Adaptive Semantic Support Provisioning in Mobile Internet Environments

Antonio Corradi, Rebecca Montanari, Alessandra Toninelli
Dipartimento di Elettronica, Informatica e Sistemistica
Università di Bologna
Viale Risorgimento, 2 - 40136 Bologna - Italy
{acorradi, rmontanari, atoninelli}@deis.unibo.it

Abstract

The Mobile Internet scenario encourages the design and development of context-aware applications that provide results depending on context information, such as the relative position of users, user preferences, device capabilities and available resources. A key requirement for the provisioning of context-aware applications is to give computer systems the ability to understand context information. Semantic languages are well suited to leverage the possibility to express, process and reason about context information and to facilitate knowledge sharing and interoperability among previously unknown entities accessing services from heterogeneous devices. However, the exploitation of semantic languages for the design and deployment of context-aware applications raises new challenges, mainly due to the high degree of heterogeneity that mobile devices exhibit in terms of computing power, memory, operating system, and supported software. Semantic languages require complex and heavyweight support facilities, e.g. metadata interpreters, reasoning engines and ontology repositories, that may not fit the capabilities of all user access devices, especially of the resource-limited ones. Novel solutions are required that are capable of transparently and dynamically adapting semantic support functionalities to the properties of the different user access devices. The paper proposes a novel middleware solution that exploits the visibility of two kinds of metadata, user/device profiles and policies, to tailor semantic support functionalities and that offers a wide set of mechanisms for providing on demand appropriate semantic support to mobile portable devices.

1. Introduction

The widespread availability of wireless network connectivity together with the increasing adoption of portable devices has brought to the wide diffusion of pervasive computing environments and has encouraged the design and development of context-aware applications that provide results depending on context information. Context is a complex notion that has many definitions. Here we adopt the definition of context as any information that can be used to characterize the situation of a person or of a computing entity, such as location information, system capabilities, services offered and entities' role and desires [1]. New conceptual models and implementing technologies are needed to enable not only context representation, but also its comprehension by computing entities. Semantic languages have recently gained considerable attention as a suitable means to express metadata, such as physical context information [1], [5] or mobile devices' capabilities and users' preferences [6]. In fact, they offer many advantages. Firstly, ontologies allow knowledge sharing between independently developed context-aware systems. Secondly, Semantic Web languages permit explicit context representation at a high level of abstraction while enabling automated reasoning about this representation. Finally, these languages can be used as meta-languages to create special purpose languages, e.g. policy languages [1].

However, the exploitation of semantic languages by heterogeneous and possibly resource-limited devices poses new challenges. In order to exploit semantic languages, in fact, several support services are needed, ranging from appropriate repositories to store and retrieve ontologies, to inference engines able to reason about them, to metadata interpreters. Semantic support services may be rather complex and typically require a large amount of computational/memory resources that may not fit devices with strict limited resources. Let us consider, for instance, a resource-limited device such as a PDA which requests to discover a local service with specific semantic characteristics. Given its scarce computational capabilities, the PDA cannot host an inference engine on-board, but may still expect to be provided with an external semantic support, namely a reasoner and a complete ontology base, that enables it to complete its discovery. It may also happen the case of a device, e.g. a laptop, that hosts on-board both a set of ontologies and an autonomous reasoner, but not the whole set of ontologies due to memory limitations. In this case, the laptop needs to be supplied with domain-specific ontologies to properly reason during service discovery.

We claim that the heterogeneity of mobile devices in terms of computational power and memory capacity requires novel middleware solutions capable of adapting the semantic service support to the different device capabilities. In particular, this novel middleware should be able to configure semantic support functionalities according to devices properties and user needs. Some initial research ideas are starting to emerge that aim to build the kind of architecture suited to mobile environments [1], [3], [4]. None of them enables to adapt the configuration of semantic support to the various capabilities of mobile devices.

To address these issues, we propose a framework for configurable semantic support to mobile users, called MASS (**M**iddleware for **A**daptive **S**emantic **S**upport). MASS focuses on two peculiar aspects. Firstly, it exploits the visibility of two kinds of metadata, user/device profiles and policies, to tailor semantic support functionalities. This configuration feature enables the framework to adapt semantic functionalities to several kinds of users and devices, thus dealing with the heterogeneity typical of pervasive environments. Secondly, it allows each mobile device to exhibit its semantic functionalities, so that they can be accessed by users in the vicinity, and it enables the device to discover and to exploit semantic support capabilities offered by the nearby devices. The paper is organized as follows. Section 2 describes the configuration model adopted in our framework. Section 3 presents an overview of the middleware architecture. Concluding remarks and future research directions are given in Section 4.

2. MASS Configuration Model

MASS adopts metadata to represent context information and interacting parties. Metadata can be exploited to describe at a high level of abstraction the structure properties of the entities composing a system and to specify the desired management operations to govern a system. MASS distinguishes two kinds of metadata: profiles and policies.

Profiles describe the characteristics, the requirements and the capabilities of a system component, e.g., a user, a device or a service. MASS profiles are composed of two different parts: *capabilities* and *preferences*. Capabilities include all the information needed to qualify an entity in terms of what it is capable of. For instance, a device may have the capability of performing a reasoning process or can have sufficient memory to host a certain quantity of knowledge base. Preferences are used to express the desired configuration settings choices. For example, a user may request to be informed about the knowledge base available in the system.

Policies are high-level directives regulating the system's behaviour, in terms of which actions a subject can or must perform. MASS adopts policies to drive the configuration process that tailors semantic support to fit specific user/device properties. In particular, MASS distinguishes two kinds of policies. Authorization policies define what a subject can or cannot do on specific target resources if certain context conditions are met. For example, the transfer of a certain amount of ontologies on a mobile device is authorized if the device's storage space has a minimum, predetermined capacity. Obligation policies specify actions which have to be carried out at certain event occurrence, given that specific conditions are verified. For instance, if a mobile user has stated that she is interested in any change occurring in the knowledge base stored at the fixed node, she must be informed when new ontologies are added to this knowledge base.

MASS exploits Resource Description Framework (RDF) and Web Ontology Language (OWL) to express metadata. The main advantage offered by semantic languages, such as RDF and OWL, is that they are adequate both for machine processing and for automated reasoning. OWL allows to define application-specific ontologies, which can subsequently be exploited to describe a component and its behaviour. The component's description, also written in OWL, is then included in the component's profile. Through metadata parsing and automated reasoning, the system can acquire useful information about the entities joining it, even if they were unknown before interaction.

3. MASS Middleware

MASS is built on top of the Java-based CARMEN system that supports the design, development and deployment of context-dependent services for the wireless Internet, and interacts with third-party semantic support frameworks. CARMEN is centered on the distributed deployment of active middleware proxies over the fixed network to support service provisioning to portable devices [2]. CARMEN provides any portable device with a companion middleware proxy (*shadow* proxy) that autonomously acts on its behalf, possibly negotiates service tailoring to fit user/device characteristics and follow s user/device movements among network localities. CARMEN implements shadow proxies by exploiting the Mobile Agent programming paradigm. In particular, CARMEN provides proxies with execution environments, called places, that typically model nodes. Places can be grouped into domains that correspond to network localities, e.g., either Ethernet-based LANs or IEEE 802.11b-based wireless LANs. With a finer degree of detail, a shadow proxy is implemented by one CARMEN agent running on a place in the domain where the portable device is currently located. In fact, the domain abstraction allows to define a well-specified management boundary: each domain holds references to the entities currently members of the domain (both MAs and services) and to the metadata applicable to these entities.

As a new user searching a semantic-based service enters a CARMEN domain, her proxy forwards the searching request to the MASS middleware which initiates a configuration phase. Profiles and policies, along with the reasoning abilities needed to interpret them, are used by the MASS middleware during the configuration phase. In particular, when a user first enters the domain, she presents her profile and her device's profile in order to supply the framework with an explicit representation of her characteristics. The information acquired through profile parsing and reasoning is exploited, together with the rules encoded in configuration policies, to take appropriate decisions about the user's configuration settings. For instance, if the user's device exhibits a storage space of dimension 2x (profile's capability) and the transfer of ontologies on a mobile device is allowed if the device hosts a storage space of at least x capacity (authorization policy), then MASS will consent to perform the ontology upload, if required. Note that the upload requirement could be stated as a preference within the profile. After the configuration phase is terminated, the proxy receives the customized and appropriate semantic support.

MASS middleware facilities are shown in Figure 1. The **Configuration Manager** tailors the semantic support functionalities available within the domain on the basis of user/device profile. The **Ontology Manager** is responsible for maintaining, updating and retrieving needed ontologies. The **Reasoning Manager** enables communication between the reasoning elements available in the system and other interacting parties. Finally, the **Discovery Manager** allows the discovery and advertising of nearby semantic-enabled devices, and retrieves application-level services within the domain.

Configuration Manager (CM). This component tailors semantic support for each user/device joining the domain on the basis of its profile. At the incoming of a new user within the CARMEN domain, the Configuration Manager retrieves and parses the user/device profile in order to acquire all useful information to choose appropriate configuration settings. There are several possible configuration settings, such as the case of a device which can carry only a small amount of knowledge and requires both the remaining ontologies and a proper reasoner, or the case of a computing entity which is able to perform autonomous reasoning but needs the whole set of ontologies to deduce inferences:

- **Ontology-On-Demand.** This is the case of a device which hosts only its own ontologies, i.e. ontologies describing metadata used in the device's profile. In this configuration setting, external ontologies are requested on-demand.
- **Embedded Ontology.** This is the case of a device with rich storage space and memory management features. In this configuration setting, the user prefers to acquire and locally store all the ontologies she needs, either to perform reasoning or to enhance its local knowledge base.
- **Remote Reasoning.** In this configuration setting, the reasoning process does not occur on-board, but it is delegated to an external component. MASS middleware, and specifically the RM service, intercepts the request for remote reasoning, forwards it to the appropriate reasoner and gives back the answer to the requesting device. Note that this option may also be useful in the case a device prefers to delegate reasoning, e.g. for saving battery, notwithstanding its reasoning capabilities.
- **Embedded Reasoning.** This is the case of a device that can rely on its embedded reasoning capabilities.

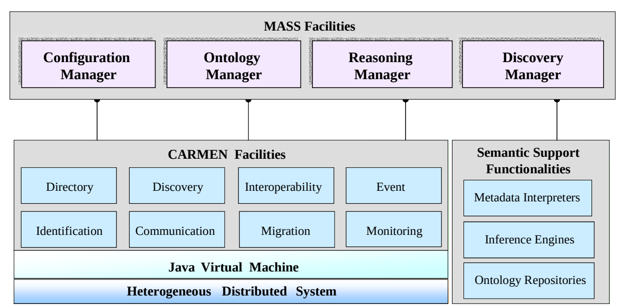

Figure 1. MASS Middleware

Discovery Manger (DM). The Discovery Manager is in charge of performing semantic-driven discovery to retrieve services available in the domain. DM integrates CARMEN discovery facilities with semantic functionalities obtained through the interaction with other MASS managers, namely the Ontology Manager and the Reasoning Manager. In particular, when DM receives a request for semantic-enhanced service discovery, it first parses the request to individuate metadata-related ontologies. Then, it asks OM to retrieve these ontologies and RM to perform reasoning about them, in order to dynamically deduce useful information for the service discovery. If, for instance, a "printer" service is required, by means of reasoning the DM can recognize that an "ink-jet printer" service represents a subclass of the previous one and is therefore a good candidate to semantically fulfill the request. The interaction among DM, OM and RM may take place several times during the discovery activity.

Ontology Manager (OM). The Ontology Manager is responsible for maintaining and managing the whole ontological knowledge stored within the domain. OM coordinates the various knowledge sources provided by the domain entities, e.g. the framework's global knowledge base, if it exists, and the application-specific ontologies carried by mobile devices. In particular, in the case OWL ontologies have to be imported from one knowledge base to another, OM performs entailment and consistency checks in order to avoid conflicts and inconsistencies between old and new modules.

Reasoning Manager (RM). The Reasoning Manager forwards reasoning requests to the appropriate reasoning engine, and then returns the result to the requesting entity. Note that the choice of the most appropriate reasoner may be driven by different criteria, ranging from performance evaluations to inference logic. In order to properly manage incoming requests and their corresponding answers, RM performs syntactical and/or logical transformation needed to produce a reasoner-compliant query from the original request and to perform the corresponding backward conversion. For example, if a user could only generate queries using RDF Query Language (RDQL) while the reasoner could only accept Prolog clauses, RM would provide for the translation from RDQL to Prolog and vice versa. Finally, RM is in charge of managing a Reasoning Cache, that is a storage space dedicated to reasoning activity which contains the most frequently and/or recently used ontologies.

4. Conclusions and Ongoing Work

Semantic languages have recently gained attention as a means of expressing context-related metadata in pervasive computing applications. However, the exploitation of semantic support requires a considerable amount of memory and computational resources that may not fit resource-limited devices. We propose a novel middleware which is capable of adapting semantic support to the different characteristics of mobile devices and provides mobile users with the visibility on semantic functionalities hosted by nearby devices.

At present the development of MASS is still in the early stage of research with initial middleware service prototypes. More work is needed to evaluate and test MASS in real application scenarios.

References

[1] H. Chen, T. Finin, and A. Joshi, "Semantic Web in a Pervasive Context-Aware Architecture", *Artificial Intelligence in Mobile Systems 2003*, Universitat des Saarlandes, Saarbrucken, Germany, October 2003.

[2] P. Bellavista, A. Corradi, R. Montanari, and C. Stefanelli, "Context-Aware Middleware for Resource Management in the Wireless Internet", *IEEE Transactions on Software Engineering*, Vol. 29, No. 12, December 2003.

[3] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, and D. Riboni, "Integrated Profile and Policy Management for Mobile-Oriented Internet Services", Technical Report Firb-Web-Minds N. TR-WEBMINDS-04.

[4] R. Masuoka, B. Parsia, and Y. Labrou, "Task Computing – the Semantic Web meets Pervasive Computing – ", *Proc. of 2nd International Semantic Web Conference (ISWC2003)*, Sanibel Island, Florida, October 2003.

[5] X. Wang, J.S. Dong, C.Y. Chin, S.R. Hettiarachchi, and D. Zhang, "Semantic Space: An Infrastructure for Smart Spaces", *IEEE Pervasive Computing*, Vol. 3, No. 3, July-September 2004.

[6] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, H. Butler, and L. Tran (Eds.), "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", W3C Proposed Recommendation, October 2003.