A Mechanism for Semantic Web Services Discovery in Mobile Environments

Rafael Besen, Frank Siqueira
Department of Informatics and Statistics (INE)
Federal University of Santa Catarina (UFSC)
Florianópolis, Brazil
{rbesen, frank}@inf.ufsc.br

***Abstract*—This paper describes SeMoSD (Semantic Mobile Service Discovery), a mechanism for discovery of services in mobile environments based on web services technology. The service discovery mechanism was designed taking into account the resource constraints of mobile devices. Through SeMoSD, mobile devices are able to profit from the use of semantic technologies for service discovery. Services are described using ontologies, and may be executed on either mobile or fixed nodes. Queries for services are interpreted by a semantic reasoner, resulting in more accurate search results.**

Keywords- Semantic Web Services, Ontology, Mobile Devices, DPWS, WSMO, Ubiquitous Computing.

I. INTRODUCTION

Computing resources provided by mobile devices may be shared with other nodes and employed to perform computing tasks cooperatively. In order to allow interoperation in a distributed environment, mobile devices must be able to locate each other on the network. Therefore, mechanisms for discovery of networked resources play an important role in the construction of distributed application in mobile environments.

The first step in the discovery process is the description of the shared resource. One widely adopted approach for sharing networked resources is the Service-Oriented Architecture (SOA) [1], which models services provided by computing devices that can have their execution requested remotely.

Services are usually described syntactically, through the specification of service names (i.e. a method or procedure) and messages (i.e. parameters). A purely syntactic description is a very simple and limited way to do this, because the same identifiers employed to describe the service upon registration must be specified when the service is queried.

On the other hand, a semantic description of services allows a richer amount of information regarding the provided services to be specified. Such technology is starting to be employed for the description and location of networked services, but is still novelty in the context of mobile computing devices.

This work describes SeMoSD (Semantic Mobile Service Discovery), a service discovery mechanism that aims to bring the benefits of semantic technologies to mobile computing environments. This is achieved through the definition of a model for interaction among devices, which is based on the DPWS (Devices Profile for Web Services) standard [2]. Semantic description and query processing rely on WSMO (Web Services Modeling Ontology) [3] and its execution engine WSMX (Web Services Modeling eXecution environment) [4].

The remainder of this paper is organized as follows. Section II analyzes the use of web services technology in the context of an environment composed by mobile devices. Semantic technologies adopted in this work are described in Section III. Section IV presents the architectural characteristics and execution dynamics of the proposed semantic discovery mechanism. An application scenario which employs the proposed discovery mechanism is described in Section V. Section VI presents some related research projects and compares them with the solution proposed in this paper. Finally, Section VII sums up the contributions given by this work and suggests further developments in this field.

II. MOBILE DEVICES AND WEB SERVICES

Mobile computing devices, such as smartphones, media players, tablets, wireless sensors and RFID tags, become more and more common every day. However, allowing these devices to interoperate seamlessly and to execute management tasks autonomously is still a distant reality. Devices from different vendors often employ different communication technologies, making them unlikely to be able to identify each other and exchange data in order to work together. In addition, in the context of mobile devices, care must be taken with some issues, such as frequent network disconnection, reduced processing power and storage capacity, and limited battery life.

Since these devices may join and leave the network at any time, requiring system reconfiguration, devices depend even more on the discovery mechanisms for locating new partners for the execution of cooperative tasks and for restoring dependencies on remote services.

The currently adopted mechanisms for device discovery in mobile environments are mostly based on syntactic information. Bluetooth SDP [5], Jini [6], UPnP [7] and Salutation [8] have discovery mechanisms with this limitation.

Attempts have been made to employ the Web Services technology [9] in the context of mobile devices. This technology has been widely adopted for building distributed

Copyright (c) IARIA, 2011 ISBN:978-1-61208-113-7

applications and for integrating legacy software. One of the main reasons for the success of this technology is the adoption of a set of broadly available standards – such as the eXtensible Markup Language (XML) [10], and the SOAP [11] and HTTP [12] communication protocols. This fact allows developers to create web services using several different programming languages and operating systems, in a truly heterogeneous environment.

However, the lack of appropriate mechanisms for description and discovery of services is still a limiting factor for the use of this technology. The Web Services Description Language (WSDL) [13] and the Universal Description, Discovery and Integration (UDDI) [14] provide means only for syntactic description and discovery of services, while semantic information on services is required for locating services without human intervention. Furthermore, these standards do not provide the required flexibility for describing and discovering services in a constantly changing environment, such as an ad-hoc network composed by mobile devices.

A. DPWS

The limitations for hosting web services in mobile devices are partially solved with the use of a standard called DPWS (The Devices Profile for Web Services) [2]. DPWS is a standard published by OASIS, which defines a communication model based on the Web Services technology, tailored for use in mobile computing devices. The adopted strategy allows devices connected to a network to syntactically describe the services provided by them and advertise these services to other devices. Devices can then have their services discovered dynamically and invoked by other devices which are able to understand the syntactic description of the service. Any device with connectivity and processing resources – such as mobile computers, smartphones, tablets, sensors and many others – can implement this standard and access or provide services.

DPWS identifies two kinds of services: hosting services, which represent devices, provide data about themselves to allow device discovery; and hosted services, which are computational services hosted by hosting services. They have their own network addresses in order to be reachable by other devices.

The flow of DPWS messages between a client device and a hosting service (device) with one hosted service is shown by Fig. 1. A client device broadcasts a *Probe* request, expressing its will to find other devices with characteristics specified in the message body. A device that matches the request replies with a *Probe Match* message. After getting in touch with one or more devices, the client may send the *Get Metadata* message to receive more information related to the device, including the list of available services. With this knowledge in hand, the client can send *Get Metadata* messages directly to the hosted service to get the required information for invoking the service. After receiving this information, the device can finally invoke the service.

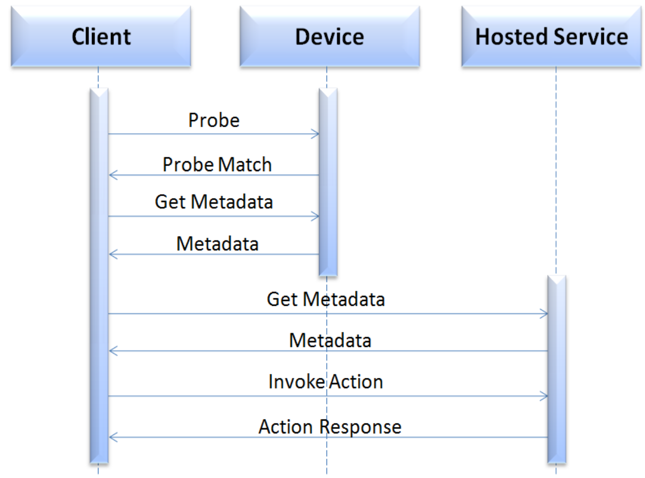

Figure 1. DPWS Messages

III. SEMANTIC TECHNOLOGIES

Better accuracy in service location may be obtained through the use of semantic technologies, which are based on the use of ontologies. Services described using ontologies are the base of a technology called Semantic Web Services [15].

An ontology can describe a set of interrelated concepts which belong to a certain domain, allowing these concepts to be associated to service operations and data exchanged with services. The use of ontologies allows searches to produce more accurate results, since it allows the exact meaning of data and the search context to be precisely specified. Furthermore, the described concepts are machine-readable, reducing the necessity of human intervention in the service discovery process.

Due to the resource constraints presented by mobile devices, the execution of semantic searches and the storage of semantic data may need resources that most devices are unable to provide. On the other hand, the provision of semantic web services can make feasible the creation of a ubiquitous computing environment [16], using the description of services to integrate them, and making a better use of resources provided by mobile devices.

The following sections present, respectively, the semantic language and the execution engine adopted in this work for semantic description of services hosted by mobile devices. These were chosen due to their superior characteristics and expressiveness over other semantic technologies, such as OWL-S [17].

A. WSMO

The Web Service Modeling Ontology [3] is a promising technology for developing Semantic Web Services. The interactions and data exchanged with a service are described using ontologies written in WSML (Web Services Modeling Language) [18]. The low-level representation of this language is based on XML, and shared resources are identified by URIs (Universal Resource Identifiers).

Copyright (c) IARIA, 2011 ISBN:978-1-61208-113-7

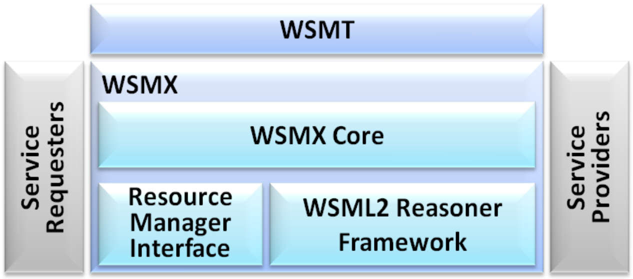

Figure 2. WSMX Architecture

WSMO allows roles and concepts to be expressed separately. Client requirements and available services are specified in different documents. Service description and implementation are also defined separately. This strategy improves reuse and maintains backward compatibility with systems that lack semantic capabilities.

The description and discovery of services is based on ontologies, which must define the meaning of data exchanged through messages. Services are described using concepts, relations and instances of this ontology. To identify a desired service, a goal is specified, containing all features needed in a service. Semantic reasoners compare goals with the description of the available services in order to find one or more matching services.

B. *WSMX*

The Web Services Modeling eXecution environment [4] is an execution engine for WSMO. WSMX allows services to be integrated in an automatic, flexible and easy way. The WSMX engine allows developers to integrate legacy systems without requiring internal changes in their source code.

The WSMX is composed by a set of components, each one with a service interface and a specific role. These components, which are illustrated by Fig. 2, include a reasoner framework [19][17], which allows the user to choose or add semantic matching algorithms; and WSMT [20], which is a tool that helps users to manage semantic descriptions and to interact with the execution engine.

IV. SEMOSD – SEMANTIC MOBILE SERVICE DISCOVERY

The Semantic Mobile Service Discovery infrastructure is aimed at providing a repository for storing and locating semantically-rich service descriptions in an environment composed by mobile devices. Such a repository is necessary because a typical mobile device does not have enough computing resources for storing semantic service descriptions in a local cache and also for executing the semantic reasoning algorithms.

The repository implementation and all the interaction with it are based on widely available standards and technologies, which had to be adapted and integrated to allow the execution of semantic services on mobile devices. In the center of this infrastructure is an implementation of the DPWS standard, which provides the communication mechanisms for interaction among mobile devices.

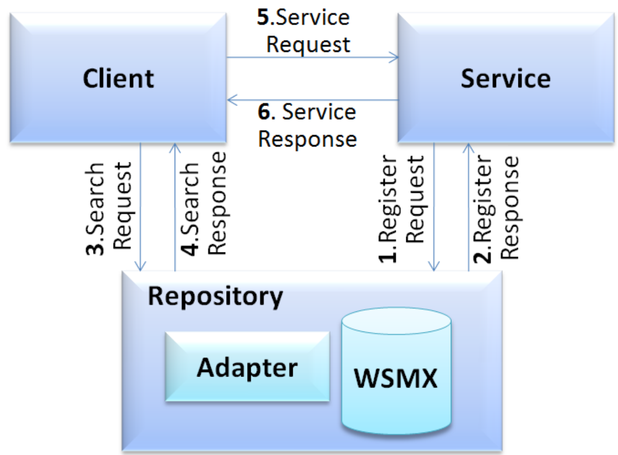

Figure 3. Execution Model

Metadata obtained through DPWS holds service descriptions. Since the standard does not define how services are described, it is possible provide a semantic description for a service as metadata. To maintain backward compatibility with already existing DPWS applications, which are unable to understand a semantic description, a syntactic description is maintained in this attribute. This is done by extending the metadata class to carry also semantic data, keeping the syntactic data untouched. Therefore, clients that are unable to interpret the semantic description may participate in a regular DPWS client-server interaction.

Ontologies must be employed to semantically describe services, defining device concepts, services, relationships and instances. In this work we have adopted WSMO, which was described in Section **Erro! Fonte de referência não encontrada.**. Services described semantically (i.e., using concepts and relationships defined in an ontology) may be more easily and precisely located by clients. The characteristics of the desired service may be described by the client building a goal using WSML syntax. Alternatively, a client may just specify parameter values to specialize a goal stored in the semantic repository. After locating one or more service descriptions in the repository, the client can choose one or more services to interact with, issuing regular service invocations.

The semantic repository employs an adapter, which receives DPWS requests from client devices, parses these requests and sends them to WSMX. The adapter must implement the DPWS protocol stack and run on the same platform as the WSMX engine. Reasoners provided by WSMX are responsible for locating services that match the characteristics described in the goal executed by the client device. The adapter waits for a response from the reasoner and, as soon a list of matching services becomes available, a response is given to the client device.

Since the reasoning algorithms require more processing power than is currently available in most mobile devices, and taking into account the importance of this service in the environment, the semantic repository might have to be hosted by a device with more processing power and storage capacity and with non-intermittent network connectivity.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-113-7

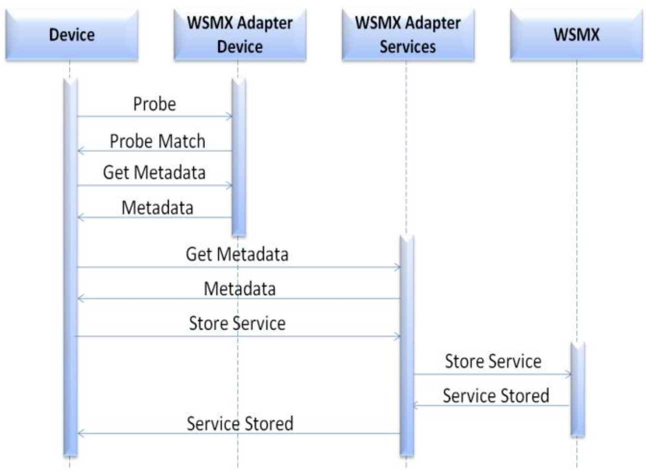

Figure 4. Service Behaviour

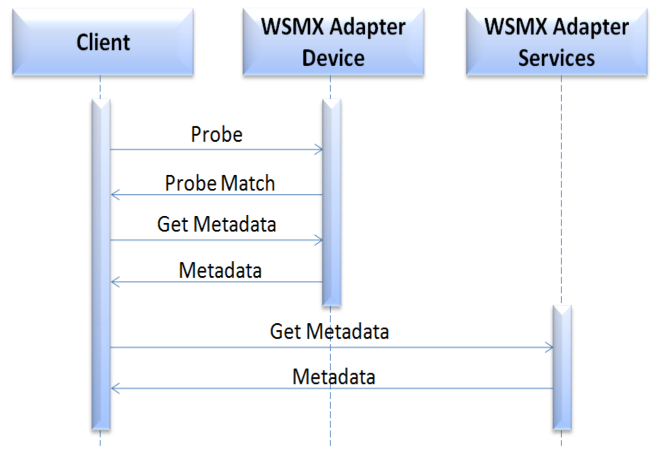

Figure 5. Client Behaviour

A. *Execution Model*

Fig. 3 explains the interaction among devices during the publication of a service description and the execution of a search for the described service. Upon startup, a device – either service or client – must broadcast a DPWS probe in an attempt to locate a semantic discovery mechanism within its reach. Thus, the first request is a regular DPWS request, which does not require any semantic knowledge. To allow the semantic discovery mechanism to be located through a regular DPWS request, it must always be described in DPWS Metadata as "WSMX Service".

If there is no semantic discovery mechanism available, the device tries to locate services using the discovery mechanism provided by the DPWS protocol. When a semantic discovery mechanism connects to the network, all services are notified through an announcement message and, from this moment on, they may start using the mechanism – i.e. they may register their services and ask for the location of services described semantically.

Fig. 4 explains in more detail the request flow between a service provider and the SeMoSD infrastructure. When a provider connects to the network, it broadcasts a DPWS probe message, willing to find a semantic discovery mechanism. After identifying a repository (*Probe Match* message), the provider can obtain metadata from the device and from the WSMX service (*Get Metadata* and *Metadata* message), and store the description of its services (*Store Service* message, confirmed with a *Service Stored* message).

The same startup procedure is executed by clients to locate the WSMX Service. The exchanged messages are illustrated by Fig. 5. After this initial step, clients may specify goals that will be executed by the WSMX Service, in an attempt to locate a required service. New goals can be executed, and preexisting goals can be parameterized. When a request is received, the WSMX Adapter parses its contents and sends the request to WSMX. Then, the semantic reasoning is executed and the response is sent to the adapter, which forwards the response to the client. With the search results in hand, the client can invoke these services directly using the DPWS protocol.

B. *Prototype Implementation*

A prototype of the SeMoSD infrastructure was developed in Java, using JDK version 6. The Java Platform was chosen due to the extensive support for mobile devices and the easy integration with WSMX. Version 0.5 of WSMX was adopted in this prototype, due to the fact that the 1.0 version is still in beta release and is not stable. The adapter, which allows the integration between DPWS and WSMX, translates requests into WSML using the WSMX Integration API.

Clients and services for testing purposes were developed using the JMEDS implementation of DPWS, but any other DPWS implementation can be adopted for this purpose. WSMO4J was employed for describing services and goals in WSML.

Performance measurements obtained during tests with this prototype have shown that searches executed through the adapter, which was co-located with the WSMX service, took approximately 139 ms longer than a direct request to the WSMX service. The total search time in a WSMX service with a minimal set of registered services takes 3.59 seconds – i.e., the overhead imposed by the proposed mechanisms is lower than 4%.

V. CASE STUDY

Fig. 6 shows a usage scenario chosen to illustrate the use of the SeMoSD infrastructure. This scenario consists in the computing support for a disaster relief operation in a site where, due to natural phenomena such as an earthquake, a mudslide, intense flooding or a hurricane, several people, either injured or with their lives at risk, must be rescued as soon as possible.

When multiple rescue teams arrive at the location, they must be aware of the resources available at the site and in the neighborhood. Besides, it is necessary to provide information about the rescue to families and to the press.

The semantic infrastructure can be useful to allow the cooperation among rescue teams and to coordinate the use of human and physical resources, such as medical staff, firemen, truck drivers, and also ambulances, bulldozers, fire trucks, helicopters, and so on.

Figure 6. Rescue Operation Scenario

Suppose that each city or state has a repository that gathers information about all the local resources available for a disaster relief operation, which are described using the ontology depicted in Fig. 7. Information systems and mapping services are also necessary for executing the disaster relief operation. With such information available, teams can work cooperatively, coordinating the use of local resources and of services provided by all teams that take part in the operation.

Upon arrival at the site, rescue personnel must connect their mobile devices to the local semantic repository, and identify the services available. Each team must publicize a semantic service providing information about the services it provides, such as transportation, healthcare and so forth. Available equipment with network connectivity may also provide real-time information about their location and availability.

For example, a team rescuing people from a collapsed building may request assistance from first-aid teams to provide preliminary medical care. These can locate ambulances to take the injured to the closest hospital with available beds, equipment and personnel to give the required treatment. To do so, first-aid teams and ambulances must provide their GPS coordinates and current availability. Hospitals must also provide real-time information regarding the available medical services. Based on this information, cooperation strategies may be more precisely and efficiently architected and executed.

VI. Related Work

Most of the existing discovery mechanisms for mobile environments do not provide support for semantic discovery of devices and services. Some technologies that fall in this category are already in use, such as Bluetooth SDP, Jini and UPnP. Others, such as TOTA Approach [21] and the Device Service Bus [22], have been proposed recently and are still experimental work. In dynamic environments with multiple heterogeneous devices, it is very unlikely that devices will be able to interpret syntactic data without human intervention.

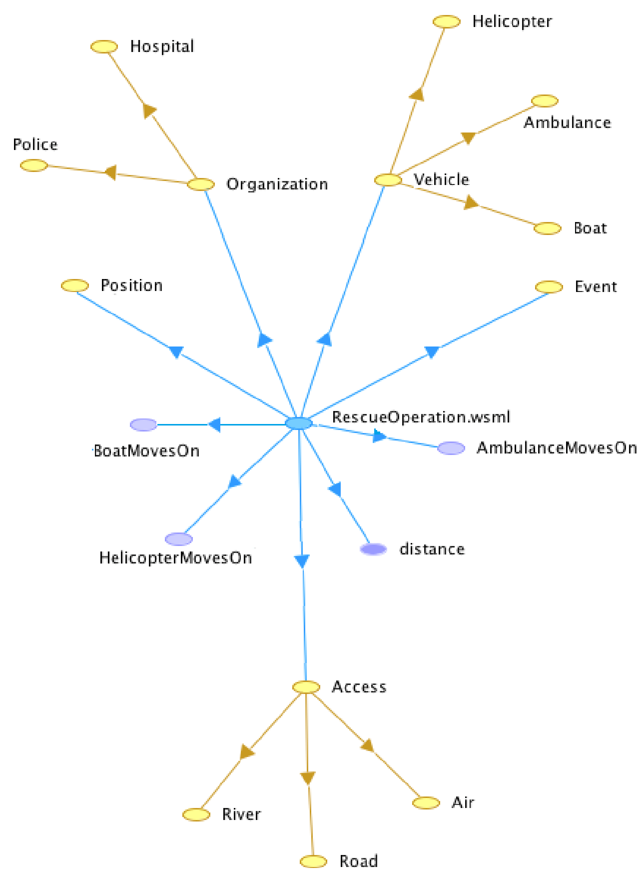

Figure 7. Ontology for Rescue Operations

Some applications employ semantic data to describe services, such as AIDAS [23], which is developed in Java and uses information from mobile devices to catch user context information. Based on this information, semantic reasoning is performed to find the available services. However, the mobile devices are employed only to retrieve information, and the proposal is limited to the Java language.

Another technology that employs semantic knowledge to locate services is Home-SOA [24]. This project uses OSGi components, because the authors believe that web services are not the best choice to do the communication, due to the high traffic created. Home-SOA defines communication interfaces that are limited to languages supported by the Java Virtual Machine, and its use is also limited to small domestic networks, composed mainly by multimedia devices.

The GEA architecture [25] is targeted at government services. To describe services, GEA uses ontologies written in OWL-S [17]. Services in this architecture are mainly to process documents and to access document models and sheets. The use of mobile devices is not addressed by this architecture.

Table 1 presents a comparison between these projects and the semantic discovery service proposed in this paper. Other projects in this area also have limitations regarding the supported programming languages, the lack of adherence to open standards, the use of restrictive environments, or the lack of support for mobile devices.

Copyright (c) IARIA, 2011 ISBN:978-1-61208-113-7

TABLE I. CHARACTERISTICS OF SEMANTIC MODELS

Feature	Semantic Models			
	AIDAS	***Home-SOA***	***GEA***	***SeMoSD***
Communi -cation	Web Services	OSGi	Web Services	Web Services (DPWS)
Language Constraints	Java	JVM	None	None
Scope	Any	Home Media	Electronic documents	Any
Mobility Support	Limited	Yes	None	Yes

VII. FINAL REMARKS

This paper described SeMoSD, a semantic discovery infrastructure aimed at reaching a balance between providing accuracy of service location and adequacy for mobile environments. Its adoption simplifies the construction of flexible ubiquitous computing environments.

The proposed infrastructure is based on standards which, combined, allow the semantic location of services hosted by mobile devices. The amount of network messages necessary to locate a service is reduced comparing to other location mechanisms, avoiding the excessive use of battery, which limits the lifetime of devices.

A prototype of the proposed mechanism was implemented to verify its feasibility, and tests with this prototype have shown that it introduces a small overhead (lower than 4%) in the total search time. As future work, we intend to test the prototype in a real-life application in order to validate the proposed mechanism.

REFERENCES

[1] T. Erl, "Service-Oriented Architecture: Concepts, Technology and Design", Prentice Hall, 2005.
[2] OASIS, "OASIS Devices Profile for Web Services (DPWS) Version 1.1", July 2009, http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01, Accessed: September 25, 2010.
[3] ESSI WSMO working group, "D2v1.3. Web Service Modeling Ontology (WSMO)", October 2006, http://www.wsmo.org/TR/d2/v1.3/, Accessed: September 25, 2010.
[4] T. Hasselwanter, P. Kotinurmi, M. Moran, T. Vitvar, and M. Zaremba, "WSMX: a Semantic Service Oriented Middleware for B2B Integration", Proc. 4th Intl. Conf. on Service Oriented Computing, pp. 477-483, Chicago, USA, December 2006, DOI: 10.1007/11948148_43.
[5] BlueTooth SIG, "BlueTooth Specification v. 3.0", April 2009, http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications, Accessed: September 25, 2010.
[6] Sun Microsystems, "Jini Technology Core Platform Specification v2.0", June 2003.
[7] UPnP Forum, "UPnP Device Architecture 1.0", December 2003.
[8] Salutation Consortium, "Salutation Architecture Specification", 1999.
[9] W3C, "Web Services Architecture", February 2004, http://www.w3.org/TR/ws-arch/, Accessed: September 25, 2010.
[10] W3C, "Extensible Markup Language (XML)", September 2003, http://www.w3.org/XML/, Accessed: September 25, 2010.
[11] W3C, "SOAP Version 1.2 Part 1: Messaging Framework", April 2007, http://www.w3.org/TR/soap12-part1/, Accessed: September 25, 2010.
[12] R. Fielding et al., "Hypertext Transfer Protocol – HTTP/1.1", IETF RFC 2616, June 1999, http://www.w3.org/Protocols/rfc2616/rfc2616.html, Accessed: September 25, 2010.
[13] W3C, "Web Services Description Language (WSDL) 1.1", March 2001, http://www.w3.org/TR/wsdl, Accessed: September 25, 2010.
[14] OASIS, "UDDI Version 3.0.2", October 2004, http://www.uddi.org/pubs/uddi_v3.htm, Accessed: September 25, 2010.
[15] J. Cardoso, "Semantic Web Services: Theory, Tools and Applications", Information Science Reference, 2006.
[16] M. Weiser, "Hot Topics: Ubiquitous Computing", IEEE Computer, 26(10), pp. 71– 72, October 1993, DOI: 10.1109/2.237456.
[17] W3C, "OWL-S: Semantic Markup for Web Services", November 2004, http://www.w3.org/Submission/OWL-S/, Accessed: September 25, 2010.
[18] ESSI WSMO working group, "The Web Service Modeling Language WSML", August 2008, http://www.wsmo.org/wsml/wsml-syntax, Accessed: September 25, 2010.
[19] S. Grimm, U. Keller, H. Lausen, and G. Nagypal, "A Reasoning Framework for Rule-Based WSML", Proc. 4th European Semantic Web Conference (ESWC), Innsbruck, Austria, June 2007, pp. 114-128, DOI: 10.1007/978-3-540-72667-8_10.
[20] M. Kerrigan, "The Web Services Modelling Toolkit (WSMT)", Technical report, 2005, http://www.wsmo.org/TR/d9/d9.1/v0.2/, Accessed: September 25, 2010.
[21] M. Mamei and F. Zambonelli, "Programming pervasive and mobile computing applications: The TOTA approach", ACM Trans. Softw. Eng. Methodol., 18(4), July 2009, DOI: 10.1145/1538942.1538945.
[22] G. M. Araújo and F. Siqueira, "The Device Service Bus: A Solution for Embedded Device Integration through Web Services", 24th ACM Symposium on Applied Computing, Honolulu, Hawaii, USA, March 2009, pp. 185-189, DOI: 10.1145/1529282.1529322.
[23] A. Toninelli, A. Corradi, and R. Montanari, "Semantic-based discovery to support mobile context-aware service access", Computer Comunication, 31: pp. 935-949, January 2008, DOI:10.1016/j.comcom.2007.12.026.
[24] A. Bottaro and A. Gérodolle, "Home SOA – Facing Protocol Heterogeneity in Pervasive Application", Proc. Intl. Conf. Pervasive Services, Sorrento, Italy, July 2008, pp. 73-80, DOI: 10.1145/1387269.1387284.
[25] V. Peristeras and K. Tarabanis, "Advancing the Government Enterprise Architecture – GEA: The Service Execution Object Model", R. Traunmüller (Ed.), Electronic Government 2004, pp. 476–482, 2004, DOI: 10.1007/978-3-540-30078-6_83.