Journal of Mobile Multimedia, Vol. 6, No.1 (2010) 026-048

UBIQUITOUS WEB FOR UBIQUITOUS COMPUTING ENVIRONMENTS: THE ROLE OF EMBEDDED SEMANTICS

AHMET SOYLU PATRICK DE CAUSMAECKER

K.U. Leuven, Interdisciplinary Research on Technology Education and Communication (iTec)

Kortrijk, Belgium

Ahmet.Soylu@kuleuven-kortrijk.be, Patrick.DeCausmaecker@kuleuven-kortrijk.be

FRIDOLIN WILD

The Open University, Knowledge Media Institute (KMI)

Milton Keynes, United Kingdom

f.wild@open.ac.uk

Received August 6, 2009
Revised September 14, 2009

Pervasive Computing enlarged the traditional computing setting into the human layer of the earth. Different devices in pervasive environments are seamlessly integrated in various ways. Examples are wired and wireless networks, infrared connections, Bluetooth etc. Apart from these local ties, pervasive computing environments are mostly connected with the World Wide Web environment as the simple most important digital information layer. In this paper we propose an approach, a conceptual upper context model and a service oriented framework for exploiting the vast amount of valuable information contained by the World Wide Web for the benefit of context-aware pervasive computing environments. Machine understandability of the web information is the key challenge for our purpose since a typical pervasive setting covers a variety of devices with different characteristics. We base our approach and models on semantic web activities, particularly on Embedded Semantics, which are of crucial importance for bringing structure into the World Wide Web. We first propose a conceptual upper context model, and then based on this model we propose a web harvesting approach and a service oriented application framework. Finally we demonstrate and evaluate our approach and models through examples which are based on specific scenarios, each representing a different perspective, and we partially realize our approach and the framework for the e-Learning domain.

Key words: Pervasive Computing, Ubiquitous Computing, context awareness, Semantic Web, ontologies, Embedded Semantics, Microformats, RDFa, eRDF, e-Learning.

1 Introduction

Pervasive computing [1, 2] environments are composed of stationary and mobile devices which are working collectively for the benefit of the users. These environments are highly dynamic, mobile (i.e. users are mobile, devices are mobile, applications are mobile), highly connected and heterogeneous. This brings many challenges along such as context awareness, resource management etc. Different devices in these environments are seamlessly integrated to each other in various ways. Examples are wired and wireless networks, infrared connections, Bluetooth etc. Apart from these local ties,

This article is based on “Embedded Semantics Empowering Context-Aware Pervasive Computing Environments”, Soylu A., De Causmaecker P., Desmet P., 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 7-9 July 2009, Brisbane, Australia, © 2009 IEEE.

Pervasive computing environments are also mostly connected with the World Wide Web environment as shown in Fig. 1.

Pervasive Computing enlarged the traditional computing setting into the human layer of the earth. World Wide Web, being the simple most important digital information layer, must be integrated over this new enlarged computing setting. It is reasonable to call such enlarged web environment uWeb, Ubiquitous Web. The challenging task of prominent importance is to exploit valuable web information. The Semantic Web [3] is the key enabler of such an approach. It will enable intelligent environments to filter, search, and recommend information from the web environment according to the context of the user. Currently information on web pages is based on natural human language and aimed at humans. Therefore human-beings are able to process information on the Web, to deduce facts, to infer reasons, to associate and link different kind of distributed information, but computers are not.

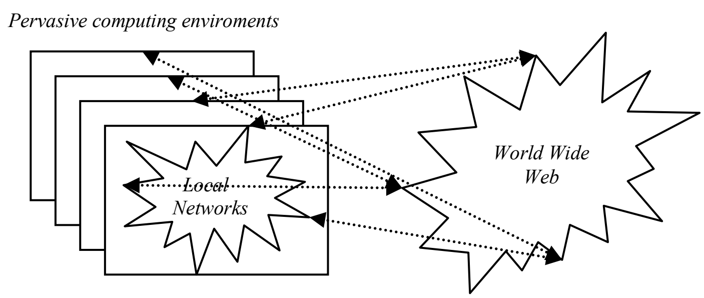

Figure 1. Abstract view of pervasive computing environments and the World Wide Web is depicted.

In this paper we extend and adapt our previous work [4], which was a learning object harvesting proposal based on the semantic web activities (particularly on Embedded Semantics). It allowed pervasive computing systems to exploit web information in an anytime-anywhere fashion. The approach we propose is in twofold: (1) technical and (2) theoretical. From a technical point of view, bringing structure to the web information is the key function of the semantic web activity. Languages such as XML, RDF and OWL (and their derivatives) of the semantic web activity have already been employed for varying challenges of pervasive computing environments such as services [5] (e.g. WSDL, OWL-S etc.), context modelling and representation [6] (e.g. RDF, RDFS, OWL) etc. XML, RDF and OWL can be seen as the layers of a tree where each layer requires different skills and targets different needs [7]. XML constitutes the first layer of this structure. It requires limited skills and solves the less complex problems. It structures the data without implying any meaning. RDF expresses the meaning, but it does not allow describing complex relationships between objects such as cardinality constraints, unions, disjoint classes etc. RDFS extends RDF and provides a limited capability to define simple relationships. OWL is used to define complex relationships and sets of interference rules between data objects by providing more vocabulary. XML, RDF, OWL etc. aim at machine understandability. Duplicate efforts are still needed to create a piece of data both in RDF for machines, and in simple (X)HTML for humans. Other possible problems are consistency, synchronization etc. There are efforts to prevent such problems. These are the main subject of this paper. eRDF [8], RDFa [9] and Microformats [10] are leading technologies in this field allowing embedding semantic data (i.e. in place data annotation) into web pages that can be both machine and human understandable.

Embedded Semantics will enable pervasive computing environments to filter, compose, decompose and present web information while loosely coupling web environments and pervasive computing environments. On the other hand, we consider web information in a broader sense from a theoretical point of view. We first provide a conceptual upper context model and locate information within this conceptual model as one of the primary entities. Although the solution we are proposing is limited in scope according to the proposed conceptual model, existence of such a conceptual model enables the proposed approach to be compliant with the pervasive computing vision in general. Besides by providing a general understanding of Pervasive Computing, the conceptual model provides us a line of research where various challenges exist. The resulting model enables different devices in pervasive environments to use the Web collectively, as human users do, for the benefit of various users among their territories. That means humans are not the only users of the Web anymore; machines harvesting web information, in anywhere-anytime fashion, for the benefit of human users also are.

Before going into details of the challenges, we define some terms within the scope of this paper. During the course of this paper we are going to use the terms “web resources”, “digital objects”, “web information”, “metadata” and “vocabulary”. Web resources refer to a variety of documents, which reside in the web environment, in different formats such as (XHTML, PDF, TXT etc.). Digital objects refer to two kinds of things. The first one is the meaningful, fine grained, structured and commonly published chunks of information residing in the web environment (such as user profiles, events etc.), and the second one is fine grained pieces of information representing common characteristics of web resources. Vocabulary refers to sets of words used to give a semantic structure to commonly used chunks of information (i.e. texts), where metadata refers to sets of words used to describe characteristics of web resources such as author, size etc. Finally, web information refers to both web resources and digital objects. In this paper, we address the following interrelated challenges in particular: (1) to enable a variety of devices in pervasive computing environments to extract and use valuable web information, (2) to free digital objects from presentation by giving them a semantic structure, (3) to enrich web resources (mainly a variety of multimedia documents) by annotating their characteristics, (4) to enhance search and retrieval of commonly published chunks of information e.g. people, events, blog posts etc., (5) to enable applications to mediate between characteristics of web information and context of use in order to provide context adaptive search, retrieval and access of such information.

The rest of the paper is structured as follows; in section 2, we criticize our approach from a context awareness and an adaptivity point of view, and we propose a conceptual upper context model. In section 3, we briefly refer to embedded semantic technologies while we explain and exemplify the details of the approach and the framework in section 4. We realize our approach and framework for the e-Learning domain in section 5. We discuss and evaluate the proposed approach and models in section 6, and we conclude the paper in section 7.

2 Context Awareness

Pervasive computing environments primarily require vast amounts of heterogonous devices in the user environment to collectively act for the benefit of the users without creating any distraction at the users' site. This implies that computing systems need to reach a level of understanding of the settings in which they are being used and of the complex relationships between the entities of these settings. This requirement leads to context awareness. Context awareness enables pervasive computing environments to better serve users by applying available context information where context is any information characterizing the situation of an entity [11, 12]. We consider perception and adaptivity as particularly important aspects of such an approach. Perception enables applications to collect crucial context information such as characteristics, capabilities and requirements of different entities in computing settings while adaptivity enables applications to collectively mediate between different characteristics, capabilities and requirements of these entities. We consider adaptivity as the primary relation between context and computing in the context-aware computing settings while the user is the primary reference point for such an adaptivity.

Pervasive computing environments are highly heterogeneous and complex. They enable computational entities in such environments to collectively act for the benefits of the users by adapting themselves to the context. This requires systematic approaches rather than ad-hoc solutions. Therefore we first provide a conceptual upper context model which will be developed into an upper context ontology where domain specific ontologies can be plugged in. Benefits of such a model are considered in twofold: (1) operational, and (2) conceptual. From an operational point of view, after formalizing such conceptual model and domain specific ontologies for context representation, reasoning and inference mechanisms can be exploited for the sake of adaptivity. Generic knowledge and domain knowledge can be shared across diverse applications and devices for the sake of semantic interoperability and collectivism. On the other hand from a theoretical point of view, it enables us to handle challenges of the pervasive computing vision in a divide and conquer manner. Instead of providing exhaustive solutions for such settings, this conceptual model enables us to narrow down our scope based on some specific challenges according to the whole model and to provide specific solutions which are still compliant with the overall vision and understanding.

2.1 Conceptual Upper Context Model

Since there is no widely accepted upper context model in the Pervasive Computing domain, we propose our own upper conceptual context model which has been previously introduced in [13]. We do not claim completeness of this model, however it is open to possible extensions. We refer to related literature before introducing our proposal. [14] categorizes context based on grouping similar context dimensions into: (1) computing context, (2) user context and (3) physical context. Later [15] extends this categorization with (4) time context. [11] provides a similar context categorization: (1) physical context, (2) social context, and (3) internal context. Inspired by the above categorizations, we propose eight main elements for context aware settings: users, devices, applications, environment, information, time, history and relationships (see Fig. 2 and Fig. 3). We will argue for a layered categorization of context without considering any taxonomical relation: (1) user context (internal, external), (2) device context (hard, soft), (3) application context, (4) environmental context (physical, digital – e.g. network -), (5) information context, (6) time context, (7) historical context, (8) relational context. We split up the environment context category into physical environment and digital environment as shown in Fig. 3. Physical environment refers to nearby physical objects, their identities etc. while digital environment refers to digital entities (i.e. applications and information), their identities and characteristics etc. The concept device can be seen as part of both the digital environment and the physical environment since a device has both a physical existence (e.g. physical location) and a digital existence (e.g. digital location such as an IP address). Information context refers to properties of meaningful information pieces available in different formats (e.g. text, image etc.). It is surprising to see that information has not been considered as an independent entity neither in available context categorizations nor in various context models in the literature (to the best of authors knowledge) [13]. This categorization provides a clear layering for context-aware system development. The proposed categorization constitutes an upper conceptual model which is depicted in Fig. 2 and Fig. 3. Some possible domain specific concepts are also shown (e.g. knowledge and goal.)

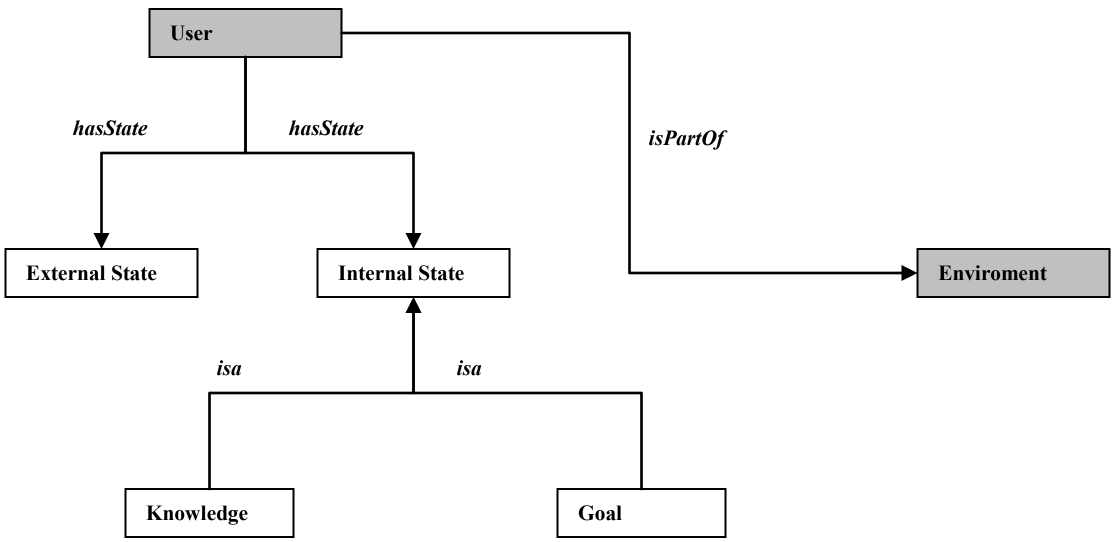

Figure 2. First part of the proposed upper context conceptualization; external state and internal state concepts are shown as a part of user concept where user concept is part of the environment [16].

2.2 Digital Objects and Adaptivity

In this paper our focus is on the information (i.e. on digital objects, see Fig. 3) which are part of the digital environment together with the applications. Each digital object has its own characteristics (i.e. context) such as media type, size etc. Digital objects can be compound, that is, they can be composed of other digital objects.

Considering the adaptivity for the information in the web environment, the Adaptive Web [17] field often focuses on the mediation (i.e. adaptivity) between user models (e.g. knowledge, goal etc.) or profiles (e.g. interests etc.) and the information based on their syntactic characteristics in order to filter, search, and recommend these web resources (not fine grained and not structured, mostly in the form of pages, images etc.). However in an extended computing setting (i.e. Pervasive Computing), context is not limited to user knowledge, and goals etc. anymore (and from another point of view context is not just the location). It includes environment, devices, applications, etc. Hence characteristics, requirements and capabilities of digital objects must be mediated with context of other entities based on semantic descriptions (e.g. output capabilities of a device with media type of a digital object etc.).

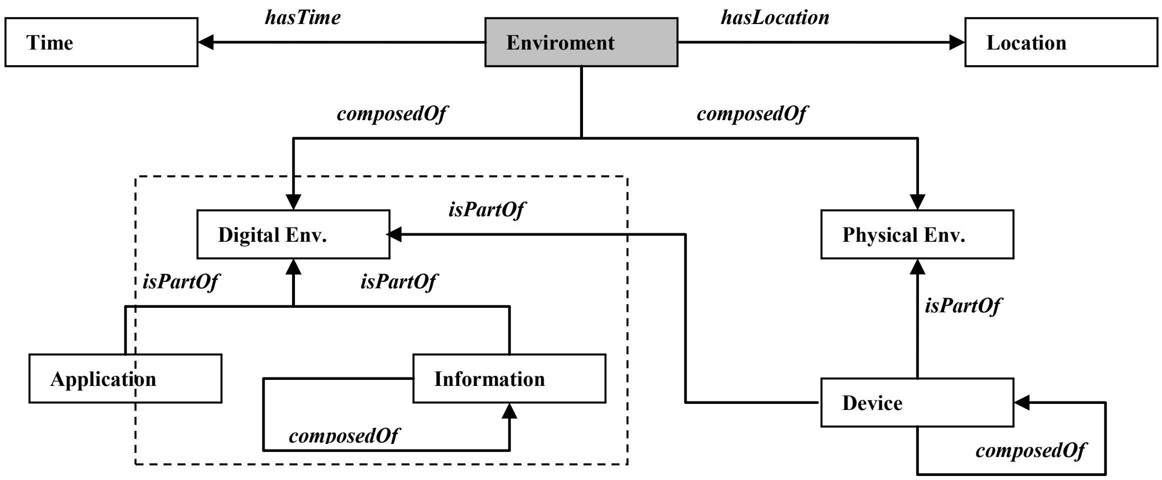

Figure 3. Second part of proposed upper context conceptualization; environment concept is composed of digital environment and physical environment concepts [16].

There are numerous metadata standards to represent basic characteristics of fine grained web resources such as LOM [18] for learning objects, Dublin Core [19] for web resources etc. XML, RDF, etc. have been used as a representation format for these objects, however they do not fully conform to the requirements of our approach since they require higher competence, and explicit data annotation (even duplication). However they are still crucial for our approach as a representation format for harvested information (see section 4).

The branch of Embedded Semantics fits our objectives well. It represents characteristics, requirements and capabilities of digital resources by making use of common metadata standards filling the gap between the web environment and the context-aware pervasive settings.

3 Embedded Semantics

We propose four layers of abstraction for information (including contextual information, see Fig. 4) which is extended from the three layer of abstraction proposed by [20]: (1) storage layer, (2) exchange layer, (3) conceptual layer, and (4) representation layer. The representation layer, particularly Embedded Semantics, constitutes the missing link in the current approaches.

RDFa, eRDF and Microformats are important technologies for embedding semantics into web pages. RDFa is a W3C specification based on expressing structures via attributes of languages of XHTML and HTML. eRDF, Embedded RDF, allows some very important parts of the RDF model to be embedded but does not attempt to extend this to the full RDF model [9]. Microfromats are composed of (X)HTML tags that are used to embed information into web pages which are understandable both by machines and humans while considering the human as the first priority [21].

Representation Layer (e.g. HTML, RDFa, eRDF, Microformats)
Conceptual Layer (e.g. OWL, RDF, UML etc.)
Exchange Layer (e.g. XML, JSON, CVS etc.)
Storage Layer (e.g. tupples)

Fig. 4. Four layers of abstraction for contextual information and information itself: storage layer, exchange layer, conceptual layer, and representation layer [16].

RDFa and eRDF are based on the RDF framework, and they only provide syntax in order to express RDF by (X)HTML attributes. Attributes in an (X)HTML page are not viewable by the users, so RDFa and eRDF use these attributes to embed RDF vocabulary into the (X)HTML pages. The main difference between RDFa and eRDF is that, RDFa reflects the full capability of RDF via completing the missing abilities of (X)HTML with new attributes. eRDF does not introduce any new attributes, and it prefers not to express RDF models which are not expressible by the capabilities of (X)HMTL such as blank nodes, containers, and typed literals. This implies that every eRDF structure is expressible via RDF but not every RDF structure is expressible via eRDF. RDFa introduces new attributes therefore it is not compatible with XHTML 1.0 and breaks the syntax of XHTML. However RDFa is expected to be supported by later versions of XHTML. RDFa and eRDF are based on the framework (i.e. syntax and knowledge representation ontology [22]) provided by RDF where Microformats offer both syntax and a set of fixed vocabulary as it does not rely on RDF or any other framework. Therefore Microformats are domain specific; you cannot express anything with Microformats unless its syntax and vocabulary is defined by its community. This also implies that the extracting procedure is the same for every eRDF and RDFa involving (X)HTML pages where it is different for every document which involves Microformats. However since Microformats do not have any relation with RDF or such framework, they can be directly embedded into an (X)HTML page without any need of both (X)HTML and RDF mark-up writing. eRDF and RDFa's being based on RDF enables users to mix and use different name spaces. Microformats use a flat name space, it is already predefined and you cannot extend or mix it, besides you cannot add new metadata elements since it is a community approach. It is obvious that RDFa and eRDF provides much more flexibility than Microformats, and strong ontology languages were built on top of them (e.g. OWL). Real life applications of either RDFa or eRDF are still rare while there are ample real life examples of Microformats by hobbyists or enterprises like "Yahoo Locals". Microformats do not either aim to be a panacea for all taxonomies, ontologies or does not aim to be infinitely extensible or open ended [23].

While Microformats can encode explicit information to aid machine readability, they do not address implicit knowledge representation, ontological analysis, or logical inference [10]. The simplicity of this approach may be the reason of its success. Its limitations, however, are the reason why it is called 'lower case' Semantic Web. A living community being the key, it is not astonishing that most Microformats express well-established and accepted metadata those normalized by Dublin Core, iCal, vCard, FOAF, or ATOM. For instance, using the hAtom Microformat, a web site can immediately provide a feed through intelligently marked up items. There is no longer a need to publish the same content in other formats like RSS since applications can extract raw ATOM data from the (X)HTML of the web site, again without converting to and from RDF [24].

Fig. 5 contains a comparison between (X)HTML and Microformats. Information of a particular user presented in an ad-hoc manner in the former case, that is how the Web is presently. In the latter case Embedded Semantics impose semantic meaning and structure on the information. Meanings are represented by the vCard vocabulary using class attributes. This is nearly an effortless process and does not require altering current web technology. Use of XML and RDF would provide a surface syntax for structuring, but we would have to duplicate some information in (X)HTML for human reading.

```
<table>
<caption>Person</caption>
<tr>
  <td>Name</td>
  <td>Surname</td>
</tr>
<tr>
  <td>Emel</td>
  <td>Dolma</td>
    </tr>
</table>
```

```
<span class="fn n">
<span class="given-name">
   Emel
</span>
<span class="family-name">
  Dolma
</span>
</span>
```

Figure 5. Comparison between plain (X)HTML and hCard Microformat embedded (X)HTML.

Tab. 1 demonstrates semantic and structural difference between plain (X)HTML and Microformat embedded (X)HTML according to the Fig. 5. Pure (X)HTML requires ad-hoc extraction techniques where embedded semantics provide a standard way of information extraction.

Table 1. Comparison of semantics and structure of (X)HTML and Microformat embedded (X)HTML.

Tag Name	**Content**	**Tag Name**	**Content**
caption	Person	given-name	Emel
td	Name, Surname, Emel, Dolma	family-name	Dolma

4 Web Harvesting

In this section, we start with some short scenarios from different perspectives (e.g. smart spaces, Pervasive Computing, e-Learning) in order to give a better insight of the harvesting framework before going into detail. Bob has an intelligent home with various devices (TV, refrigerator etc.) These are connected to each other and to the internet. These devices are able to identify different users in this home. Furthermore they are able to construct a user profile for each user reflecting user's habits etc. Considering TV related behaviour, assume that there are different TV stations embedding their schedules in their web pages through one of the aforementioned technologies, RDFa, eRDF or Microformats, by using a common semantic (i.e. common vocabulary) for program descriptions and event descriptions (e.g. iCal). This intelligent TV can recommend TV programs for a particular user based on the user profile by harvesting and querying embedded information found in the web pages. We can further extend this scenario to cover pervasive aspects. Consider a refrigerator which is also connected to the Web and which is able to detect food items which it contains (possibly through RFID). When a particular type of food is exhausted, it harvests web pages of different markets and according to the shopping habits and economical considerations of the family it recommends a list of markets. It communicates with the TV for display purposes. TV and refrigerator use RDF for communicating shared information.

The following scenario is inspired from [25] where the author envisions the human layer of the earth as a one vast downloadable, writeable and searchable surface. Based on this example scenario, proposed harvesting framework and the implementation will be briefly described and exemplified. Tom decides to visit a famous city A during his summer vacation. He is really interested in history. He carries his internet connected PDA which has a GPS feature. There is a web community where members of this community tag places which they have visited by using a Google Earth kind of application together with historical information and pictures belonging to these places. This application embeds the coordinates of the places, information about these places and information about available pictures (or documents in general) by using Microformats (e.g. Microformat Geo [26] which represents WGS84 geographical coordinates). Tom is also member of this community, his PDA observes his coordinates during his trip, and it queries pages in the community site to find coordinate matches. When a match is found, this implies that a particular historical or interesting place is around. Hence, Tom's personal agent in his PDA harvests the important web information about this particular place from the community web pages instead of fetching the pages as a whole. Later his PDA represents extracted web information by voice or screen display to Tom according to the result of mediation between characteristics of Tom's PDA, web information and preferences of Tom. Tom's personal agent detects that characteristic of one of the images does not match with the capabilities of Tom's PDA (e.g. size), therefore it only displays an alternative text. Furthermore, Tom might bookmark the place which he is presently visiting to his community page together with some interesting information and pictures of this place by using his PDA's GPS and wireless internet features. More specifically this scenario describes an informal learning situation which employs a community driven digital library (why not uLibrary?) where both learning and community library are context-aware.

4.1 Simplified framework

The complete simple framework which satisfies the above scenarios is depicted in Fig. 6; the first step is structuring meaningful information pieces (i.e. digital objects) in web pages by using RDFa, eRDF or Microformats. It is not possible to model or describe semantics of every kind of information that resides in the Web although RDFa and eRDF allow any custom semantic to be defined. Embedded information cannot really be shared without having a common understanding of semantics behind this particular information (i.e. vocabulary for metadata). Hence, having standard semantics for particular information collections such as vCard, iCal, XFN etc. is crucial. Web applications should be able to choose among these technologies as long as the semantics are standard or known to other related bodies. This is because the technologies for harvesting these embedded semantics, GRDDL and XSLT, are able to harvest semantics represented by Microformats, RDFa and eRDF.

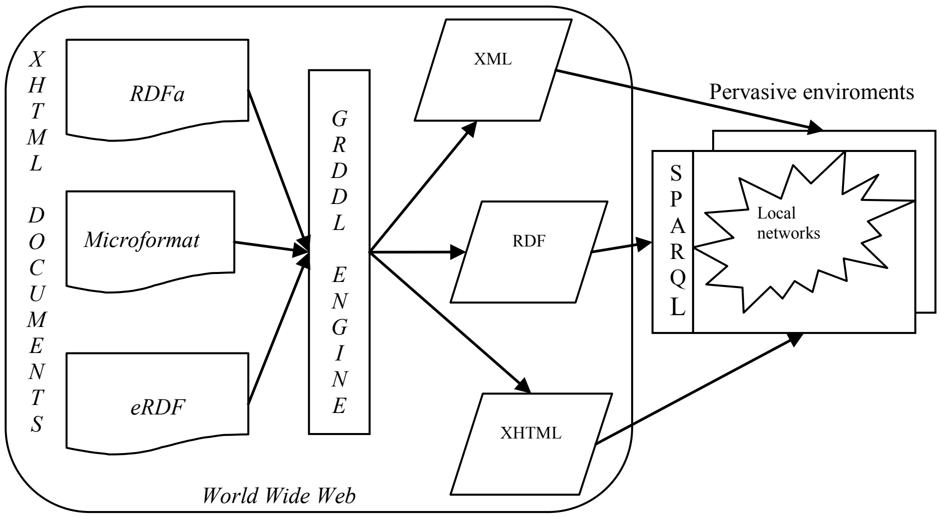

Figure 6. A simple web harvesting framework.

GRDDL [27] is employed in this framework to transform Embedded Semantics in any format such as XML, (X)HTML and RDF. RDF holds particular importance for us since it has strong reasoning tools and applications built top of it, hence harvested data is transformed into RDF for further machine processing (e.g. reasoning). SPARQL [28] allows querying RDF data. Devices harvesting embedded semantics from the web pages can use RDF transformations for querying purposes. We previously noted that pervasive computing environments are heterogonous, and this means that each device in such settings has different hardware configurations (CPU, screen size, memory etc.) and software (available libraries, services etc.) components. We free the information by separating information from presentation. We enable different devices to employ harvested information in different ways (e.g. compose, decompose etc.).

```
<div class=”geo”>Geo:
   <span class=”latitude”>30.386142 </span>,  <span class=”longitude”>-120.092834</span>
</div>
```

Figure 7. XHTML code piece with embedded Microformat.

Example code, given in Fig. 7, demonstrates the Microformat code piece for the Geo. There is no standard vocabulary (i.e. metadata) to represent knowledge about places to the best of the authors knowledge. Even though it does exist, since Microformats is a community approach, it is not possible to come up with a Microformat without community commitment. The case is different for eRDF and RDFa. Semantic information is annotated by using the class attribute of the elements. Class value for div element refers to “geo” which points out that this embedded information is for geographical coordinates where “latitude” and “longitude” are metadata elements referring to the particular coordinate elements. Code piece given in Fig. 8 belongs to the basic XSLT file which transforms above Microformat into RDF format.

```
<xsl:template match="/">
<xsl:for-each select="//*[@class='geo']">
    <rdf:Description>
       <xsl:attribute name='about' />
       <xsl:apply-templates/>
      </rdf:Description>
</xsl:for-each>
</xsl:template>
<xsl:template  match="htm:*[@class='latitude']">
     <geo:lat>  <xsl:value-of select="."/> </geo:lat>
</xsl:template>
<xsl:template match="htm:*[@class='longitude']">
     <geo:long>  <xsl:value-of select="."/>  </geo:long>
</xsl:template>
```

Figure 8. XSLT code piece to transform Geo Microformat to RDF format.

This embedded Microformat results in the RDF code piece depicted in Fig. 9 when it is harvested by Microformat enabled agents by using XSLT code given in Fig. 8 and GRDDL.

```
<rdf:RDF  xmlns:rdf="…" xmlns:geo="…">
  <geo:Point>
     <geo:lat>30.386142 </geo:lat>  <geo:long>-120.092834</geo:long>
  </geo:Point>
</rdf:RDF>
```

Figure 9. RDF representation of Geo Microformat.

Fig. 10 depicts a SPARQL query which queries the RDF code piece in Fig. 9. This query asks for the RDF triples (i.e. object, subject and predicate) where the object is “geo latitude”, and the predicate is 30.386142.

```
SELECT ?object ?subject ?predicate WHERE { ?object <geo:lat> “30.386142”}
```

Figure 10. Example SPARQL query.

4.2 Extended framework

We extend the simplified framework given in section 4.1 and propose an extended framework based on service oriented architecture and agents. The main component of this architecture are: (1) a knowledge base, (2) ontology base, (3) query service, (4) reasoning engine, and (5) harvesting agents as shown in Fig. 11.

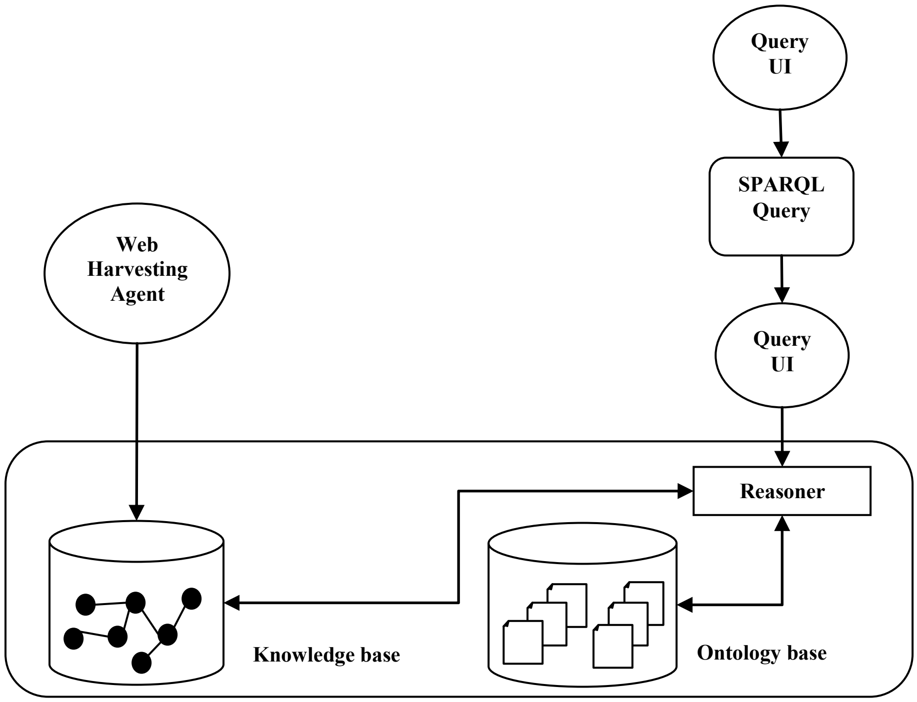

Figure 11. The extended framework demonstrating proposed web harvesting procedure with ontology support.

The ontology base holds generic and domain specific terminological knowledge for representing harvested information in order to support reasoning and inference. The knowledge base holds harvested information in the form of RDF triples where harvesting web information is done by harvesting agents. The reasoning engine provides an inference and rule based reasoning mechanism. The query engine is responsible to respond queries coming from clients. There are several query and response formats which our query engine supports such as SPARQL, RDQL etc. as a query format, and XML, RDF etc. as a response format. Query format and response format is subject to arrangement between client and query engine before submitting the query [29].

Existence of a reasoning engine and an ontology base is the key enhancement with respect to the simplified framework provided in section 4.1. Although in the context of this paper we partially implement the extended framework (the simplified framework is fully implemented) where the reasoner and the ontology base are missing. We can maximize the utility of the semantic web family in such a framework by formalizing domain and generic ontologies. The simplified framework only operates according to simple meanings of information however by extending it to an ontological setting, we are able to use complex relationships between characteristics of this information pieces and other entities within the ontology. A particular example is enabling the query engine not only to retrieve the digital objects which have an exact matching with the user query, but also the ones which have a degree of semantic similarity with the user query. This is done by using reasoning mechanisms such as subsumption (Class A subsumes class B if and only if class A includes a set of instances of class B [5]) , and rule based reasoning.

5 Case Study: Harvesting Learning Objects

In this section the implementation of the proposed harvesting approach and the framework for the e-Learning domain will be introduced in detail, the particular focus is on harvesting learning objects.

The implementation of our proposed framework for harvesting learning objects assumes learning objects as an instance of digital objects as previously defined for the sake of simplicity (although in general learning objects are digital and non-digital artifacts – e.g. images, texts, etc. - that can be used for educational purposes). Learning objects can be part of the courseware ranging in size and complexity from a single graphic to an entire course [30]. Enabling learning objects to be re-usable and sharable by different tools, systems, and applications is the core challenge for e-Learning interoperability. Hereby, metadata plays a key role: there are several bodies which have published standards and specifications to describe metadata information. In the context of this work, LOM (Learning Object Metadata) standardized by IEEE is of prominent interest, though other standards are available and compatible with the approach chosen. Most of these standards consist of many elements with varying degrees of relevance (depending mainly on the application and the usage context). For instance, LOM currently offers more than 71 elements. From this standardized set offered by LOM, we have derived an application profile by mixing in elements from LOM, Dublin Core (DC) and additional custom elements. Here application profiles play a key role, they aim to facilitate the application-oriented implementation of educational metadata specifications by allowing mixing and matching metadata elements in order to meet specific requirements for a particular context [31] [32]. Simply saying, an application profile might be a subset of a standard or can be a mixture of elements from different standards. The application profile forms the basis for a Microformat to embed learning resources into web pages that can later-on be harvested by crawlers. Embedding Microformat structures into XHTML can be realized by hand with simple XHTML mark-up or it can be automated with the help of XSLT transformations over the XML bindings of LOM.

After proposing a light-weight Microformat for learning objects, a web service (based on SOAP and WSDL) has been created which harvests learning objects (bound to Microformat) in XHTML pages and transforms them into RDF format via XSLT or XSLT and GRDDL combined. If the XHTML file references an XSL file providing transformation information on how to translate Microformat bindings of learning objects into RDF then GRDDL is used. Otherwise a predefined XSL transformation is applied to perform the necessary transformation into RDF. Additionally, an SQI target (i.e. SQI service) has been provided which uses above mentioned harvesting mechanism and allows querying learning objects metadata which is harvested into the central storage facility. SQI is an interoperability infrastructure that enables heterogeneous systems to communicate for the purpose of learning object retrieval by using a common query language (in our case SPARQL) [29]. Finally a simple search client has been implemented which employs SQI target. The extended framework has been partially implemented as shown in Fig. 12.

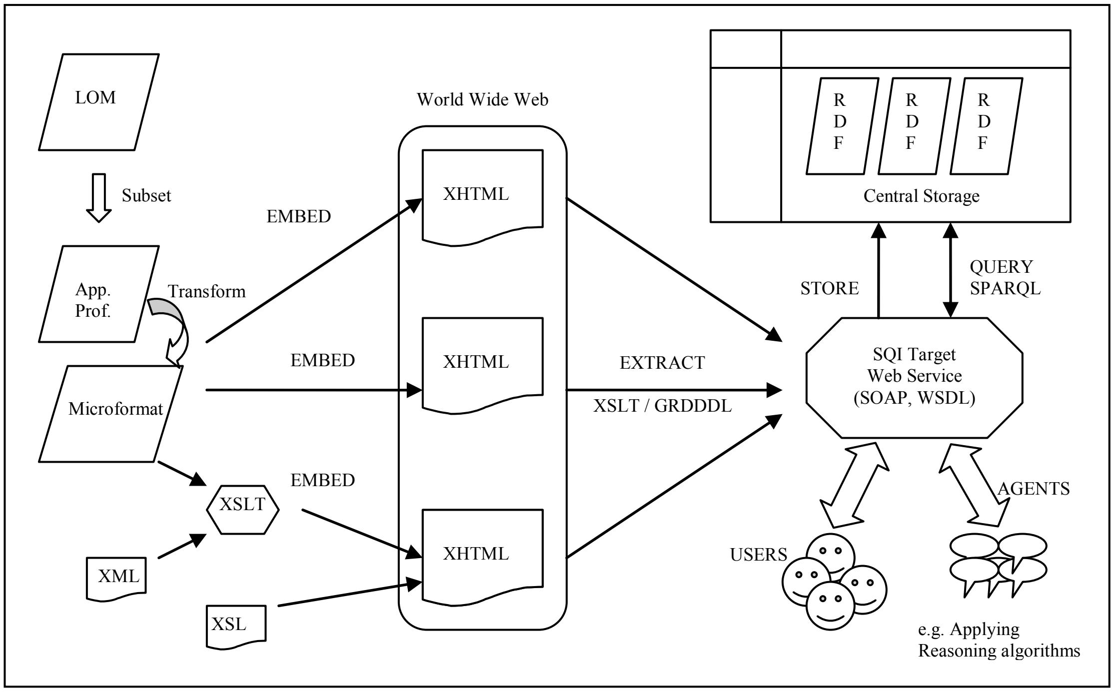

Figure 12. Partially implemented extended framework.

5.1 Implementation

Simplicity and a minimalist approach have been considered as the guiding mantra of the prototypical implementation described below, especially while selecting data types and metadata elements. An e-Learning application profile guideline, which was submitted to the 2006 CEN/ISSS WS/LT workshop [33], has been followed to guide the construction of the application profile. The basic principles of deriving an application profile are as follows [33]: (1) one or more application profile can be selected as base standards, (2) a target system which is compliant with the base standards must also be compliant with the application profile, or a target system compliant with the application profile should be compliant with the base standard, (3) the application profile cannot be less restrictive than the base standards which means application profile can make restrictions of base standards harder but cannot relax them. The procedure roughly consists of identifying the following steps: requirement analysis, selection of data elements, multiplicity requirements, data elements from multiple name spaces, local data elements, obligation of data elements, value spaces, relationship and dependencies, and data type profiling (see [33] for further details). The primary purpose of the proposed application profile is to facilitate re-usability and interoperability of learning objects both by individuals and automated software applications such as agents.

The minimalist approach defines the borders of this application profile. Therefore, it is important to address common needs instead of providing a comprehensive profile since schemas of Microformats are not subject to change often. Elements of the application profile have been derived from IEEE LOM 1482-12-1 2002 and Dublin Core, enriched by two custom elements. The application profile's reference model and XML bindings are based on SCORM 2004 3rd Edition. The selection follows and extends the proposals for application profiles done by Friesen and Campbell [34, 35] and later work of Godby [36] (which is also based on Friesen and Campbell's spreadsheets). A total of 35 application profiles have been investigated and lead to the conclusion that there is a close match between Dublin Core elements and the frequently used IEEE LOM elements. Therefore we followed the idea to map the 15 Dublin Core elements into the IEEE LOM elements if available, which means that we have included only LOM elements having a respective Dublin Core match. If there is no such match, the Dublin Core element is taken as is. A total of 18 elements have been proposed, seven of them adopted from the Dublin Core name space, nine of them adopted from the IEEE LOM name space, and two additional ones that have been derived from IEEE LOM through a custom name space. This is done just in order to simplify the two aggregate elements. Tab. 2 provides an overview on the selected elements. Further details such as data types, multiplicity requirements etc. have been omitted for the sake of brevity.

Table 2. Elements of the proposed Microformat (LO: LOM, DC: Dublin Core, CO: Custom)

Set	Selected Elements
LOM	Title, Language, Description, Keyword, Coverage, Type, End User, Context, Format
DC	Date, Creator, Contributor, Publisher, Rights, Relation, Source
CO	Identifier, Classification

The following step was constructing our own Microformat based on selected metadata elements. Some basic principles of Microformats are as follows [37]: (1) don't repeat yourself (DRY): this principle is against the cases where metadata and presentation is separated from each other. Data should reside only in a single place, and when metadata is updated it has to reflect all layers, (2) visible metadata: there have been several attempts to associate metadata with HTML pages like "meta" element which is invisible to the user but can be detected by computers. However the "meta" element has been used for abusing search engines. Many of them have been forgotten and stayed out dated. Visible metadata principle is a lesson learnt from these previous attempts and making metadata visible prevents these problems, (3) re-use: Microformats are built on widely adopted standards such as XHTML and it uses well established schemas and standards such as RSS, vCard etc., besides it enables Microformats to be compound, that is, Microformat can be used inside another Microformat, (4) a specific problem should be solved and solution should be as simple as possible, (5) and the most famous one, "human first machine second". The POSH [37], Plain Old Semantics, principles are also crucial. It aims separating presentational and semantic behaviour of (X)HTML by using semantic elements and attributes separate from presentational elements and attributes. Furthermore while leaving semantic elements and attributes on XHTML page; it moves all presentational behaviour over CSS, Cascading Style Sheet, files. Some basic principles of POSH is as follows [37]: all the XHTML pages must be validated, that means these pages must follow basic syntax and rules of XHTML like correct nesting, and should use Strict HTML4.0/XHTML1.0. Semantic mark-up and presentational mark-up should not be mixed into each other, instead presentational issues should be handled via CSS. Fewer HTML elements should be used, many presentational elements and attributes can be easily represented with CSS such as "color" attribute or "<center>" attribute. XHTML elements should be used appropriately and should be used for their intended use, for instance “<h1>” should not be used for getting large text, or “<em>”, “<strong>” element should not be used for getting bold text, tables should be used for tabular data not for layout etc. Finally, there are also design patterns, they give Microformat authors a vocabulary for expressing their ideas consistently with what has already been done [37] such as abbr design pattern, date-time design pattern etc.

Our proposed Microformat was quite straight forward in this case, because the base applications profile itself is already simple and flat as there is no aggregate element used in application profile at all. Most of the elements have been directly associated with a ‘class’, i.e. class design pattern, attribute. The ‘creator’, ‘contributor’ and ‘publisher’ elements are used together with ‘vcard’ identifier. This is because the domains associated with these elements are based on ‘vcard’. The abbr design pattern has been used for the ‘date’ and ‘language’ elements. The only elemental microformat, i.e. the ‘rel-tag’ Microformat, is utilized for the ‘keyword’ element. Fig. 13 depicts the extraction of the proposed Microformat.

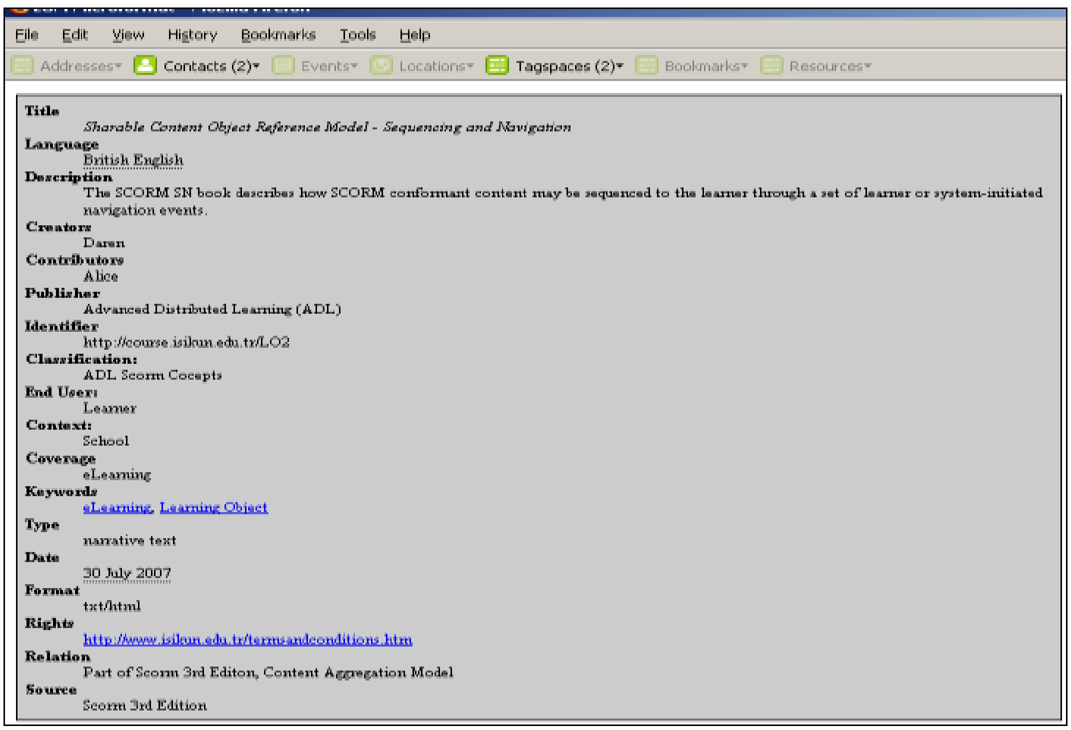

Figure 13. Extracted metadata of an exemplary learning object encoded with our Microformat.

An SQI target has been built and enhanced with GRDDL and SPARQL by using the RAP RDF API [38]. Several sample Microformat-enriched web pages have been generated which are available to the SQI target. Then, a simple search client which is based on AJAX, JavaScript, CSS, and PHP has been set up to employ this SQI target, after its harvesting the sample pages via GRDDL, to query Microformat-encoded semantics via SPARQL. These web pages have been generated for test purposes, however any individual or authoring tool supporting our proposed Microformat is capable of producing such Microformat embedded pages by means of individual authoring or by means of automated transformation of current content as far as a valid match found between their previous (or base) metadata descriptions and our Microformat (this is usually the case if DC or LOM has been employed as base). For instance the metadata for our sample learning objects are created with respect to the XML LOM bindings.

Thereafter, a simple XSL transformation embeds this semantics into the web pages according to our Microformat automatically, and our SQI target harvests this semantic information using GRDDL and transforms it into RDF. Then, the query results are sent to the search client in the form of a LOM XML binding, and finally the search client visualizes the extracted XML via XSL and CSS. In Fig. 14 the search results for the query term 'elearning' are displayed.

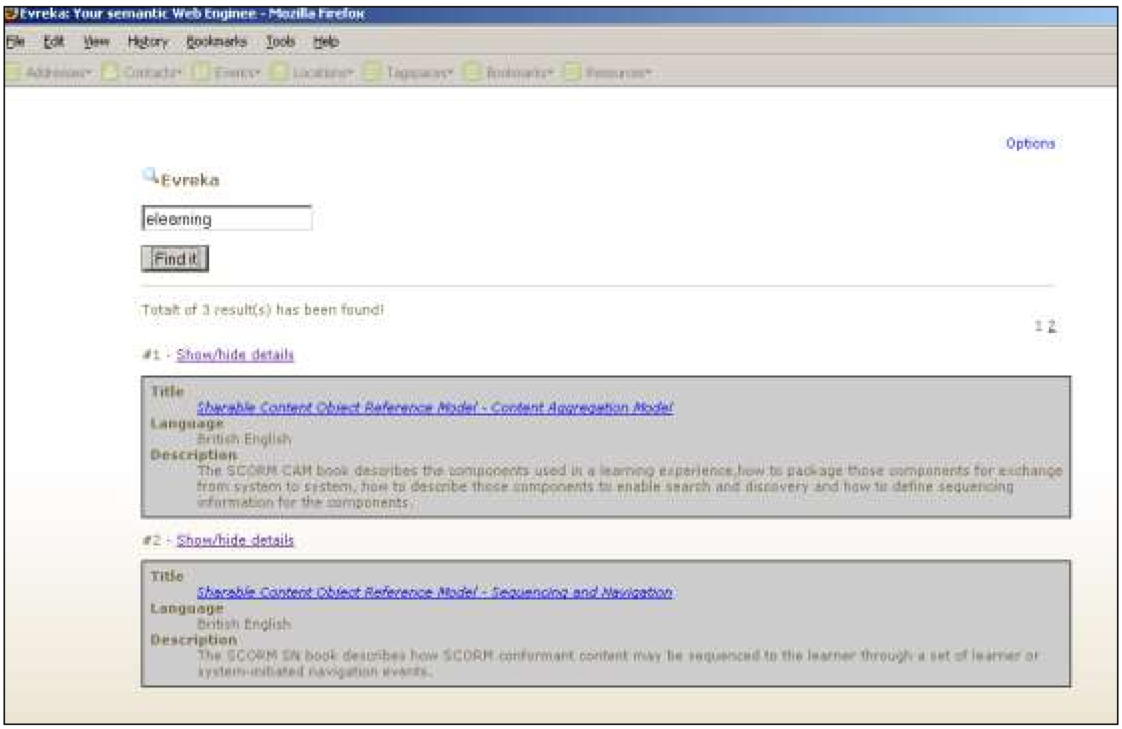

Figure 14. Results for term 'elearning' are visualized with our search client employing the SQI target.

It is important to highlight that the search client serves as a proof of concept, particularly for SPARQL, i.e. the SQI target only employs SPARQL's basic facilities. Once the learning objects are represented with RDF or even empowered ontologically, query languages for RDF like SPARQL will allow the application of complex relationship queries and operations over learning objects in the sense of retrieval, analysis, content construction etc.

Comparing our Microformat-based approach to an ontology-based approach, it has to be mentioned that Microformats have clear shortcomings with respect to RDF. Particularly, Microformats lack strengths like 'open-ended design', 'the ontological expression power', and 'extensibility'. However, a switch from Microformats to RDFa is in principle possible, though at the time of writing not promising due to the widespread acceptance and the ease-of-use of Microformats. The drawbacks of Microformats constitute the main drawbacks of the framework and the sample application.

6 Discussion and Evaluation

In this section we refer to literature in order to evaluate our approach and models according to contemporary research.

Since the approach and framework we proposed are rather structural, we first evaluate them by making a comparison and analysis of existing work in the related literature. We consider two different approaches to merge Web and Pervasive Computing, the first one is from application point of view where researchers use the Web as an application and communication space mapping the real world to the Web as presented in [39] where real life objects have web presences. The second approach is from information point of view, which we follow in this paper, where mapping is from Web to real world. This approach provides more down-to-earth solutions with respect current state of the art and enables us to get benefit from the Semantic Web in pervasive computing environments immediately. There are several scenarios and implementations in the literature which prove the necessity of such an approach although the required attention is not given. In [40] details of an intelligent conference system is given, while in [41] an intelligent meeting system is introduced. These two approaches both require harvesting schedule information from corresponding page of the entities (e.g. users, room, meeting, conference) as a part of their scenarios which demonstrate solutions for different challenges in the field. The former one notes that schedule harvesting is handled by a web parser (we understand a custom and application specific (X)HTML parser), while the latter one refers to harvesting schedule data from OWL documents which are explicitly provided by the corresponding pages. The former approach is not standard but rather application specific, the later one requires information to be duplicated. The straightforward and standardized way of realizing such requirement, based on the model we introduced would be annotating schedule data by using the iCal vocabulary with Embedded Semantics. Another example is [42] which presents a location-aware information service infrastructure for e-Learning. Based on this infrastructure RFID tags are placed on many sight-seeing places where each place is uniquely tagged with an id which is called “ucode”, information about these places are stored in remote databases. Handheld computers obtain “ucode” information from these tags and access remote databases to retrieve the related information. Authors note that in practice enormous numbers of “ucode” tags and remote databases exist. Therefore they also provide large-scale directory servers. Authors also refer to cost of information authoring, they suggest a Wikipedia like community authoring for their system. However they claim that this Wikipedia style mechanism is restricted to digital world and their architecture makes real world places annotated with digital information. The approach we proposed is much more cost efficient with respect to this architecture, we are able to use the World Wide Web as an enormous distributed information source instead of setting up and authoring our own remote information sources. We are able to automatically employ wide community of internet users as authors since otherwise it might be very costly together with management of large-scale information servers. We are able to achieve such location-aware information service which maps web information to real world in a cost effective and simple manner by simply utilizing GPS feature and Embedded Semantics as demonstrated in our example scenario. Finally an ongoing project in MIT Media Lab - which is a novel hardware infrastructure, 6th Sense [43], to integrate physical world and internet - is just another example which our proposal can advance.

Considering the conceptual context model, there are already several works in the literature where a conceptual upper context model is provided [44, 45, 46]. Although there might exist different representations of a conceptual model, extensibility is an important characteristic. Previous models do not capture required entities and relations for our approach such as information as an independent entity, besides our model attempts to be holistic and generic. However for the further development and formalization process, we will inspire from and re-use components of these existing ontologies since there is still a considerable amount of match between these models.

Our approach can be also evaluated from information retrieval point of view to some extent. Data annotation can be seen in twofold within the context of this paper: (1) annotated data describes web documents (e.g. text documents – title, author, size etc.), (2) annotated data is primary data (e.g. user profile – name, age etc.). These two different categories do not make any difference for our approach from a structural point of view since the main purpose is enabling machines to understand and use data. From an information retrieval point of view, for the former category we do not claim any novelty since the case is only metadata versus document modelling based retrieval. However in the latter case, by the using metadata approach, we reduce the problem to a data-centric information retrieval problem [47] which enables us to use techniques available for structured text search [48]. This is because, we employ standard metadata elements (i.e. vocabularies) within an ontology where every item in this metadata standards might either be employed as a data property whose value range is predefined or as an object property where the range of such a property is another class within the ontology. The former case imposes an exact match condition based on syntactic similarity where the latter case can be enhanced by applying a semantic similarity approach based on subsumption tests. An example is the case of the user where she wants to learn about famous places of her current region by using the scenario we introduced in section 5. Our reasoning engine can abstract the coordinates of the users and decide that she is in Paris. Since Paris is subsumed by France, our intelligent query service does not only return famous places in Paris but in whole France by applying an ordering function probably (e.g. places in Paris first). Evaluation of such a semantic similarity based retrieval approach can be done using following instruments although realization of such evaluation is subject to another work: Precision (P) is the fraction of retrieved objects that are relevant where Recall (R) is the fraction of relevant objects that are retrieved [47]. However we can still foresee the advantage of such an approach based on the promising results obtained in similar approaches which are used for semantic discovery of web services in [5, 49].

7 Conclusion and the Future Work

In this article we demonstrated an approach, a conceptual upper context model and a service oriented framework for enabling context-aware pervasive computing environments to exploit vast amount of information residing in the web environment. We stressed that semantic web activity, particularly Embedded Semantics and ontologies, holds crucial importance for such context-aware pervasive environments while standardization (e.g. metadata standards) is also crucial. We followed a systematic approach by first providing a conceptual model, instead of providing ad-hoc solutions.

As a future work, we will formalize our upper contextual model and make use of this upper ontology together with a domain specific ontology for our research on context-aware pervasive learning environments. Harvesting vast amounts of learning resources in the web environment is an important part of its success. Furthermore the extended architecture requires to be fully implemented and evaluated. An immediate work which we would like to realize as a future work is a semantic content output filter (see Fig. 15).

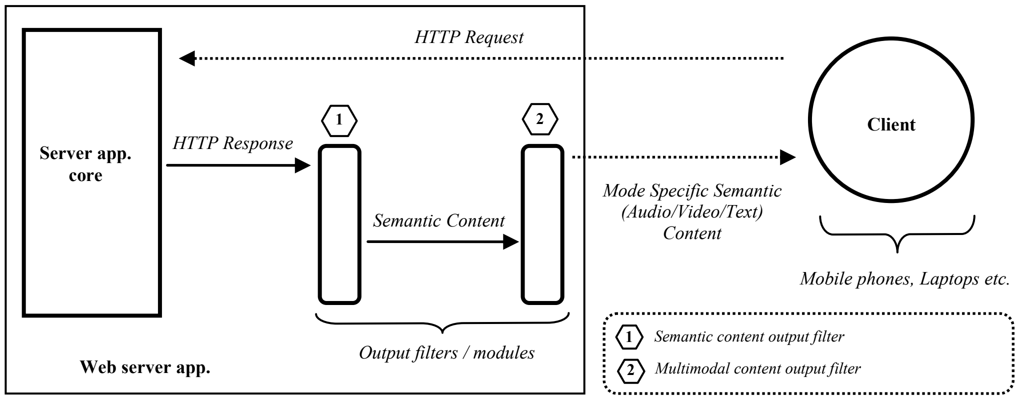

Figure 15. (1) Semantic content output filter, and (2) multimodal content output filter.

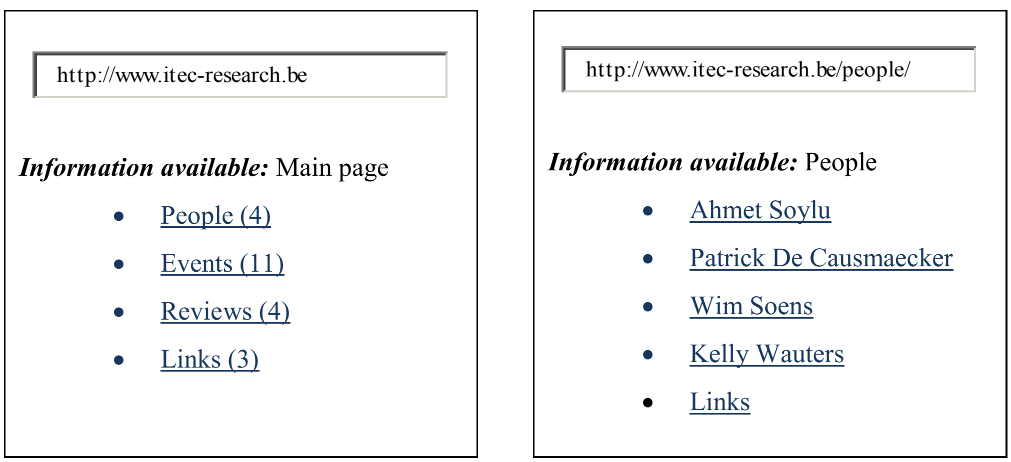

Figure 16. Execution of semantic content output filter is illustrated.

Web server applications need to have generic infrastructures in order to serve and interact with semantic information available. Utility of such an approach is supporting resource-limited devices and bandwidth utilization. Small hand-held devices usually contain limited memory and work with limited internet bandwidth. Hence it is usually not feasible for these devices to fetch and display web pages which involve many multimedia content and are encoded with heavy HTML, and it was not possible to only retrieve related information since it is hard to extract or detect. Such semantic component for web servers enables such devices to interact with only related information and leave out remaining unnecessary content which should reduce the size of information to be transferred drastically. A typical page in “Yahoo Locals” has an average size of 800-900 KB where an embedded Microformat in this page for an event only has an average size of 1 KB. The approach can be further optimized by using more light weight exchange formats such as JSON instead of XML or RDF as demonstrated in [50]. Furthermore extracted semantic information might pass through other filters such as a filter which converts textual information into audio (multimodal content output filter as we name it). Execution of such semantic content output filter is illustrated in Fig. 16., when user accesses a web page through a resource limited device, the output filter is activated. It filters out all the embedded information available through the requested page and delivers the list of semantic information available. Afterwards the user is able to navigate through the semantic information available.

Acknowledgements

This paper is based on research funded by the Industrial Research Fund (IOF) and conducted within the IOF Knowledge platform "Harnessing collective intelligence in order to make e-Learning environments adaptive" (IOF KP/07/006).

References

1. Weiser, M. The Computer for the 21st Century. Scientific American, (1991), 94-98.
2. Satyanarayanan, M. Pervasive Computing: Vision and Challenges. IEEE Personal Communications, (2001), 10-17.
3. Berners-Lee, T., Hendler, J., Lassila, O. The Semantic Web. In Scientific American, (2001).
4. Soylu, A., Kuru, S., Wild, F., Mödrichter F. e-Learning and Microformats: A Learning Object Harvesting Model and a Sample Application. in Proceedings of Mupple'08 Workshop of EC-TEL, (Maastricht, 2008), 57-65.
5. Mokhtar, B. S, Preuveneers, D., Georgantas, N., Issarny, V., Berbers, Y. Easy: Efficient semantic service discovery in Pervasive Computing environments with QoS and Context Support. Journal of System and Software, Elsevier Science, New York, USA, (2008), 785- 808.
6. Strang, T., Linnhoff-Popien, C. A Context Modeling Survey. in Proceedings of Advanced Context Modeling, Reasoning and Management of Ubicomp2004, (Nottingham, 2004).
7. Wilson, M., Matthews, B. The Semantic Web: Prospect and Challenges. in Proceedings of Database and Information Systems. in Proceedings of 7th International Baltic Conferences, (2006), 26-29.
8. Talis, RDF in HTML: Embedded RDF. Available at http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml, (2006).
9. Adida B., Birbeck, M. W3C RDFa Primer. Available at http://www.w3.org/TR/xhtml-rdfa-primer/#id85078, (Retrieved on 2009).
10. Khare, R., Çelik, T. Microformats: A pragmatic path to the Semantic Web. in Proceedings of the International World Wide Web Conference, (2006), 865-866.
11. Han, L., Jyri, S., Ma, J., Yu K. Research on Context-Aware Mobile Computing. in Proceedings of Advanced Information Networking Applications – Workshops, (2008), 24-30.
12. Dey, A. K., Abowd, G. D. Towards a Better Understanding of context and Context-awareness. Technical Report GIT-GVU-99-22, Georgia Institute of Technology, College of Computing, (1999).
13. Soylu, A., De Causmaecker, P., Desmet, P. Context and Adaptivity in Context-Aware Pervasive Computing Environments. in Proceedings of 2009 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, (Brisbane, 2009), 94-101.
14. Schilit, B., Adams, N., Want, R. Context Aware Computing Applications. in Proceedings of Mobile Computing Systems and Applications. (1994), 85-90.

15. Chen, G., Kotz, D. A Survey of Context-Aware Mobile Computing Research. Technical Report, TR2000-381, (Dartmouth, 2000).
16. Soylu, A., De Causmaecker, P., Desmet, P. Context and Adaptivity in Pervasive Computing Environments: Links with Software Engineering and Ontological Engineering. Journal of Software, Vol. 4, No. 9, (2009), 992-1013.
17. Brusilovsky, P., Kobsa, A., Nejdl W. (Ed.) The Adaptive Web. in Lecture Notes in Computer Science, Springer-Verlag, (2007).
18. IEEE Standard for Learning Technology—Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata, IEEE Computer Society, IEEE Std. 1484.12.3 – 2005, (2005).
19. Dublin Core Metadata Element Set, Version 1.1, Available at http://dublincore.org/documents/dces/, (Retrieved on 2008).
20. Reichle, R., Wagner, M., Khan, M. U., Geihs, K., Lorenzo, L., Valla, M., Fra, C., Paspallis, N., Papadopoulos, G. A. A comprehensive context modeling framework for Pervasive Computing systems. in Proceedings of 8th IFIP International Conference on Distributed Applications and Interoperable Systems, (Oslo, 2008).
21. Allsopp, J. Microformats: Empowering Your Markup for Web 2.0. FriendsofED, Berkeley, (2007).
22. Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O. Ontological Engineering. Springer-Verlag, (2003).
23. Simpson, J. Microformats vs. RDF: How Microformats relate to the Semantic Web. Available at http://www.semanticfocus.com/blog/entry/title/microformats-vs-rdf-howmicroformats-relate-to-the-semantic-web, (2007).
24. Graf, A. RDFa vs. Microformats. DERI Technical Report, (2007).
25. Spohrer, J. C. Information in Places. IBM Systems Journal, Vol. 38, No. 4, (1999).
26. Microfromat Geo: Draft Specification. Available at http://microformats.org/wiki/geo, (Retrieved on 2009).
27. Connolly, D. W3C Gleaning Resource Descriptions from Dialects of Languages. Available at http://www.w3.org/2004/01/rdxh/spec, (2004).
28. Prud'hommeaux, E., Seaborne, E. SPARQL Query Language for RDF. Available at http://www.w3.org/TR/rdf-sparql-query/, (2008).
29. Simon, B., Massart, D., Assche, V. F., Ternier, S., Duval, E., Branter, S., Olmedilla, D., Miklos, Z. A Simple Query Interface for Interoperable Learning Resources. in Proceedings of 1st Workshop on Interoperability of Web-based Educational Systems, (Chiba, 2005).
30. Collier, G. e-Learning Application Infrastructure. Whitepaper, Sun Microsystems (2002).
31. Sampson, D. The evaluation of Educational Metadata: From Standards to Educational Metadata. in Proceedings of IEEE International Conference on Advanced Learning Technologies ICALT'04, (2004).
32. Duval, E., Smith, N., Coillie, M. Application Profiles for Learning. in Proceedings of IEEE International Conference on Advanced Learning Technologies ICALT'06, (2006).
33. CWA: Guidelines and support for building application profiles in e-learning (CEN Workshop Agreement 15555:2006E), European Committee for Standardization, (2006).
34. Friesen, N. Survey of LOM Implementations. CanCore, (2003).
35. Campbell, M.L. UK LOM Core Update. CETIS Technical Report, (2003).
36. Godby, C. J. What do Application Profiles Reveal about the Learning Object Metadata Standard?, ARIADNE, (2004).
37. Microformats, http://microformats.org/wiki/, (Retrieved on 2009)
38. Melnik, S. RAP: RDF API for PHP. SourceForge, Available at http://sourceforge.net/projects/rdfapi-php, (August, 2009).

39. Debaty, P., Goddi, P.,Vorbau, A. Integrating the physical world with the web to enable context-enhnaced mobile services. Mobile Networks Applications, Vol. 10, No. 4, (2005), 385-94.
40. Chen, H., Finin, T., Joshi, A., Kagal, L. Intelligent agents meet the Semantic Web in smart spaces. IEEE Internet Computing, IEEE Computer Society, (2004).
41. Nicklas, D., Grossmann, M., Minguez, J., Wieland M. Adding High-level Reasoning to Efficient Low-level Context Management: a Hybrid Approach. in Proceedings of Percom'08, (2008), 447-452.
42. Sakamura, K., Koshizuka, N. Ubiqutious computing technologies for ubiquitous learning. in Proceedings of International Workshop on Wireless and Mobile Technologies on Education, (Japan, 2005).
43. Mistry, P., Maes, P., Chang, L. WUW – Wear your world – A wearable gestural interface. CHI'09, (Boston, 2009).
44. Preuveneers, D., Bergh, D. V. J., Wagelaar, D., Georges, A., Rigole, P., Clerkcx, T., Berbers, Y., Coninx, K., Jonckers, V., Bosschere, D. K. Towards and Extensible Context Ontology for Ambient Intelligence. Lecture Notes in Computer Science, Springer Verlag, (2004), 148-159.
45. Chen, H., Finin, T., Joshi, A. The SOUPA Ontology for Pervasive Computing. Ontologies for Agents: Theory and Experiences. Birkhauser, (2005), 233–258.
46. Gu, T., Pung, H. K., Zing, D. Q. A Service Oriented Middleware for Building Context-aware Services. Journal of Network and Computer Applications (JNCA), Vol. 28. No. 1., (2005), 1–18.
47. Manning, C. D., Raghavan P., Schütze, H. Introduction to Information Retrieval. Cambridge University Press, (2008).
48 Trotman, A. Searching structured documents. Information Processing & Management, (2003).
49. Broens, T., Pokraev, S., Sinderen, M. V., Koolwaaij, J., Costa, P. D. Context-Aware, Ontology-Based Service Discovery. in Proceedings of Second European Symposium on Ambient Intelligence (EUSAI2004), (Eindhoven, 2004).
50. Erfianto, B., Mahmood, A. K., Rahman, A. S. A. Modeling context and exchange format for context-aware computing. in Proceedings of The Student Conference on Research and Development, (Malaysia, 2007).