Loosely Coupling Ontological Reasoning with an Efficient Middleware for Context-awareness *

Alessandra Agostini Claudio Bettini Daniele Riboni
DICo, University of Milan
via Comelico 39, I-20135 Milan, Italy
{agostini,bettini,riboni}@dico.unimi.it

Abstract

Context-awareness in mobile and ubiquitous computing requires the acquisition, representation and processing of information which goes beyond the device features, network status, and user location, to include semantically rich data, like user interests and user current activity. On the other hand, when services have to be provided on-the-fly to many mobile users, the efficiency of reasoning with these data becomes a relevant issue. Experimental evidence has lead us to consider currently impractical a tight integration of ontological reasoning with rule based reasoning at the time of request. This paper illustrates a hybrid approach where ontological reasoning is loosely coupled with the efficient rule-based reasoning of a middleware architecture for service adaptation. While rule-based reasoning is performed at the time of service request to evaluate adaptation policies and reconcile possibly conflicting context information, ontological reasoning is mostly performed asynchronously by local context providers to derive non-shallow context information. A limited form of ontological reasoning is activated at the time of request only when essential for service provisioning.

1 Introduction

A consensus seems to have been reached on the need for deep context-aware adaptation and personalization of services in the domain of mobile and ubiquitous computing. Many research efforts are undergoing to define new representation formalisms as well as to provide middleware architectures to support context-awareness in this domain (e.g., [4, 8, 16]). We believe that a comprehensive solution should take into account at least the following requirements: *i) Support for distributed context data*: context data are naturally provided by different sources in some cases delivering conflicting information. *ii) Interoperable context representation*: context data must be exchanged among various entities therefore it is highly advisable the use of a standard language, a shared vocabulary and a non-ambiguous semantics. *iii) Support for context dynamics*: it must be possible for multiple entities (e.g., users, providers, agents) to define how some changes on context reflect on other context data; moreover, changes in context must be asynchronously communicated to the interested entities. *iv) Efficiency*: the time needed for adaptation should not significantly affect the final user. *v) Reasoning with socio-cultural context information*: the integrated profile must describe data that cannot be modeled with simple attribute/value pairs (e.g., the user current activity), but need more sophisticated representation formalisms.

We have defined an architecture to support adaptation in mobile environments –presented in [1], [6], and [20]– that fulfils requirements *i)*, *ii)*, *iii)*, and *iv)*. The architecture is based on distributed context data represented by means of CC/PP profiles [18]. Adaptation and personalization parameters–which belong to CC/PP profiles too–are determined by policy rules defined by both the user and the service provider. In this paper we describe an extension of this framework that aims at satisfying requirement *v)* while preserving efficiency. Our proposal is supported by an implementation of the architecture, by experimental evaluations of reasoning execution times, and by implementing prototypes of some typical mobile services.

The necessity of extending our framework originates from the fact that (as others have independently observed in [17]) we found CC/PP to have serious limitations in representing non-shallow context data. The limitations are particularly evident when trying to represent context information regarding the socio-cultural environment of the user. A switch to ontology languages for the representation of context seems to be advocated by many researchers in the field (e.g., [9, 12]). However, this switch introduces the problem of the classical trade-off between expressiveness and

*This work has been partially supported by Italian MIUR (FIRB "WebMinds" project N. RBNE01WEJT_005).

efficiency. Indeed, it is well known that reasoning with the logics underlying ontologies is in most cases intractable [3].

Our solution is based on the integrated representation of shallow contex data (e.g., device capabilities and network parameters) and context data belonging to more complex domains (e.g., user activity and interests) by means of CC/PP profiles which contain a kind of reference to ontological classes and relations. In order to preserve efficiency, ontological reasoning with non-shallow data is mainly performed in advance with respect to the service provision. Ontological reasoning at the time of the service request is made on-demand only in particular cases; that is, when the integrated profile lacks some context information that is crucial for providing the service. Actually, analyzing pragmatically various case studies, we believe that there are only few cases in which non-shallow data cannot be calculated in advance. Moreover, in order to keep complexity acceptable for mobile computing services, our approach is to perform ontological and rule-based reasoning separately, and use the ontology query engines provided by well-known tools like Racer [13] as an intermediate layer.

The rest of the paper is structured as follows: the following section illustrates how we represent context data, and how they are bounded to ontologies. In Section 3 we recall the main aspects of our middleware architecture, while in Section 4 we explain how ontological reasoning is performed. Section 5 briefly reports on experimental results, Section 6 discusses related work, and Section 7 concludes the paper.

2 Context Modeling

The context information we intend to model includes, in principle, *any information that can be used to characterize the situation* [10] of a mobile user requesting a service. It includes spatio-temporal information (e.g., user's location, time), environmental conditions (e.g., lighting, noise level), data about the technological infrastructure (e.g., device features, network connections, available bandwidth), user preferences, as well as socio-cultural information (e.g., user current activity and interests). It is easily observed that this information is naturally distributed since it is handled or collected by different entities by different means and it is subject to different privacy policies. In our framework we use the term *profile* to indicate a subset of context information collected and managed by a certain entity. One of the goals of our architecture is to aggregate profiles to obtain the whole context data that can be used by a service provider for obtaining an effective adaptation and personalization of the service. We divide profile data in two classes: *shallow profile data* and *ontology-based profile data*.

```
<RDF xmlns:cityTrvl="http://www.dakwe.
   dico.unimi.it/owl/cityTrvl-20040902#">
 xmlns:uaprof="http://www.wapforum.org/
            UAPROF/ccppschema-19991014#">
<ccpp:component>
 <Description about="http://....">
  <type resource="http://...#BrowserUA"/>
  <uaprof:AvailableStorageMemory>
   37.45
  </uaprof:AvailableStorageMemory>
 </Description>
</ccpp:component>
<ccpp:component>
 <Description about="http://...">
  <type resource="http://...#Context"/>
   <cityTrvl:currentLocationType>
    outdoor
   </cityTrvl:currentLocationType>
 </Description>
</ccpp:component>
</RDF>
```

Figure 1. An excerpt of a profile

2.1 Shallow Profile Data

We denominate shallow profile data those attributes which can be modeled in a natural way by using attribute/value pairs, provided that the semantics for attributes and their allowed values is clear. This class contains data about environmental conditions and technological infrastructure but only few attributes regarding the user and socio-cultural information can be modeled in this way. We represent this type of data by CC/PP profiles. A CC/PP profile is an RDF graph composed by sets of *components* each one containing various *attributes* with associated values. Components and attributes refer to the *CC/PP vocabulary* where they are formally defined. Since existing vocabularies mostly cover only hardware, software, and network capabilities of mobile devices, we extended them to include a much richer set of context information.

2.2 Ontology-based profile data

Initially, in our framework we modeled all contextual data by using CC/PP [1]. However, CC/PP has many shortcomings when it comes to model non-shallow profile data like e.g., user activities. Indeed, CC/PP vocabularies define both the semantics of each attribute and the list of its possible values in natural language by using the `<rdfs:comment>` resource, leading to possibly different interpretations. Moreover, the 2-level structure (components and attributes) imposed by CC/PP greatly affects its expressive power. For representing non-shallow profile

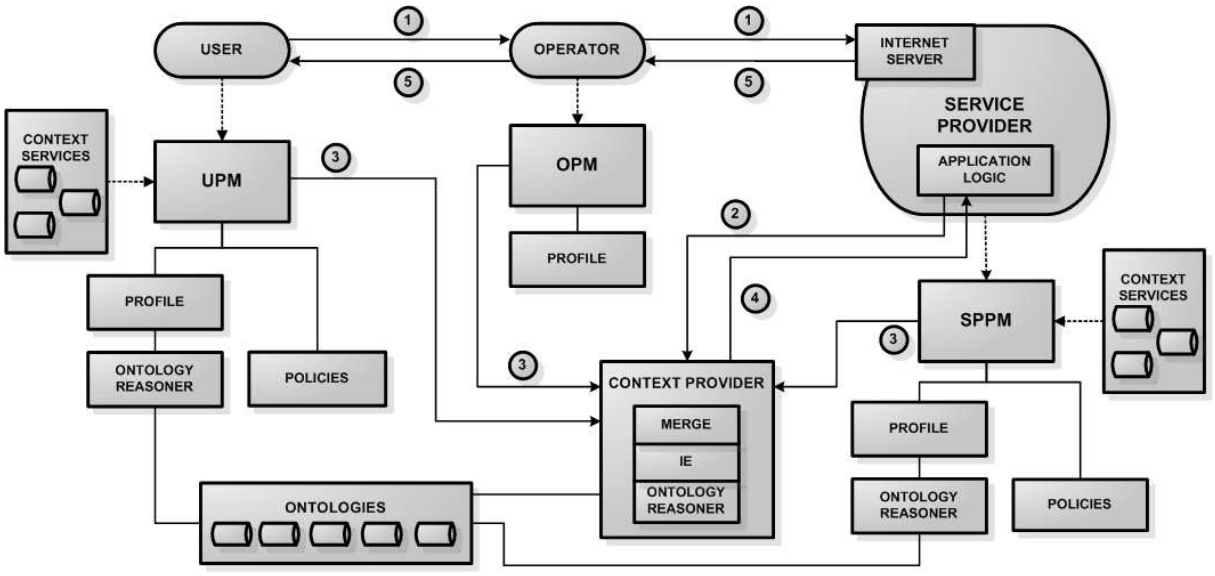

Figure 2. Architecture Overview.

data a natural choice is using ontologies; in fact, they have a higher expressive power than CC/PP and, in most cases, offer reasoning services. In our framework adopting ontologies has two main purposes. First of all, public/shared ontologies support knowledge sharing among the various involved entities. For instance, if user's interests/expertise are described via a shared ontology the service provider can correctly interpret them without risking misunderstandings. Naturally, in these cases, the ontology hierarchy can be walked through for looking for more specialized/general terms. Secondly, ontologies are used for consistency check of contextual data instances and for reasoning. E.g., automatically deriving, based on other context data, that the user is indeed busy in an "InternalMeeting". In this second case, ontologies can be private to a specific profile manager.

We currently use OWL-DL [15] as the ontology language, because both we want to take advantage of the reasoning services it supports and it is becoming a de-facto standard in various application domains. However, mainly for interoperability purposes, differently from other approaches [9, 23], we decide to keep storing all of our profile data in CC/PP profiles, but linking those attributes modeling non-shallow context data to ontology concepts that formally define their semantics. In order to adhere to the CC/PP specification, the mapping between a CC/PP attribute and an ontology concept is defined in the vocabulary which defines the attribute, using the `<rdfs:comment>` resource.

Figure 1 shows an excerpt of a profile containing both kind of attributes. The attribute belonging to the first component refers to the UAProf [22] CC/PP vocabulary and represents the available storage memory of the user's device. On the contrary, the second component refers to an ontology modeling user activities (see Section 4.1.1). The semantics of the attribute *currentLocationType* and of its value (*"outdoor"*) is provided by the fragment of the ontology defining properties and relationships of the concepts related to "places".

3 A middleware architecture for adaptation

The middleware architecture we consider in this paper is an extension of the one we presented in [1] to take into account ontologies and ontological reasoning. Our framework is based on a notion of extended profile data, which cover all the aspects that can be useful for improving the adaptation of services. The service adaptation is determined by the evaluation of both service provider and user's policies (i.e., rules that determine adaptation parameters on the basis of context).

We consider three main entities involved in the task of building the integrated profile, namely: the *user* with his devices, the network *operator* with its infrastructure, and the *service provider* with its own infrastructure. Clearly, the architecture has been designed to handle an arbitrary number of entities. A Profile Manager devoted to manage profile data is associated with each entity and will be called `UPM`, `OPM`, and `SPPM`, respectively. We assume that the user's location is kept up-to-date by an external *location server* that communicates it to the `UPM`. The `UPM` and `SPPM` are also in charge of managing local adaptation policies, and of interacting with ontology services. The motivation for having local ontology reasoners is twofold. On one side,

we believe that each entity, other than accessing shared ontologies, may hold private ones and may use values in ontological reasoning that should remain private. On the other side, this choice enables what we call on-demand ontological reasoning, which will be described in Section 4.2. Ontological reasoning is not performed by the OPM, since it only manages shallow context data. Ontological reasoning is not performed by user's devices too, due to the limited capabilities of mobile appliances. As a matter of fact, one of the advantages of having a profile manager dedicated to user-side data is to allow the execution of computationally-intensive reasoning tasks that would be at least unsuitable to execute on a mobile device.

We illustrate the system behavior by describing the main steps involved in a service request (see Figure 2). At first (step 1) a user issues a request to a service provider through his device and the connectivity offered by a network operator. The HTTP header of the request includes the URIs which are used to contact the UPM and the OPM. Then (step 2), the service provider queries the Context Provider to retrieve the profile information needed to perform adaptation. In step 3, the same module queries the profile managers to retrieve profile data and user's policies. Profile data are aggregated by the MERGE module in a single profile which is given, together with policies, to the IE (Inference Engine) for policy evaluation. On-demand ontological reasoning is performed only if the integrated profile lacks values for ontology-based profile data that are necessary for providing the service. In this case, the Context Provider populates the ontology with the integrated profile, performs ontological reasoning, and adds the new context information to the integrated profile. In step 4, the integrated profile is returned to the service provider. Finally, profile data are used by the application logic to properly adapt the service before its provision (step 5).

Currently, our framework supports a business model in which the Context Provider entity belongs to the trusted domain of the service provider. However, with slight modifications our framework can support different business models (e.g., the Context Provider could belong to the trusted domain of the network operator).

The mechanisms of profile aggregation, policy evaluation, and conflict resolution are based on prioritization techniques and are presented in detail in [6]. Issues regarding profile data acquisition, user interfaces, data privacy, and intra-session adaptation have been addressed elsewhere [1, 6, 20] and are not the focus of this paper.

4 Ontological Reasoning

Ontological reasoning is supported by ontology languages, like OWL-DL, that can be mapped to certain classes of description logics [3]. Reasoning services are based on subsumption computation for these logics and usually include consistency and classification, as well as checking for instances of specific concepts based on their properties. While web ontologies are relatively new, research on subsumption computation is not, and it is well-known to be intractable even for relatively simple logics [3]. Despite the progress made by reasoner implementations, the delay that the reasoning services inevitably introduce in service provisioning is one of the motivations for performing ontological reasoning off-line at each profile manager. In selected cases only, ontological reasoning can be performed on-demand by the Context Provider. In particular, on-demand ontological reasoning is performed when the profile lacks some crucial context information that can be possibly obtained after populating a shared ontology with the integrated profile.

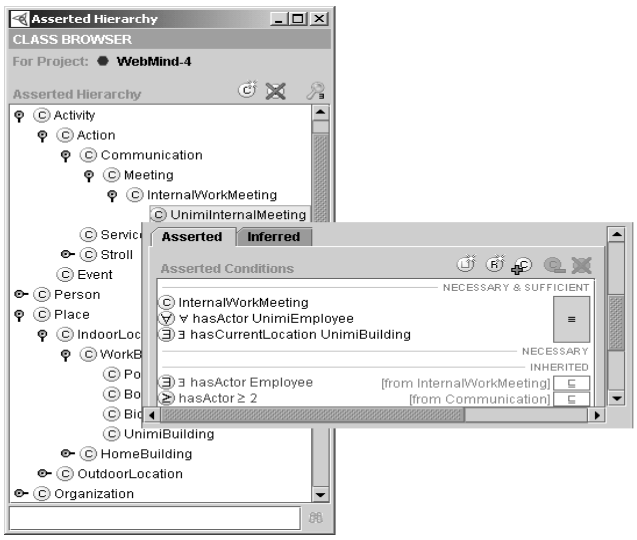

Figure 3. An excerpt of an ontology modeling the socio-cultural context of mobile users

4.1 Off-line ontological reasoning

User and service provider profile managers use shared and private ontologies to represent non-shallow context data as explained in Section 2.2. The goal of ontological reasoning is to provide the additional services of consistency checking, and of implicit context data derivation.

4.1.1 An ontology for modeling the socio-cultural context of mobile users

Figure 3 shows part of the OWL-DL ontology we defined for modeling the socio-cultural environment of mobile users. The ontology is intended to be (locally) maintained by an entity trusted by the user, in our case the UPM; in

fact, it is populated with user's sensitive data. The ontology is composed by nearly 150 classes and relations that describe features among which there are the user activities (actions, movements, ...), interests, contacts, calendar items, and places. As an example, the *UnimiInternalMeeting* and *UnimiEmployee* classes are defined as follows (see also Figure 3):

$$UnimiInternalMeeting \equiv Activity \sqcap \geq 2\,Actor \sqcap \forall Actor.UnimiEmployee \sqcap \exists Location.UnimiLocation$$

$$UnimiEmployee \equiv Employee \sqcap \exists Employer.\{unimi\}$$

In order to model some more general concepts, such as time and place, we adopted well-known ontologies (e.g., DAML-Time).

A detailed description of the ontology is outside the scope of this paper. However, Example 1 shows an application of this ontology for determining the current activity of a user, and Section 5 briefly reports preliminary performance results about off-line reasoning with this ontology.

4.1.2 Consistency checking and derivation of implicit context data

Consistency is crucial in the definition of an ontology as well as in its population. Indeed, when the ontology is populated with instances obtained from local repositories of the profile manager, consistency checking is performed in order to capture possible inconsistencies (e.g., the same instance belonging to disjoint classes, a person localized in different rooms at the same time).

More importantly, ontological reasoning is performed by the UPM and by the SPPM in order to derive new context data. The ability of ontological reasoning to derive new context data introduces the second argument in favor of this technique: privacy. Indeed, when ontological reasoning is performed by the UPM, the profile manager may release to the service provider some high level context information without releasing details that have been used to derive that information. The following example illustrates this aspect:

Example 1 *Consider the case of Alina, the user of an adaptive messaging service. The service properly filters and redirects messages by considering various context data, including the user current activity. User activities are modeled by the ontology described in Section 4.1.1. Alina is currently in her office with Will, a colleague of hers. Her calendar has an entry about a scheduled meeting between them, and she keeps Will's contacts in her cell phone. Since the* UPM *is a trusted entity, it has read access to all these data. Since Alina and Will are both employed by the same organization and are currently together in a place belonging to that organization, the ontology reasoner derives that their current activity (and in particular, Alina's current activity) is "InternalWorkMeeting". Hence, upon the* Context Provider *request, this information is given to the service provider, which accordingly applies an adaptation policy, for example, redirecting non-priority calls to Alina's answering machine.*

Note that the service provider only knows Alina's current activity, and ignores other sensitive information such as her current location, contacts, and people she is with.

4.1.3 Activation rules for ontological reasoning execution

The execution of off-line ontological reasoning is controlled by each single profile manager (UPM and SPPM) by means of *ontological reasoning activation rules*. In fact, each profile manager knows exactly which contextual data are modeled by its own local ontologies and therefore under which conditions–e.g., a change in a specific context attribute–ontological reasoning can, possibly, produce new implicit contextual data. For instance, the UPM can decide to execute ontological reasoning any time a user add a new appointment on his calendar for attempting to better specify user activity. These activation rules are essentially conditions over changes in profile data that fire the execution of ontological reasoning when met. For instance, the following rule determines the execution of ontological reasoning when the user's location changes of more than 100 meters:

If *changes(currentLocation,100m)* **then** *execOntReas*

Of course, both the UPM and the SPPM are provided with a monitor module, which is in charge of monitoring changes in context data that can fire activation rules. Whenever a rule fires, the profile manager executes ontological reasoning and properly updates profile data.

4.2 On-demand ontological reasoning

In particular cases contextual data can be derived through ontological reasoning only populating the ontology with information provided by different entities. In this case, reasoning must be performed on-demand at the time of the service request. For the sake of simplicity, we describe the mechanism of on-demand ontological reasoning by means of the following example.

Example 2 *Consider the case of John, the user of a location-based recommendation service. The service provides to mobile users a list of events ordered accordingly to their proximity to the user and to the user's specific interests. Suppose that John submits a query regarding*

music to the service. For this reason, the attribute MusicPreferences *is crucial for provisioning the service. Suppose the integrated profile does not contain entries for the* MusicPreferences *attribute, but only a list of John's preferred artists. So, the* `Context Provider` *populates a shared ontology modeling music genres and artists with the integrated profile. Then, the* `Context Provider` *performs ontological reasoning, inferring from John's preferred artists that his favorite music genre is R&B. Finally, the* `Context Provider` *adds this information to the integrated profile, and the application logic orders the events accordingly.*

In order to specify which attribute values are necessary for provisioning a specific service, the service provider can mark part of profile data as *crucial*, using the interface for inserting rules and required context data.

When, after the evaluation of rules by the IE, an attribute marked as crucial does not have associated values, the `Context Provider` performs on-demand ontological reasoning, populating a shared ontology with data obtained from the integrated profile. These data possibly comprehend contextual information that can assist ontological inferences, which are not available before the user's request. After having populated the ontology, the `Context Provider` performs ontological reasoning in order to derive values for crucial attributes, and adds their values (which can possibly be null) to the integrated profile. Finally, the integrated profile is sent to the service provider application logic. If some crucial attributes still have no values, the application logic can choose to deny the service, or to ask the user to explicitly specify a value for them. In Section 5 we describe an application of the mechanism for adapting a real service.

5 Experimental Evaluation

The main software modules of the architecture have already been developed [1]. In order to demonstrate the validity of our proposed solutions we have also developed some prototype services which make use of ontological reasoning (both off-line and on-demand) in order to perform adaptation. Figure 4 shows some screenshots of an adaptive location-based service [2, 5] that exploits our framework for providing *POIsmarts* to mobile users. POIsmarts can be seen as the convergence of bookmarks as used in web browsing and *points of interest* as used in GPS navigation systems. A POIsmart stores information about a virtual and/or physical location which is of interest to a user, as well as multimedia content associated with the resource (e.g., a vocal comment to the quality of a restaurant). The list of POIsmarts is delivered to the mobile devices considering context data that goes beyond user location and device capabilities, including non-shallow information such

Figure 4. The POIsmart application

as the user's interests and current activity. As an example, screenshots in Figure 4 show the implementation of the on-demand reasoning scenario presented in Example 2. Suppose that the user (John) submits a generic query regarding "music" to the POIsmart system (screenshot A), and that John's profile manager does not provide information regarding his music preferences, but only a set of his preferred artists. Then, since the *MusicPreferences* attribute is crucial for answering the user's query, the *Context Provider* asks the SPPM for on-demand ontological reasoning, providing it with the set of John's preferred artists. The SPPM populates its ontology with these data and derives –through ontological reasoning– John's preferred music genres, which in turn determine the order of POIsmarts returned to the user (screenshot B).

We also performed some preliminary experiments on executing off-line reasoning with the OWL-DL ontology we defined for modeling the socio-cultural environment of mobile users presented in Section 4.1.1. We recall that the

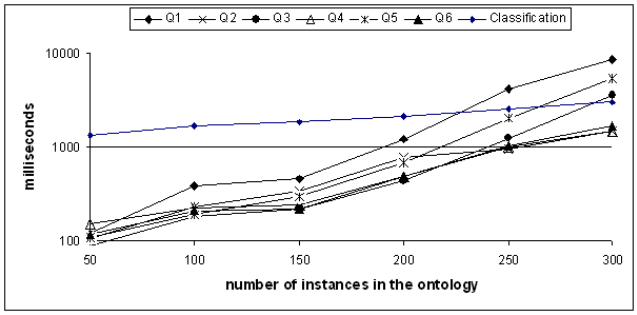

Figure 5. Performance results about ontological reasoning with socio-cultural information

ontology contains nearly 150 classes and relations. The reasoning tasks we performed correspond to submitting queries regarding the instances of a specific class of the ontology. Queries are evaluated using the Racer [13] ontology reasoner, on a two-processor Xeon 2.4 GHz workstation with 1.5GB of RAM. . Experiments are conducted populating the ontology classes with an increasing number of instances. Queries from $Q1$ to $Q6$ are made on different classes (e.g., *ContactProfile*, *WorkMeeting*) having an increasing number of instances (from 1 to 248, depending on the query and on the total number of instances). Results are shown in Figure 5, in logarithmic scale. Query response times exhibit a strong correlation with the number of instances. In the actual implementation of our prototype services, we could verify that the UPM ontologies are generally filled with less than 200 instances, and thus queries are executed in less than two seconds. These results show that off-line ontological reasoning with our ontology is feasible on dedicated servers, when the ontology is populated with few hundreds of instances. Clearly, when performed at the time of the service request, the delay introduced by ontological reasoning executed by servers dedicated to multiple users at a time can affect the scalability of the system. It is the responibility of the service provider to cautiously select (using triggers) the conditions that determine the execution of on-demand ontological reasoning, in order to preserve the scalability of its provisioning infrastructure.

6 Related Work

The adoption of ontologies for context-awareness purposes is not new and it is increasing in the last few years. For instance, in CoBrA [8] the context is modeled via a shared OWL ontology. The same group is working on defining SOUPA [9], a standard ontology for the specific domain of ubiquitous and pervasive applications. The main purpose, in this case, is to support knowledge sharing and interoperability in ambient intelligence scenarios, where efficiency is not the main focus. The context modeling of SOCAM is based on ontologies too [12]. A centralized module collects context information and a reasoning engine evaluates first-order logic rules for inferring new context information. Similarly to SOUPA, Wang et al. [23] define the CONON ontology which is composed by a general-purpose upper ontology and by application-specific lower ontologies. Reasoning is performed real-time and is based on both description logic and user-defined logic rules. However, they admit that this approach is unsuitable for time-critical applications.

The possibility of overcoming the restrictions of rule-based and description logic (DL) reasoning through a form of combination of these classes of languages has been widely investigated. Early proposals date back to some of the so-called "second generation DL systems" (e.g., CLASSIC [7] and LOOM [19]). This research field has recently gained new momentum from the semantic web community. One of the main research issues in that area is to provide very expressive DL languages with the possibility of performing powerful reasoning tasks not only on terminological knowledge (classes and relations) but also on assertional knowledge (instances). To this end, SWRL [14] extends OWL-DL and OWL-Lite with Horn clauses. The extended languages overcome most of the expressive restrictions of the primitive ones, but reasoning tasks in the new languages are undecidable, and the development of optimized tools for reasoning with SWRL subsets is currently at an early stage. Other recent proposals try to combine DL and logic programming while keeping decidable reasoning, imposing constraints on the form of the rules [21] and/or adopting quite simple DL languages [11]. However, decidability does not guarantee in general that reasoning is computationally feasible. Thus, in our opinion similar approaches are unsuited for modeling context for the provision of adaptive real-time services, especially if –in particular cases– reasoning must be performed at the time of the service request. For these reason, in our framework ontological and rule-based reasoning are performed separately. The main characteristic of this approach is that the evaluation of rules does not affect the assertional part of the ontology (*ABox*), i.e., the information flow is one-way from the ABox to the logic program knowledge base. This feature clearly limits the expressive power of the language, compared with logics in which the information flow is bi-directional (e.g., [14, 21]). The main benefit of this approach is that complexity remains the same of the adopted policy language (i.e., linear in the number of rules) when on-demand ontological reasoning is not performed; complexity is the same of the description logic reasoning tasks in the other case.

7 Conclusions and future work

In this paper we addressed the problem of introducing ontological reasoning in a middleware architecture supporting adaptation based on distributed context data and policies. Our investigation led to the choice of adopting OWL-DL ontologies to represent non-trivial aspects of context, and to prefer forms of off-line ontological reasoning, while resorting to on-line ontological reasoning only when strictly required. Our findings and technical choices are supported by implementation and experiments. As an extension of this work we are considering to adopt a more expressive ontology language supporting, for example, limited forms of property composition.

References

[1] A. Agostini, C. Bettini, N. Cesa-Bianchi, D. Maggiorini, D. Riboni, M. Ruberl, C. Sala, and D. Vitali. Towards Highly Adaptive Services for Mobile Computing. In *Proceedings of IFIP TC8 Working Conference on Mobile Information Systems (MOBIS)*, pages 121–134. Springer, 2004.

[2] A. Agostini, C. Bettini, and D. Riboni. Demo: Ontology-based Context-aware Delivery of Extended Points of Interest. In *To appear in Proceedings of the 6th International Conference on Mobile Data Management (MDM'05)*. ACM, 2005.

[3] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.

[4] C. Becker, G. Schiele, H. Gubbels, and K. Rothermel. Base: A micro-broker-based middleware for pervasive computing. In *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom) 2003*, pages 443–451, 2003.

[5] C. Bettini, N. Cesa-Bianchi, and D. Riboni. A Distributed Architecture for Management and Retrieval of Extended Points of Interest. In *To appear in Proceedings of the first International Workshop on Services and Infrastructure for the Ubiquitous and Mobile Internet (SIUMI'05)*. IEEE, 2005.

[6] C. Bettini and D. Riboni. Profile Aggregation and Policy Evaluation for Adaptive Internet Services. In *Proceedings of The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (Mobiquitous)*, pages 290–298. IEEE, 2004.

[7] A. Borgida, R. J. Brachman, D. L. McGuinness, and L. A. Resnick. CLASSIC: A Structural Data Model for Objects. In *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data*, pages 58–67, 1989.

[8] H. Chen, T. Finin, and A. Joshi. Semantic Web in the Context Broker Architecture. In *Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications (PerCom 2004)*, pages 277–286. IEEE Computer Society, 2004.

[9] H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In *Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Boston, MA, August 2004.

[10] A. K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing. Special issue on Situated Interaction and Ubiquitous Computing*, 5(1), 2001.

[11] B. Grosof, I. Horrocks, R. Volz, and S. Decker. Description Logic Programs: Combining Logic Programs with Description Logics. In *Proceedings of the 12th International Conference on the World Wide Web (WWW-2003)*, 2003.

[12] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang. An ontology-based context model in intelligent environments. In *Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, San Diego, California, USA*, January 2004.

[13] V. Haarslev and R. Mller. RACER System Description. In *Proceedings of Automated Reasoning, First International Joint Conference (IJCAR 2001)*, volume 2083 of *Lecture Notes in Computer Science*, pages 701–706. Springer, 2001.

[14] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3c member submission, W3C, May 2004.

[15] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From SHIQ and RDF to OWL: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[16] R. Hull, B. Kumar, D. Lieuwen, P. Patel-Schneider, A. Sahuguet, S. Varadarajan, and A. Vyas. Enabling Context-Aware and privacy-Conscius User Data Sharing. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management*, pages 187–198. IEEE, 2004.

[17] J. Indulska, R. Robinson, A. Rakotonirainy, and K. Henricksen. Experiences in using CC/PP in context-aware systems. In *Proceedings of the 4th International Conference on Mobile Data Management*, pages 247–261. Lecture Notes in Computer Science, Springer Verlag, LNCS 2574, 2003.

[18] G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. H. Butler, and L. Tran. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0. W3C Recommendation, W3C, January 2004. http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/.

[19] R. MacGregor and R. Bates. The LOOM Knowledge Representation Language. Technical Report ISI/RS-87-188, Information Science Institute, University of Southern California, Marina del Rey (CA), USA, 1987.

[20] D. Maggiorini and D. Riboni. Continuous Media Adaptation for Mobile Computing Using Coarse-Grained Asynchronous Notifications. In *2005 International Symposium on Applications and the Internet (SAINT 2005), Proceedings of the Workshops*. IEEE Computer Society, 2005.

[21] B. Motik, U. Sattler, and R. Studer. Query Answering for OWL-DL with Rules. In *The Semantic Web ISWC 2004: Third International Semantic Web Conference*, pages 549–563, 2004.

[22] OpenMobileAlliance. User Agent Profile Specification. Technical Report WAP-248-UAProf20011020-a, Wireless Application Protocol Forum, October 2001. http://www.openmobilealliance.org/.

[23] X. H. Wang, T. Gu, D. Q. Zhang, and H. K. Pung. Ontology Based Context Modeling and Reasoning using OWL. In *Proceedings of Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 18–22. IEEE Computer Society, 2004.