Exploiting Ontologies to Achieve Semantic Convergence Between Different CC/PP-like RDF Schemes for Representing Devices' Capabilities: the SADiC Approach

Francesco Cannistrà
fracan@inwind.it
The-Web-Middle-Earth.com

ABSTRACT. *The CC/PP and the UAProf are two related frameworks that aim at defining a general and extensible format to describe the capabilities of the user-end terminals for accessing contents and services provided by the Internet and by the Web in particular. Both CC/PP and UAProf are based on RDF and have logically equivalent architectures. However, notwithstanding their logical bindings, they appear to be parallel standards, i.e., equivalent but not compatible. This paper explains the reasons for the incompatibility between CC/PP and UAProf and presents the approach followed by the Semantic API for the Delivery Context (SADiC) in order to achieve rigorously the required semantic convergence between these frameworks – as well as, in general, between all the CC/PP-like RDF schemes – by exploiting the concepts of the Semantic Web, without influencing the standards' bodies themselves.*

1 Introduction

Nowadays the edge population of the Internet is growing through the proliferation of heterogeneous and special purpose terminals (e.g. mobile devices) hooked up to specific network access channels (e.g. wireless networks) and offering to the users intrinsically limited service fruition capabilities. In this new scenario the users' expectation to access the services provided by the Internet (and by the Web in particular) pervasively – regardless of the specific characteristics of the device used from time to time – is fostering the service providers to engage issues regarding the device independent provision of information contents [3]. The parameters that can influence the way a user perceives and enjoys contents are many and span from the capabilities of the used device and its equipments to the constraints imposed by the network access channel, possibly including also the preferences of the user. The set of all these attributes that characterize a client fruition environment is called the *delivery context* [3, 4]. Provided with the delivery context information, the Web servers should be able to select or to adapt the output of a service for the specific requirements of the client that requested it in order to deliver a functional representation of contents that is suitable for their fruition by means of the peculiar characteristics of the *access mechanism* exploited by each user [4].

Recently the Composite Capability/Preference Profiles (CC/PP) [6, 7] – being developed by the World Wide Web Consortium (W3C) [2] – and the related User Agent Profile (UAProf) [9, 10, 11] – from the Open Mobile Alliance (OMA, formerly the WAP Forum) [8] – are emerging as standards that define a general format for expressing the delivery context information by means of profiles. They are both based on the W3C Resource Description Framework (RDF) [5] and they describe a profile as a structured set of RDF assertions. Even tough the documents specifying the CC/PP and the UAProf emphasize the need for converging the two frameworks as a shared belief, in point of fact the two working groups have proceeded almost in parallel with their respective standardization efforts so that, from a rigorous point of view, the CC/PP and UAProf now appear as equivalent but not compatible standards.

The Semantic API for the Delivery Context (SADiC) [1] acknowledges the problems actually affecting the interoperability of CC/PP with UAProf. SADiC is a Java API for processing and interrogating CC/PP and UAProf profiles. SADiC provides many features and, in particular, it succeeds in achieving rigorous semantic convergence between CC/PP and UAProf – as well as between all the RDF-based schemes implementing the basic semantics of CC/PP.

The remainder of this paper is structured in order to introduce gradually the approach of SADiC to achieve the required semantic convergence between CC/PP and UAProf. Section 2 introduces the CC/PP, focusing on its original aspects concerning the addressing of interoperability and extensibility issues. Section 3 discusses the reasons because of which UAProf is not compatible with CC/PP. A certain emphasis is given to these two sections, since it's the author's opinion that the points there discussed have not yet been taken in the right consideration by the research community. Then section 4 presents the approach of SADiC and section 5 concludes the article.

Even tough some efforts have been spent to present the contents of this paper as clearly as possible, it would be preferred that, in order for a full comprehension of the paper, the readers have, at least, a basic knowledge of RDF.

2 The CC/PP as an extensible framework providing interoperability

The CC/PP aims at defining an extensible framework as a basis for interoperability of applications that exchange delivery context information on the Internet and on the Web in particular.

Basically, the CC/PP is founded on two main ideas. Firstly, it introduces a semantic structure for representing the delivery context information by means of profiles, and provides the formal means to instance and to recognize such a structure. Secondly, it provides the formal means to define the vocabularies of attribute properties that can be used to populate the structure of a profile in order to express the specific attributes of an actual delivery context. It's in this way that CC/PP tries to address the interoperability-extensibility binomial:

1. different applications interoperate by sharing the concept of profile and the formal means to instance and to recognize this concept: profiles constructed by an application are recognizable by all others;
2. the information that can be conveyed by a profile is extensible: each application can create its own vocabulary that defines attributes useful to represent specific capabilities, and such a vocabulary can even be used by all other applications (possibly in conjunction with other vocabularies) to construct profiles.

To implement a framework with such prerogatives, CC/PP founds itself on RDF. The CC/PP does define a RDF vocabulary acting as a shared *structural vocabulary* that is the backbone of the entire conceptual framework, since it defines the RDF constructs to be used in order to instance the structure of a profile through the RDF data model, and the RDF primitives to be extended in order to define vocabularies of attributes by means of RDF schemas. This way, profiles are constructed by instancing always the same skeleton structure and then by populating this structure with actual attributes taken from different vocabularies defined by time.

A CC/PP profile can be viewed as a two-levels hierarchical structure made up of components and attributes: the attributes represent the specific capabilities of the delivery context being described, while the components group these capabilities possibly with respect to a certain global aspect (e.g., hardware or software characteristics). Figure 1 shows an excerpt of an hypothetical vocabulary defining two component types (i.e., *voc:Hardware* and *voc:Software*) and the associated attribute properties (e.g., *voc:ScreenSize* and *voc:JavaCapable*), and then shows how such a vocabulary can be exploited to build an actual profile.

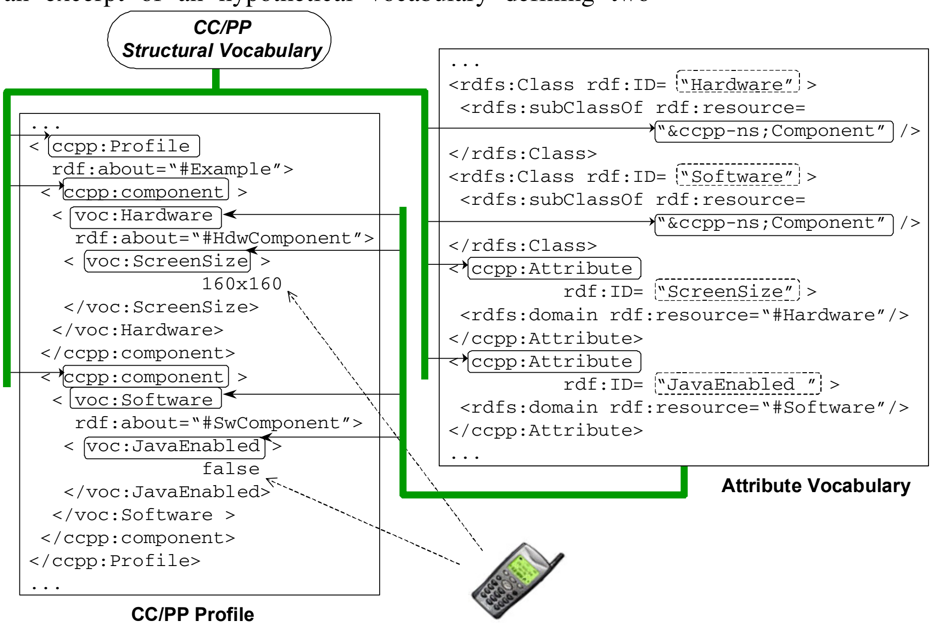

Fig. 1. A simple example showing how to define vocabularies and to construct profiles through CC/PP.

3 UAProf and its incompatibility with the CC/PP

CC/PP is vocabulary-agnostic, in the sense that it does not aim at defining any specific vocabulary of attributes that could be exploited to describe the characteristics of an actual delivery context. The CC/PP schema is just the backbone of a conceptual framework for defining vocabularies and for utilizing them in order to express the capabilities of an actual delivery context by means of an RDF description (i.e. a profile). On the contrary, the UAProf was originally invented as a specific extension of CC/PP mainly aiming at defining a rich vocabulary of attributes for constructing actual profiles. UAProf was designed to be broadly and seamlessly interoperable with the CC/PP. A precise and explicit goal of its creators was to build UAProf on the model of CC/PP as a specific implementation of it that would have also provided a vocabulary of attributes for constructing the profiles of a large range of terminals (WAP devices in particular).

Unfortunately, the development of the two frameworks has reached a status at which, if we look at them from a rigorous point of view, they can be considered only parallel, i.e., equivalent but not entirely compatible standards. The incompatibility ensues from the fact that UAProf does not rely on the RDF elements defined in the CC/PP structural vocabulary. Instead, the RDF schema introducing the UAProf vocabulary also replaces the definition of the RDF elements sustaining the CC/PP conceptual structure. The structural semantics defined by the UAProf schema is almost the same as in the CC/PP schema, but, since the RDF elements for utilizing in practice the corresponding structural concepts are tied to a different naming space, such a logical equivalence cannot be recognized at RDF level.

In fact, one of the basic principles of RDF in order to provide a language for expressing *machine-understandable* facts is that a vocabulary schema, on the one hand, explicitly introduces unequivocal terms (through URI references) – and either expresses the constraints on their use – and, on the other hand, implicitly identifies the semantics of the terms themselves. In this way, a machine, provided in advance with the knowledge base of a vocabulary and being able to recognize unambiguously the terms on which this relies, can associate terms with concepts and actual resources, and so can deduce actual relationships and meanings. The name-spacing is just an additional facility not directly related to the RDF data model. Tying the vocabulary-defined elements to an univocal naming space helps applications to recognize groups of terms relating to the same vocabulary's context, and permits the reuse of identical (tough relative) terms within different contexts. However, since a naming space is unambiguously identified by a namespace URI that is either the shared prefixed part of all the vocabulary-defined terms, terms tied to different naming spaces denote different concepts for an RDF engine. Therefore, since both the CC/PP and UAProf are based on RDF and since the RDF elements they provide for leveraging semantically-equivalent structural concepts are defined through different RDF schemas with different namespaces, it is a consequence that they appear as different RDF applications that, although equivalent, are not compatible.

Another related problem concerns extensibility. As we have seen in section 2, the basic idea of CC/PP to achieve extensibility is that the structural vocabulary can be exploited to define whatever actual vocabulary of attributes so that, provided that the profiles' structure does not rely on any specific attribute vocabulary (but is instanced through the RDF properties defined by the CC/PP schema), a profile can be populated with actual attributes coming from different vocabularies. Moreover, if an application already defined its own vocabulary (or is using an existing one) and wants to extend this vocabulary by adding new attributes, then it should formally define a new vocabulary schema that contains the definition of the added attributes only. This way, the core vocabulary used by the application would look like a super-vocabulary made up of a set of vocabularies defined throughout subsequent schemas, and so the semantic integration between profiles that reference the different vocabularies would be assured as well.

UAProf did not acknowledge this basic idea because it intended the possibility to extend or to make corrections to the vocabulary it introduces as if each time the vocabulary schema could be completely redefined (including the basic structural concepts) by replicating it with just a few modifications and then tying the updated version to a new namespace URI. As a consequence, it was attained a situation where there exist multiple instances of the UAProf schema with different namespaces and each one of these is formally incompatible with each other for analogous reasons as those explained above when comparing the CC/PP and the UAProf in general.

It is straightforward that the problems outlined in this section are a serious hindrance to the use of the CC/PP and UAProf in wide practice and make the authoring of profiles and the development of profile processors quite cumbersome, since the risk for both profiles and processors to be not widely compliant or to became suddenly meaningless is more than concrete. However, the most important concern should be about the assurance of having wide semantic compatibility at RDF level, so that the really original prerogatives of CC/PP can be actually exploited and can then provide the intended advantages as regards interoperability and extensibility. In fact, if vocabularies and profiles were created basing on always different RDF schemes that, although intended to rely on the conceptual structure of the CC/PP, are not compatible with this structure at RDF level, then the advisability itself to have built CC/PP on top of RDF would not make sense any more.

In summary, the RDF heterogeneity between CC/PP and UAProf is paradoxically leading the Web accessibility towards a vertical segmentation as depicted in figure 2. A restatement of UAProf that would obey more to the basic interoperability principles of the CC/PP and of RDF would certainly improve the situation and would be an auspicial as well. However, this would not solve the problems at all, since the industry manufactures have already started to use UAProf and, at the moment, virtually all the CC/PP enabled devices use UAProf, provided that it also defines a rich vocabulary of actual attributes that can be utilized in practice to express device capabilities. Therefore, a general and rigorous approach that would assure of formal semantic interoperability without affecting the state of the art with the standards is now, of course, required.

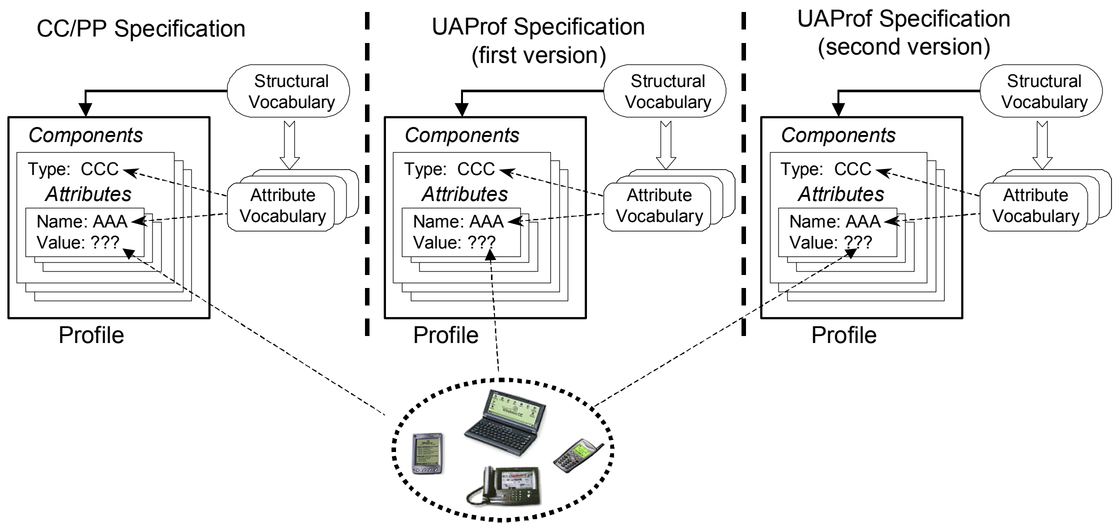

Fig. 2. The vertical segmentation of the Web accessibility ensuing from the lack of interoperability.

4 The approach of SADiC for the semantic integration of CC/PP and UAProf

Provided with the full understandings of the interoperability problems pointed out in the previous section, the Semantic API for the Delivery Context (SADiC) has been designed also to achieve rigorous semantic integration between CC/PP and UAProf – and, in general, between all the RDF-based schemes that provide a parallel implementation of the CC/PP conceptual framework – so that all these schemes can be used concurrently or jointly in wide practice, being assured of semantic interoperability without the need for any particular effort. For this aim, SADiC exploits the concepts of the Semantic Web [12] and, in particular, the notion of *ontology*.

Basically, an ontology is a collection of axioms that describe computer-usable concepts – and either introduce the vocabulary of terms that relate to them – in the perspective of representing a *domain* (i.e., an area of knowledge) for *machine-understanding* purposes. The basic idea introduced by SADiC exploits the fact that the knowledge encoded by an ontology can be imported and reused by other domains. In this way, it is possible to build complex domains that grow each over each other and that represent different levels of abstraction of the same knowledge base, possibly specializing and/or extending and/or enhancing this for a particular application purpose.

SADiC defines and relies on a core ontology that expresses the abstract knowledge base required to build an RDF-based conceptual framework implementing the basic semantics of the CC/PP architecture. The elements defined by this ontology represent the semantic abstraction of the basic structural concepts introduced by the CC/PP – e.g., the abstract concept of *attribute property* (i.e., the class of RFD properties that express the delivery context attributes), the abstract concepts of *structural properties* (i.e., the RDF properties that let instance the semantic structure of a profile within an RDF data model) and the concept of profile component (i.e., the *Component* class that acts as the root component type for all profile components).

The key aspect of the core ontology is that it does not supersede the CC/PP structural vocabulary. The core ontology just represents the lowest level of abstraction of the logical domains corresponding to all the CC/PP-like conceptual frameworks, and houses formally the shared semantics of the basic structural concepts already introduced by the CC/PP specification, not the terms that are to be used to exploit these concepts in practice. An actual domain can import the basic concepts of the core ontology and map them to its own terms. Such a domain is intended as a structural domain, since it provides an effective naming space for the CC/PP concepts and allows to utilize them through the specific terms it defines. Therefore, many lexically-different but semantically-equivalent domains can be introduced: these all exploit the same semantics of the CC/PP structural concepts, but allow to refer to them through different terms afferent to different namespaces.

Let's consider, for example, the case of the structural vocabulary proposed by the CC/PP specification and the vocabularies corresponding to the various version of the UAProf specification. Within SADiC all these vocabularies are associated with domains that just host suitable terms to refer to the shared semantics of the CC/PP structural concepts, but that do not define the concepts themselves. Since these concepts are defined elsewhere – i.e. in the core abstract ontology – and the different terms through which they can be referenced are formally mapped to them, then the wished semantic convergence and cross-interoperability are achieved automatically and rigorously.

Figure 3 sketches a simplified view of the semantic hierarchy introduced by SADiC. Note that at the leaf level there are the pure application domains, i.e. the domains corresponding to the actual vocabularies of attributes for describing a delivery context, which are defined through RDF schemas that reference and utilize a structural vocabulary associated with a specific CC/PP structural domain.

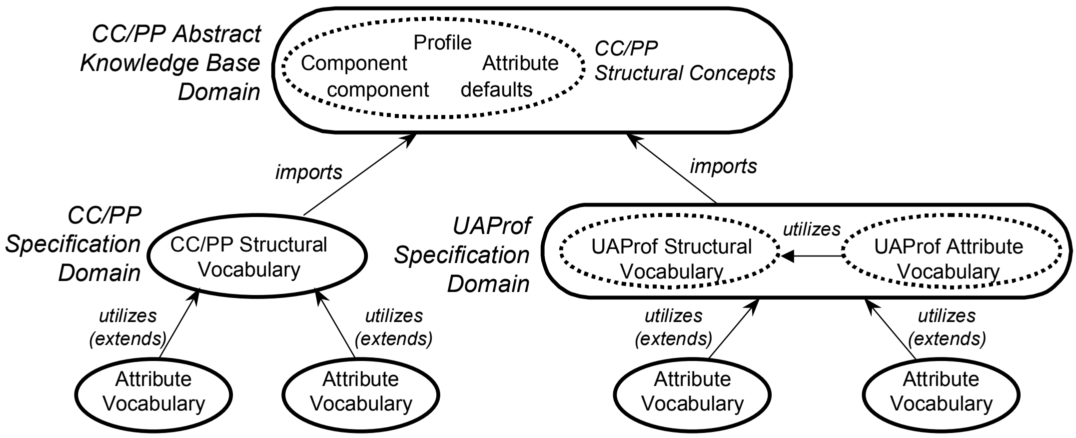

Fig. 3. The semantic hierarchy introduced by SADiC.

```
<owl:Ontology rdf:about="">
  <owl:sameAs rdf:resource="&ccpp-spec-schema-URI;"/>
</owl:Ontology>
<owl:Ontology rdf:about="&ccpp-spec-schema-URI;">
  <owl:imports rdf:resource="&ccpp-abs-ontology-URI;"/>
</owl:Ontology>
<owl:Class rdf:about="&ccpp-spec-schema-URI;#Component" >
  <owl:sameAs rdf:resource="&ccpp-abs-ontology-URI;#Component"/>
</owl:Class>
<owl:Class rdf:about="&ccpp-spec-schema-URI;#Attribute">
  <owl:sameAs rdf:resource=&ccpp-abs-ontology-URI;#Attribute"/>
</owl:Class>
<rdf:Property rdf:about="&ccpp-spec-schema-URI;#component">
  <owl:sameAs rdf:resource=&ccpp-abs-ontology-URI;#component"/>
</rdf:Property>
<rdf:Property rdf:about="&ccpp-spec-schema-URI;#defaults">
  <owl:sameAs rdf:resource=&ccpp-abs-ontology-URI;#defaults"/>
</rdf:Property>
```

Fig. 4. An excerpt of the OWL ontology for the structural domain of the CC/PP specification.

In order to define a domain, SADiC makes use of the Web Ontology Language (OWL) [14, 15, 16], the language for representing ontologies on the Web. OWL is based on RDF and is still being developed by the W3C as a component of the Semantic Web Activity [13]. SADiC requires that only the structural domains are to be explicitly defined: the pure application domains are defined implicitly by the corresponding RDF vocabulary schemas (provided that these schemas extend correctly the RDF schema defining the structural vocabulary of an already recognized structural domain). A structural domain is defined by means of a simple OWL ontology that expresses the basic facts that semantically make of such a domain a CC/PP structural domain. For this goal, it is sufficient to state that the terms introduced by an RDF schema (within its own naming space) to refer to the semantics of the CC/PP structural elements have the same *intentional meaning* as the concepts defined in the core abstract ontology – i.e., both RDF elements, though denoted by different terms, are semantically equivalent.

The listing in figure 4 shows a fragment of the OWL ontology introducing the domain corresponding to the structural vocabulary schema defined by the CC/PP specification.

This way, SADiC succeeds in mapping the semantics between structural vocabularies corresponding to different RDF-based schemes that provide a parallel implementation of the CC/PP architecture, and, therefore, the potential vertical segmentation depicted in the previous section is brilliantly avoided within a purely semantic context (see figure 5).

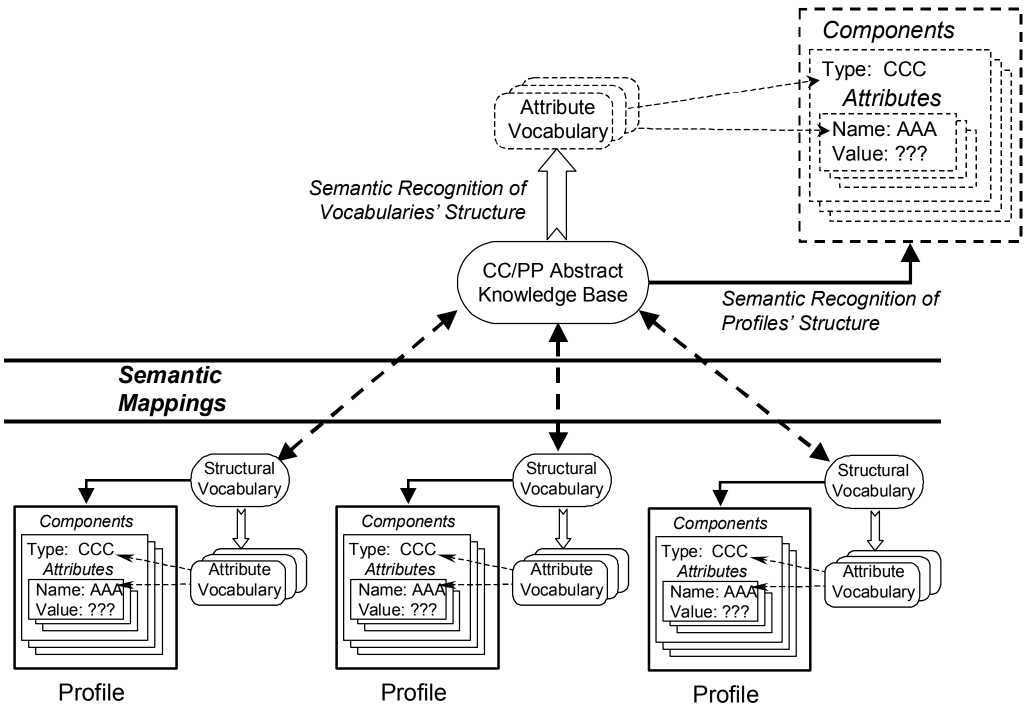

Fig. 5. How SADiC achieves interoperability.

Further to overcoming the RDF incompatibilities at structural level, SADiC also addresses problems related to the multi-versioning of attribute vocabularies. In particular, the ontology for specifying a domain can be exploited to assert that two RDF schemas – tied to different namespaces – are semantically equivalent or are subsequent versions of the same logical vocabulary. In such cases, SADiC is able to manage properly this kind of equivalence so that, for example, segmental profiles constructed relying on different versions of the same logical vocabulary can be merged together consistently.

As we have seen in section 3, the proliferation of multiple namespace URIs to refer to the same logical vocabulary is an incongruity actually affecting UAProf. In particular, there are two kinds of slightly different problems: the referencing of the vocabulary schema tied to a certain version of UAProf through different namespace URIs, and the extending the UAProf vocabulary through new RDF schemas that completely supersede the older ones (and have different namespaces either). To address the latter problem, it is sufficient to assert, through the OWL ontology defining the domain corresponding to a certain version of UAProf, that the RDF schema associated with such a version is the subsequent version of an earlier RDF schema:

```
<owl:Ontology rdf:about="&uaprof;">
  <owl:backwordCompatibleWith
      rdf:resource="&uaprof-previous;"/>
</owl:Ontology>
```

Note that the above statements also indicate that all the local terms tied to the previous naming space have the same intended interpretations in the naming space of the new version.

Instead, in order to assert that the defined domain could even be referenced through a namespace URI different from the canonical namespace URI of the considered UAProf version (and that acts as an alias for this), an *owl:backwordCompatibleWith* statement should be coupled with an *owl:sameAs* statement:

```
<owl:Ontology rdf:about="&uaprof-alias;">
  <owl:backwordCompatibleWith
      rdf:resource="&uaprof;"/>
  <owl:sameAs rdf:resource="&uaprof;"/>
</owl:Ontology>
```

5 Conclusions

This paper has discussed of the cumbersome problems actually thwarting the cross-compatibility between the CC/PP and the UAProf frameworks, and of the danger of attaining a vertical segmentation of the Web accessibility that would be quite the contrary of the original goals of the CC/PP. The paper has introduced the Semantic API for the Delivery Context (SADiC), showing how this approaches the abovementioned problems and succeeds in achieving the required formal semantic convergence between CC/PP and UAProf – as well as between all the RDF-based schemes that are intended to rely on a CC/PP-like conceptual architecture. The approach of SADiC exploits the notion of ontology in order to build an extensible hierarchy of semantically overlapping RDF domains, and uses the Web Ontology Language (OWL) in order to represent a domain and to map semantics between domains.

Even though SADiC focuses on a specific application context, its semantic approach introduces simple and general ideas that could be exploited in order to address analogous issues of semantic interoperability for other RDF-based contexts as well.

References

1. SADiC: the Semantic API for the Delivery Context, http://www.the-web-middle-earth.com/sadic/
2. World Wide Web Consortium (W3C), http://www.w3.org
3. Device Independence Activity (DIA) at W3C, http://www.w3.org/2001/di/
4. R. Gimson et al., "Device Independence Principles". W3C Note, Version 1 September 2003 available at: http://www.w3.org/TR/2003/NOTE-di-princ-20030901/
5. W3C Resource Description Framework (RDF), http://www.w3.org/RDF/
6. Early CC/PP Working Group at W3C, http://www.w3.org/Mobile/CCPP/
7. G. Klyne, F. Reynolds, C. Woodrow, H. Ohto, J. Hjelm, M. Butler, L. Tran et al., "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies". W3C Working Draft, Version 28 July 2003 available at: http://www.w3.org/TR/2003/WD-CCPP-struct-vocab-20030728/
8. Open Mobile Alliance (OMA), http://www.openmobilealliance.org
9. WAP Forum, User Agent Profiling Specification (WAP-174-UAPROF-19991110-a). November 1999, available from: http://www.wapforum.org/what/technical_1_2_1.htm (See also subsequent WAP SINs)
10. WAP Forum, User Agent Profiling Specification (WAP-248-UAPROF-20011020-a). October 2001, available from: http://www.wapforum.org/what/technical.htm (See also subsequent WAP SINs)
11. OMA, User Agent Profile Version 1.1. December 2002, available from: http://www.openmobilealliance.org/documents.html
12. T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web". Scientific American, May 2001.
13. Semantic Web Activity at W3C, http://www.w3.org/2001/sw/
14. Web Ontology (WebOnt) Working Group at W3C, http://www.w3.org/2001/sw/WebOnt/
15. D. L. McGuinness, F. van Harmelen et al., "OWL Web Ontology Language Overview". W3C Candidate Recommendation, Version 18 August 2003 available at: http://www.w3.org/TR/2003/CR-owl-features-20030818/
16. S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, L.A. Stein et al., "OWL Web Ontology Language Reference". W3C Candidate Recommendation, Version 18 August 2003 available at: http://www.w3.org/TR/2003/CR-owl-ref-20030818/
17. J. Heflin et al., "Web Ontology Language (OWL) Use Cases and Requirements". W3C Candidate Recommendation, Version 18 August 2003 available at: http://www.w3.org/TR/2003/CR-webont-req-20030818/