ubiquitous computing environment, agents act on behalf of users. Similar to human users, the agents should not be required to know all of the available context sources and where to find them at design time. Indeed, this is not possible when using dynamic context sources. However, agents can search for context sources on demand via a search service.

Based on this vision, we propose a context source discovery mechanism leveraging the well established WWW infrastructure and information retrieval technologies. Context providers publish their context service description documents to an arbitrary Web server. The documents are crawled and indexed by Internet search engines, and hence made available to the public. User agents search for needed context services via search engines and use them to perform assigned tasks.

3.2. Service Description

To automatically retrieve and consume context services, user agents in a wide area context-aware system need to understand the meaning of functional descriptions of each context service. WSDL, as a widely used Web service standard, is designed to describe the interface used by each service [11]. However, WSDL by itself is unable to express the semantics of Web services. Current research shows ontologies are a promising instrument for modeling contextual information due to their formal expressiveness and possibility for reasoning [5].

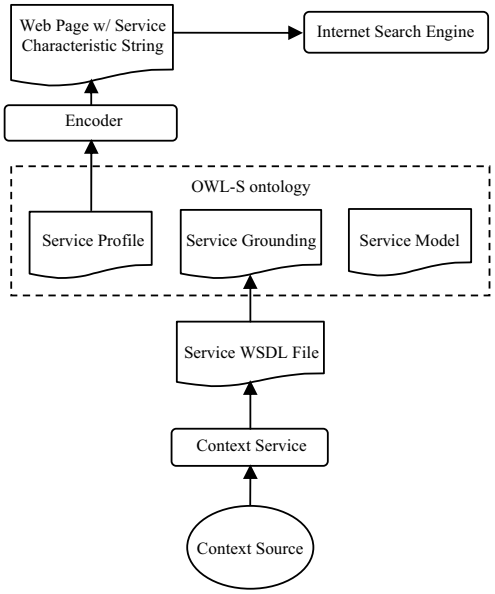

Figure 1. Context Provider-Side Workflow

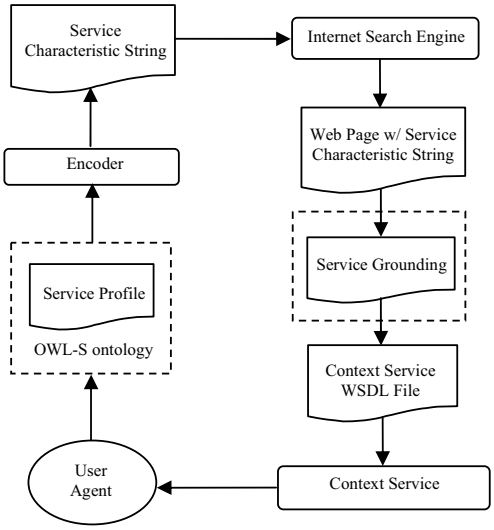

Figure 2. Context Consumer-Side Workflow

In our context service discovery infrastructure, we use OWL-S, a tool for defining and instantiating ontologies, to describe the functionality of context services. The OWL language is a Semantic Web language used by computing entities to process the content of information instead of just presenting information to humans [6]. OWL-S is derived from OWL and used to formally describe services [7]. Service descriptions written in OWL-S usually contain the following three components:

1. *Service profile* tells what the service does.
2. *Service model* gives a detailed description of a service's operation.
3. *Service grounding* provides details on how to interoperate with a service.

In our infrastructure, a context service provider generates service profiles and publishes them on a Web server. Based on the information in a service profile, service-seeking agents determine whether the corresponding service meets its needs or not. If it does, the agent will access the service according to the details provided in the service grounding.

4. Description of a Wide Area Service Discovery Infrastructure

The proposed large scale service discovery infrastructure consists of context service providers and consumers, context service ontologies, third party Internet search engines and Web servers.

Figure 1 shows the workflow on the side of the context provider. Each provider publishes its service WSDL file to an arbitrary Web server. Each service provider also generates the OWL-S descriptions from the WSDL file and publishes them to WWW. Currently there are a few toolkits available to help create the OWL-S description from the corresponding WSDL file [8, 9].

Although current Internet search engines do crawl and index Semantic Web pages (e.g., OWL-S documents), they usually throw away the semantics by removing markups from the documents during the tokenization process. To keep the semantics of the OWL-S description documents and use them in the later service discovery stage, service providers need to generate service characteristic strings from the service profile. These strings represent the functionality of the service. There are existing Semantic Web document encoders that help with this process [10].

After the service characteristic strings are generated, they are written into a Web page along with the address of the OWL-S description documents of the service and published to the WWW. After these strings are crawled and indexed by Internet search engines, they can be used as keywords to search for context providers.

Figure 2 shows the workflow on the side of the context consumer. Once a user agent is assigned a task, it decomposes the task and identifies context types needed to accomplish the task.[1] Once the context types are identified, the user agent generates the corresponding service profiles based on shared context service ontologies. The service characteristic strings are then generated for each context service and submitted to search engines as keywords.

After retrieving the index Web page of the desired context service, the agent can locate the service grounding file, and then the WSDL file. The retrieval of the service WSDL file represents the accomplishment of context service discovery.

By utilizing the Web, the proposed infrastructure automatically achieves the same scalability, fault tolerance and support for heterogeneous and dynamic context sources that the Web achieves for information sources (i.e. Web pages). The problem of automatic service discovery is solved by using Semantic Web technology and well developed Internet search engines.

5. Current Progress

We have developed a proof of concept application to demonstrate the feasibility and usefulness of the proposed infrastructure [3]. However, in this application, we only implemented the service discovery part of the proposed infrastructure: we used Internet search engines to discover context services in real time and on demand. We omitted the ontology based service description and used a proprietary description instead for simplicity. Our experiment showed that currently the service query and context retrieval process can finish in a few seconds. In the next phase, we will integrate Semantic Web document processing toolkits into our application to support OWL-S based service description and discovery.

6. Conclusion

In this paper we have identified five key challenges for context service discovery in wide area context-aware computing environments. We have proposed a large scale service discovery infrastructure and outlined our approach for context service discovery and description in context-aware computing environments. Drawing on the existing WWW and emerging Semantic Web technologies (i.e. OWL-S), the proposed infrastructure addresses the challenges and enables the development of wide area context-aware systems. Its effectiveness has been partly demonstrated by our proof of concept application.

Currently we are working on the development of context-aware systems which fully utilize the power of the proposed infrastructure. We are also aware of the security issues of context service discovery and we are working on a trust model for the large scale service discovery infrastructure.

[1] We do not discuss the details of task decomposition here since it is beyond the scope of this paper.

7. References

[1] Roy Want, Andy Hopper, Veronica Falcao, and Jonathon Gibbons, "The Active Badge Location System. ACM Transactions on Information Systems", Vol. 10, No. 1, January 1992, pp 91-102.

[2] Norman Sadeh, Ting-Chak Chan, Linh Van, OhByung Kwon and Kazuaki Takizawa, "Creating an Open Agent Environment for Context-aware M-Commerce", In "Agentcities: Challenges in Open Agent Environments", LNAI, Springer Verlag, pp.152-158, 2003

[3] Yong Liu and Kay Connelly, "SmartContacts: A Large Scale Social Context Service Discovery System", In the 3rd Workshop on Middleware Support for Pervasive Computing (PerWare 2006), Pisa, Italy, March 13, 2006.

[4] Jason Hong and James Landay, "An architecture for privacy-sensitive ubiquitous computing", In Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004), Boston, MA, USA (2004)

[5] Natalya Noy and Deborah McGuinness, "Ontology development 101: A guide to creating your first ontology", Technical Report KSL-01-05, Stanford Knowledge Systems Laboratory, 2001.

[6] Deborah McGuinness and Frank van Harmelen, "OWL Web Ontology Language Overview", W3C, 2003. http://www.w3.org/TR/owl-features/.

[7] O. W. L. S. Coalition, "OWL-S: semantic markup for Web services", 2004.

[8] WSDL2OWL-S, http://www.daml.ri.cmu.edu/wsdl2owls/

[9] The OWL-S Editor, http://owlseditor.semwebcentral.org/

[10] Tim Finin, James Mayfield, Clay Fink, Anupam Joshi, and R. Scott Cost, "Information Retrieval and the Semantic Web", In Proceedings of the 38th International Conference on System Sciences, January 2005.

[11] Web Services Description Language (WSDL), http://www.w3.org/TR/wsdl, 2001.