Towards a Semantic Enterprise Information Portal

Emanuele Della Valle and **Maurizio Brioschi**
CEFRIEL - Politecnico of Milano
Via Fucini, 2 - 20133 Milano - Italy
{dellava, brioschi}@cefriel.it

Abstract

Knowing what you know is becoming a real problem for many enterprises that have computerized most of their information and processes with a large number of ad-hoc, isolated projects. In fact, this way they lost the integrated view of their information. Their intranets are full of shared information, their extranet support a flow of data both with suppliers and customers, but the task of finding information for decision taking is every day harder. This is why, some years ago Enterprise Information Portals (EIPs) came into the limelight for their ability to give a unique and structured view of the available resources, but still EIPs are not the final solution. We believe that good opportunities for improvement, in this field, could come from Semantic Web technologies. Soon some enterprises will start integrating their information resources and services using Semantic Web technologies in isolated *corporate semantic webs*, thus they will need to update their EIPs to *semantic EIPs*.

Here we present an approach towards a semantic EIP, which is an EIP but can, in a way, "understand" the meaning of the semantic descriptions of the resources available on the corporate semantic web. Therefore, it can help in discovering resources by matching user request to resource description and can offer an homogeneous navigation experience among heterogeneous resources distributed over a corporate semantic web.

1 Introduction

As intranet solutions have increased in popularity, they have been populated with: services such as web access to databases, newsletters and forums, and shares full of documents, forms, calendars of events, news and link collections. As a consequence, most enterprises have ended up with a huge set of repositories of structured, semi-structured and unstructured information distributed over the intranet.

Such amount of information is normally *comprehensible to accustomed users* that know exactly where to find it, how to get it and, when necessary, to update it. But, occasional and novel users would have an hard time in getting to what they are looking for, because they might not even know what information is available. So "Knowing what you know" is increasingly a real problem in many enterprises.

Summing up a comprehensive solution to this problem should provide at least an answer to the following questions:

- What information do we have?
- Where is it?
- How did it get there?
- How do I get it?
- How can I add more?
- What does it mean?

In the last years portals have gathered lot of attention among enterprises interested in addressing these questions. In particular many vendors[1] have proposed portal solutions specific for enterprises called Enterprise Information Portals (EIPs). EIPs are web sites, thus they match zero installation requirement, that offer, as general-purpose portals do, a unique and structured access to heterogeneous information and web based service. Differently they focus more on editorial interfaces, making it easier for an enterprise to keep the portal up-to-date.

However EIPs cannot be considered a final solution, because they do help people in managing the information, but they still require a huge amount of manual work. For instance, consider the result set of a web search: how many retrieved pages does a user normally have to read through before getting what he/she is looking for?

So, we believe that using state-of-the-art web technologies will not be sufficient in the immediate future, since the lack of formal semantics will make it extremely difficult to make the best use (either manually or automatically) of the massive amount of stored information and available services.

In the following we will introduce what we mean by ontology-oriented metadata-based solutions (section 2) and our concept of semantic EIP (section 3). Section 4 describes a prototype we have developed as an early proof of concept. Before concluding, in section 5, we gives a short survey of related works.

[1] Hummingbird, IBM, BEA, Oracle and Sybase

2 An ontology-oriented metadata-based solution

Metadata-based solutions has already proved to be able to address, at least in part, all the questions but the last. In fact, they provide enough *machine-processable* information for automating most information retrieval tasks, but, in a pure metadata based solution, the meaning associated to the metadata is not machine-processable. So a machine can process this metadata but it cannot "reason" upon it. Going back to the web search example, this mean that a user of a metadata-based search engine might got a smaller result set, containing more relevant resources, but he/she still has to examine them in order to understand if they are pertinent.

A good deal of help can come from defining metadata using ontologies. In fact, ontologies, being explicit (hence formal) conceptualisations of a shared understanding of a domain [Gruber, 1993] can be used to make metadata machine processable. So, if the meaning of each metadata was defined using an ontology, a machine could, in a way, "understand" it and reason upon it.

However, if a single enterprise had chosen some years ago to build up such an ontology-oriented metadata-based solution, from scratch and on its own, it would have ended up in a "disaster" because no standard solution was available. It was the time, instead, for academics to experiment with such ideas. Ontobroker [Fensel *et al.*, 1999] and SHOE [Heflin and Hendler, 2000] are successful examples of such pre-semantic web applications.

To the contrary, today metadata-based ontology-oriented solutions are becoming feasible thanks to the ongoing Semantic Web researches that are leading the standardisation process of the related technologies. So far, the W3C has coordinated many activities that have already supplied a framework for describing web resources with metadata using Resource Description Framework (RDF) [RDF, 1999]. And the Web Ontology Working Group should soon provide with the final specifications of Ontology Web Language (OWL) [Dean *et al.*, March 2003] a comprehensive ontology vocabulary, complementary to the RDF Vocabulary (RDFS) [RDF, 2002].

Therefore, soon enterprises would be able to build "corporate Semantic Web" represented by services and documents annotated with metadata defined by a corporate ontology[2] (obtained merging both domain independent and domain dependent ontologies). Thus they will need an EIP able to make lever on this virtual knowledge space. Such as ontology-oriented metadata-based EIP, or simpler *Semantic EIP*, will process the metadata providing an access to this huge repository of information distributed over the intranet, making it a materialized *corporate memory*.

3 The concept

The innovative idea, first proposed by [Maedche *et al.*, 2001], is straightforward: can we use metadata defined by ontologies to support the construction of portals? And if so, does it help?

[2]Corporate ontology design issues are still not fully addressed, however we believe approaches such as [Sure, 2002] would soon provide a comprehensive solution.

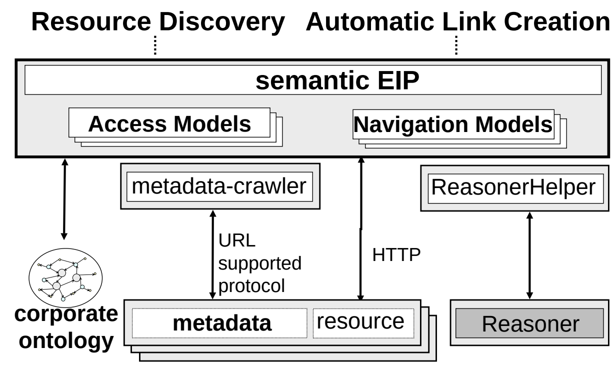

Figure 1: the logical architecture of a semantic EIP built using the proposed approach

Even if it might appear as a radical new departure actually it is not. On the contrary it is the bringing together of existing and well understood technologies:

- *Web Frameworks*, such as Struts, Jetspeed, Tapestry, WebWork and Cocoon, that, following Model-View-Controller design pattern, propose to separate data, business logic, and presentation.
- *Hypertext Architectures* and, in particular, the *WWW conceptual models* such as WebML [Ceri *et al.*, 2000], W2000 [Baresi *et al.*, Jan 2000], HDM [Garzotto *et al.*, 1993], Araneus [Mecca *et al.*, 1998], WSDM [De Troyer and Leune, 1998] and OOHDM [Schwabe *et al.*, 1996], that are proposals for the conceptual specification and automatic implementation of Web sites.
- *Ontologies*, to model the domain information space, the navigation, the access, the presentation and possibly even the operation offered by a portal.
- *Metadata*, to make resource descriptions available to machine in a processable way.

3.1 The approach

So far we have identified the key technologies, but we still have to combine them to get the best out of each. Web Frameworks are good at controlling presentation. WWW conceptual model indicates clearly what to model and how to exploit the resulting models. Ontologies are good formal models. Last but not least, metadata (especially if defined by ontologies) enable distributed approach to information and service management. The logical architecture of a semantic EIP built following our approach is presented in figure 1.

Modeling

Concerning modeling, we decided to follow an approach similar to those adopted in WWW conceptual modeling, but we prefer ontologies (written in OWL) to extended entity-relationship models (E-R). The various WWW conceptual models available in literature showed that is possible, and even convenient, to model separately at least: the domain information space, the navigation and the access.

The *domain information model* (in this case the corporate ontology) is a shared understanding of the information in the corporate memory (hence a unique model) that doesn't change, or changes slowly, over the time. For instance, in modeling CEFRIEL corporate ontology we assert that: `CEFRIEL` is an `organisation`; organisations are divided into `units`; `people works for` a unit `on` one or more `projects`; people can be divided into `researchers` and `consultants`; projects can be split in `researches` and `consultancies`; in particular a researcher `investigates` in at least a research project and a consultant `advises` in at least a consultancy; an employee can be both a researcher and a consultant at the same time; and so on.

Having chosen to adopt ontologies (e.g. OWL) in modeling the domain information, we have already got a terminological layer that is suitable for *direct* use as a terminological interface for a search facility. However, such terminology is appropriate to express "What-is-a" style queries over the corporate ontology, but it can not capture any domain (or user) specific way either to navigate through or to access to a set of documents augmented with corporate-ontology-defined metadata. Thus, we provide such a facility by requiring both navigation and access models.

The *navigation models* represent the heterogeneous paths the EIP users can adopt in traversing the corporate memory. They are not necessarily shared among users, but they are jointly employed by homogeneous categories of users. For instance, taking CEFRIEL corporate ontology as an example, researchers usually share a research-project-centric vision, so knowing each other's competency is more important than knowing which unit another researcher belongs to. Thus if the user is a researcher, then relationships between researchers and their competency should be stressed, while those between researchers and units should be left in background. On the contrary, administration staff have a clear vision of the organization chart, but don't care too much about ongoing projects. Thus the navigational model for them should emphasize relationships among CEFRIEL, its units and the people who work for them.

Finally, the *access models* represent collections of resources not strictly homogeneous, highly variable and sometimes even related to a specific user, a sort of *views*. For instance, we could offer "recently added" (the collection of all the resources added or updated recently) "most visited" or "last visited" (if we monitored the interaction of the users with the portal) and so on.

The most important aspect of domain information model is that it is unique regardless the way it is used. In particular, in our vision it is completely decoupled from the semantic EIP design. Therefore the semantic EIP cannot assume any "a priori" agreement except the use of a common set of primitives (e.g. OWL).

However, if we want to give access to the corporate memory using a semantic EIP we need to define at least some upper terminology, known by the semantic EIP, that can be employed in defining both the navigation and the access model. In WWW conceptual modeling we recognize a similar approach when they use E-R to model the domain information space, while they provide ad-hoc primitives to model navigation (entity, component, node, slot, structural link, semantic link, etc.) and access (collection, collection center, etc.).

Differently from WWW conceptual model approach we don't require to model information using a specific set of primitives, instead we believe it is possible to make lever on OWL mapping primitives. Thus we suggest to build the navigation models by *mapping* the corporate ontology terminology to the navigation upper terminology. The same approach could prove to be useful also in mapping corporate ontology terminology to the access upper terminology, but the high variability and user specificity might require more often to explicitly draw new relationships between resources in the corporate memory and sometimes also to add ad-hoc resources (e.g. centers of collection).

Presentation

We choose not to model presentation explicitly, because we recognize that most of the success of a web application depends on its presentation. In fact, modeling in details such a critical task might prove too complex, in particular because good graphic designer are not supposed to be good modelers and vice versa. However we are not suggesting to code each page from scratch, but to write templates of pages in a model view controller approach. This way we aspect the same advantages, in term of visual coherence and accessibility, as modeling but at a more affordable effort.

Centralised vs. distributed

Moreover, an ontology-oriented metadata-based solution enables a less centralized approach, thus our semantic EIP not only supports a centralized solution, but also a distributed environment where autonomous entities maintain heterogeneous shared resources, describing them with metadata defined by the corporate ontology.

For instance, in an enterprise, where a corporate semantic web is in place and a corporate ontology is shared among departments, the human resource department might offer web access to its information system augmented with a RDF description of each available service, while the R&D department might share a WebDAV folder containing some technical documentation and establish a set of relationships between those, who are authors of the documents, and their references on the human resource information system and so on.

However, a distributed approach is not mandatory, so an enterprise can freely opt for a more controlled environment, providing a set of specific editorial tools for a traditional team of knowledge workers.

3.2 Using ontologies at authoring time

At authoring time ontologies, in particular the corporate ontology, can be exploited in supporting the editorial task. It has already been shown (e.g. in Protégé 2000 [Eriksson *et al.*, 1999]) that they can be employed in automating part of process for creating editorial interfaces. But we believe most of the benefits of using ontologies should come from reducing the effort required to augment resources with metadata. In the authoring environment we envision, authors are asked only what is strictly necessary, while the rest is inferred.

For instance, the metadata a project team uses for describing the project result contains also information regarding the skills of the team members. These skills could be easily used for automatically pre-populating a skill management application, so that the authors were required to confirm what was inferred instead of filling in a tedious job description.

3.3 Using ontologies at browsing time

Web users interact with the Web in many ways, but two patterns are commonly recognized: searching and navigation. Users do search when they know exactly *what* they are looking for, hence they are able to express their requests with sufficient precision. Differently users do navigate when they don't know what they are looking for but they understand it as they browse. Moreover, users do navigate when they know *where* a document lies and they do search when they don't. A semantic EIP should exploit metadata and ontologies in order to improve both interaction patterns. In particular we want to improve searching by resource discovering and navigation by automatic link creation.

Resource discovery

Today, a good intranet search engine can only lead the user to a set of information resources, because of the lack of explicit semantic associated to them, it is unable to "analyse" them. On the other hand, once an enterprise has got a corporate semantic web, search won't be exclusively based on full text search, but it could make lever on semantics, so it could "analyse" the resources finding those that match the user request. Thus it is no more a matter of searching but it becomes a matter of discovery by matching. For instance, a user might require: a paper about security issues for mobile terminals that use Java as language. In the corporate semantic web there could be a technical report about digital signature for cel-phone that use J2ME. There's no syntactic matching between them, but in the corporate ontologies someone could have asserted that digital signature is a security issue, J2ME is a particular edition of Java and a cel-phone is a mobile terminal; so there's a semantic matching.

Automatic link creation

On the other hand, when a user has retrieved a resource (maybe using the resource discovery feature), he/she needs help in navigating to other related resource. So our idea is to insert the resource in a *navigation panel* (see figure 2) that contains automatically generated links to the related resources. In particular in our vision there are, at least three different kinds of links the semantic EIP can generate:

- *access point links* , that, contextualising the retrieved resource in the access model, render both global and contextual access point to the retrieved resource; a sort of views that guide the user in accessing the information. Where global access point are collection centers and contextual access point are link to one or more resources in the same collection.
- *categorized links*, that, contextualising the retrieved resource in the navigation model, render a set of boxes populated with links that are the result of a simple property-based query (e.g. all the resources related via a given property) over the metadata describing the retrieved resource.

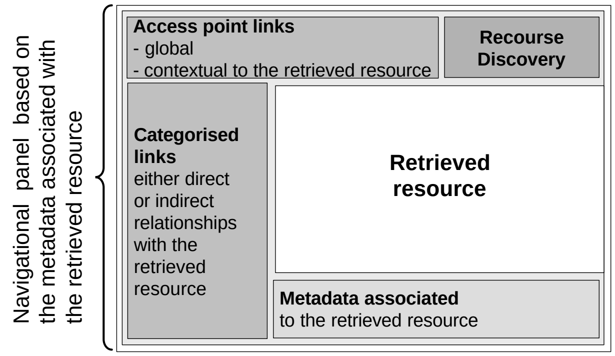

Figure 2: a schematic layout of the proposed navigation panel containing the retrieved resource, three different categories of links and an entry point for the resource discovery feature.

- *metadata links*, that should provide an intuitive navigation from and to the retrieved resource following the RDF properties used to describe it.

4 An early proof of concept

In order to proof this concept, we have built a first prototype of a semantic EIP following the presented approach (an online demo is available at `http://seip.cefriel.it`). We choose not to address authoring time issues but to concentrate instead on browsing time. In particular, we focus to automatic link creation, leaving resource discovery for a following step. Specifically, we developed a servlet-based application that uses Velocity for implementing the model-view-controller pattern and RACER [Haarslev and Moller, 2001] as reasoner. We make it "understand"[3] RDF, RDFS and property characteristics defined[4] in OWL.

Moreover we assume that our prototype "knows" two simple ontologies whose terms describe both the navigation and the access of a generic portal, hence in a domain independent way. We got inspired either by common terms used in WWW conceptual modeling and in HTML 4.1 link types[5]. We kept them explicitly as simple as possible, but still rich enough to be useful in proofing the concept. We understand that, this way, we can show only elementary mapping between a rich domain information model and such simple navigation and access models, but we will eventually develop more comprehensive navigation and access models. What we introduce in the following is only a first draft of the upper terminology we have presented in section 3.1.

4.1 Defined ontologies

On the one hand, the *navigation ontology* defines three properties:

[3]We have extended Jracer 1.7 (see `http://www.fh-wedel.de/~mo/racer/`) to improve support to RDFS and to support a small subset of OWL.

[4]owl:inverseOf, owl:TransitiveProperty, owl:SymmetricProperty.

[5]See `http://www.w3.org/TR/html4/types.html\#type-links`

- `contains`: a transitive property to express containment (e.g. a museum contains art)
- `contained`: inverse property of contains
- `related`: a symmetric property to express a relevant connection

On the other hand, the *access ontology* defines a class and four properties:

- `Home`: a class whose instances are collection centers
- `next`: a transitive property to express precedency in a connected series
- `prev`: inverse property of next
- `down`: a transitive property to express dependency in a hierarchy
- `up`: inverse property of down

4.2 Metadata links

The prototype, "understanding" RDF and RDFS, can process the metadata that describe the retrieved resource, generating links according to the following schema:

```
<resource label>[<list of labels of types>]
<propery label>
<resource label>[<list of labels of types>] or literal value
```

All the words between angle brackets are links that retrieve the resource with the corresponding label.

For instance, if CEFRIEL home page is the retrieved resource our prototype having "understood" the metadata associated with the page should generate the two following set of links:

```
CEFRIEL[Organisation] has_unit eTECH[Unit]
Brioschi[HeadOfUnit,Person] works_for CEFRIEL[Organisation]
```

The former states that CEFRIEL, which is an organisation, has got eTECH as unit and the later that Brioschi, which is a person and a head of unit, works for CEFRIEL.

4.3 Categorised links

The propotype has got 3 boxes containing categorized links:

- the *contains* box, that shows links to resources conceptually "contained" in the retrieved one. We chose to use "contained" in a relaxed way; therefore in this box can appear links for different reasons:
  - if the retrieved resource is a `rdfs:Class`, then the box is populated with links to all its individuals and all its subclasses,
  - if the retrieved resource is related to any other via `contains`, then the box is populated with links to them
- the *contained* box, that shows links to resources that "contains" the retrieved one. Even in this case, we chose to interpret "contained" in a relaxed way including both `rdfs:subclassOf` hierarchies and user defined (via `contains`) hierarchies. So the box is populated with links either to the superclasses or to the resources related to the retrieved one via `contained`.
- the *related* box, that shows links to resources that are associated to the retrieved resource via a `related` property.

As we explain in 3.1 we don't want to oblige anyone to use directly the terms defined in the navigation and access ontology, we want instead to make lever on OWL mapping primitives. Thus, the users should not connect resources present in the corporate memory directly with `contains`, `contained`, `related`. They should otherwise map properties, which already exists into the corporate ontology, to those. In particular we choose to map properties using `rdfs:subpropertyOf`. This way the reasoner can easily compute sub-property closure and "understand" that two resources are related (e.g. via `contains`) not only when it is explicitly stated, but also when it is entailed.

For instance, CEFRIEL has got 9 Units and in the corporate ontology `has_unit` is the property used to relate CEFRIEL to its Unit, so if a group of users normally interpret the `has_unit` as a containment relationship, they can put in the "navigation model":

```
has_unit rdfs:subpropertyOf contains .
```

In this way, when CEFRIEL home page is retrieved, links to all the 9 unit of CEFRIEL are placed in the "contains" box.

4.4 Access point links

Finally the prototype has got a global navigational bar and a contextual navigational bar configurable through the access model. The global navigation bar is populated with links to resources of type `Home`, while for the contextual navigation we use an approach similar to the one illustrated for categorised links (see 4.3). So our prototype populates the boxes labeled "prev", "next", "up" and "contextual navigation" with links to resources, that are associated to the retrieved resource, respectively via a `prev`, `next`, `up` and `down` property.

As we stated before with access models we represents collections of resources not strictly homogeneous, highly variable and sometimes even related to a specific user. In order to build such a model the resources, part of a collection, should be related using the terminology of the access ontology. Thus, the resources that are collection centers should be declared of type `Home`, while the others should be related using `down` and `next` (`up` and `prev` can be inferred).

So, if the corporate ontology already provide relationships that can be exploited for building a collection the mapping approach can be use even here. For instance, if the corporate memory contains a set of courses, whose priority is expressed using the property `dc:requires`[6] and composition using `dc:partOf`, a collection can be created making `dc:requires` a subproperty of `next` and `dc:partOf` a subproperty of `down`.

Otherwise, as we anticipated in 3.1, the high variability and user specificity of such collections might require to draw[7] new relationships, between resources already present in the corporate memory, and sometimes to add ad-hoc resources. For instance, if we wanted to create a "successful story" collection, we should create an ad-hoc HTML document to be

[6] A property of the Dublin Core metadata element set (see `http://dublincore.org/`)

[7] Either manually or automatically by using query mechanisms

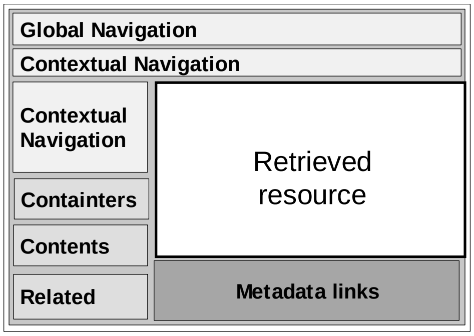

Figure 3: The outline of the prototype navigation panel.

used as a collection center and relate resources describing stories of this kind with `next`, starting from the most recent one and going back in time.

4.5 Switching between different navigation and access models

In order to show how different views ,of the same corporate memory, can be generated by combining navigation and access models, we develop also a "management service" (available on-line at `http://seip.cefriel.it/seip/manager.html`) that can be used to switch between a set of available corporate memories mounting different navigation and access models [8].

5 Related works

We already discuss the differences between the presented approach and WWW conceptual modeling, we get most inspired from. So in this section we highlight the differences from other works in the Semantic Web community.

COHSE

The approach that shows more similarities with ours is COHSE [Carr *et al.*, 2001]. In fact, its main concern is in linkage and navigation aspects between web pages. It improves the consistency and breadth of linking of web pages by deriving links among them from metadata describing their contents. But it doesn't provide a way to model explicitly different navigation models (for different large group of homogeneous user) and different access models (for specific views tailored to smal groups of users) as we do.

SEAL and SEAL-II

Another approach that presents sameness is SEAL [Maedche *et al.*, 2001] and its recent evolution SEAL-II [Hotho *et al.*, 2001] that offer a comprehensive set of industrial strength tools for building knowledge portals [Staab and Maedche, 2000]. With them we share the idea of using semantic annotation and, in particular, ontologies as an affordable way to integrate heterogeneous resources of information. We both use ontologies as a conceptual backbone for building and maintaining portals, but SEAL-II uses pre-semantic web technologies (e.g. F-logic, Ontobroker [Fensel *et al.*, 1999]) while we build our prototype using RDF, RDFS and OWL property characteristics.

Moreover, they design the corporate ontology explicitly for the portal (while it should be portal independent) and do not keep it decoupled form navigation and access models (which, otherwise, are portal specific models). On the contrary, we recognize in a metadata-bases ontology-oriented solution a major progress in interoperability, thus we push for a distributed and autonomous approach. In this scenario, the semantic EIP is only one among many applications that can "understand" the metadata that describes resource contents. Thus it cannot count on any "a priori" agreement on the corporate ontology .

Finally, we both support use of ontologies at browsing time, but, while SEAL-II build views using combined queries for schema and content, we model navigation as access explicitly as WWW conceptual models do.

KAON portal

KAON [Bozsak *et al.*, 2002] is an open-source ontology management infrastructure targeted for business applications. One of its component is KAON Portal that is a simple tool for generating multi-lingual, ontology-based Web portals. With this approach we share the model-view-controller pattern, but while KAON stress more scalability and performance issues, we focus more on giving an homogeneous navigation experience to user despite the heterogeneity that characterise the resources.

Others projects

Among the other project we want to highlights SemIPort [9], a recently started project we share some objective with, and two really successful examples of pre-semantic web application we have already cited: SHOE [Heflin and Hendler, 2000] and Ontobroker [Fensel *et al.*, 1999].

SHOE provides mechanisms that allows the definition of ontologies and the embedding in HTML pages of metadata referring to those ontologies, then a SHOE enable browser can show these claims to the user and guide him from page to page.

Ontobroker shows many similarity with SHOE. It allows the annotation of HTML pages with metadata, but it provides, with F-logic, a more expressive ontology definition language, that it uses for specification of ontologies, metadata augmentation and queries.

6 Conclusion

In conclusion, we assumed that, as soon as some corporate semantic webs are in place, enterprises will need to update their EIPs to *semantic EIPs* able to make the best use of the available information and services . Thus, in our approach to semantic EIP, the portal does not generate new resources on demand but it provides a *resource discovery* feature for retrieving resources (available on the corporate web) and *a navigation panel* that contains one of the retrieved resources and a set of automatically generated links. This way a user

[8] In case you want to try it, please remember to go back to the semantic EIP to see the differences

[9] `http://km.aifb.uni-karlsruhe.de/semiport/partners.html/overview.html`

can choose either to manually analyse the resource (as he/she does today) or move to other related resources using the links on the navigation panel.

We believe that in the next few years Semantic Web technologies will prove successful in developing an EIP because:

- RDF Schema and OWL are, at the same time, simple enough to be easily understood, and rich enough to formally express complicated models in an application independent way. In particular, ontology mapping primitives of OWL are very helpful in merging independently developed vocabularies.
- RDF, while having (at least) an XML syntax, is more flexible than XML, hence it assures a higher degree of interoperability. In particular, RDF merging functionality makes it easy to integrate metadata coming from different sources.

So, making lever on Semantic Web technologies, the described approach for semantic EIPs brings many innovation in EIP development:

- it imposes no restriction but the use of RDF, RDF Schema and OWL in building the corporate ontology;
- it doesn't require the information carried by the metadata to be coded in any particular way, thus this information is reusable;
- it enables both resources and metadata management in a distributed and, when necessary, autonomous way as long as resources are network retrievable;
- it offers a homogeneous navigation experience among heterogeneous resources distributed over a corporate semantic web through mapping of corporate terminology to the portal terminology.

So, a semantic EIP, built using the proposed approach, will give a unified view of the information present in the corporate semantic web, while the enterprise can keep developing distributed and autonomous systems on an ad-hoc basis (as required by contingency plans) and singular enterprise departments can keep their degree of autonomy in managing such systems.

Acknowledgements

The implementation of the prototype reported in this paper has been partially founded by Engineering as part of the activities of the XV Master in Information Technology of CEFRIEL-Politecnico of Milano. We thank our colleagues, in particular Paolo Castagna and Stefano Campanini, for precious contributions, and our student Lara Marinelli.

References

[Baresi *et al.*, Jan 2000] L. Baresi, F. Garzotto, P. Paolini, , and S. Valenti. HDM2000: The HDM Hypertext Design Model Revisited. Tech. report, Politecnico di Milano, Jan. 2000.

[Bozsak *et al.*, 2002] E. Bozsak, M. Ehrig, and S. Handschub. Kaon – towards a large scale semantic web, 2002.

[Carr *et al.*, 2001] Les Carr, Wendy Hall, Sean Bechhofer, and Carole A. Goble. Conceptual linking: ontology-based open hypermedia. In *World Wide Web*, pages 334–342, 2001.

[Ceri *et al.*, 2000] Stefano Ceri, Piero Fraternali, and Aldo Bongio. Web Modeling Language (WebML): a modeling language for designing Web sites. *Computer Networks (Amsterdam, Netherlands: 1999)*, 33(1–6):137–157, 2000.

[De Troyer and Leune, 1998] O. M. F. De Troyer and C. J. Leune. WSDM: a user centered design method for Web sites. *Computer Networks and ISDN Systems*, 30(1–7):85–94, April 1998.

[Dean *et al.*, March 2003] M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, and L. Stein. OWL Web Ontology Language Reference. Working draft, World Wide Web Consortium, March 2003.

[Eriksson *et al.*, 1999] H. Eriksson, R. Fergerson, Y. Shahar, and M. Musen. Automatic generation of ontology editors, 1999.

[Fensel *et al.*, 1999] Dieter Fensel, Jurgen Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnurr, Steffen Staab, Rudi Studer, and Andreas Witt. On2broker: Semantic-based access to information sources at the WWW. In *WebNet (1)*, pages 366–371, 1999.

[Garzotto *et al.*, 1993] Franca Garzotto, Paolo Paolini, and Daniel Schwabe. HDM — A model based approach to hypertext application design. *ACM Transactions on Information Systems*, 11(1):1–26, January 1993.

[Gruber, 1993] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

[Haarslev and Moller, 2001] Volker Haarslev and Ralf Moller. High performance reasoning with very large knowledge bases: A practical case study. In *IJCAI*, pages 161–168, 2001.

[Heflin and Hendler, 2000] Jeff Heflin and James A. Hendler. Dynamic ontologies on the web. In *AAAI/IAAI*, pages 443–449, 2000.

[Hotho *et al.*, 2001] Andreas Hotho, Alexander Maedche, Steffen Staab, and Rudi Studer. SEAL-II - the soft spot between richly structured and unstructured knowledge. *Journal of Universal Computer Science*, 7(7):566–590, 2001.

[Maedche *et al.*, 2001] Alexander Maedche, Steffen Staab, Nenad Stojanovic, Rudi Studer, and York Sure. SEAL – A framework for developing SEmantic Web PortALs. *Lecture Notes in Computer Science*, 2097:1–??, 2001.

[Mecca *et al.*, 1998] Giansalvatore Mecca, Paolo Atzeni, Alessandro Masci, Paolo Merialdo, and Giuseppe Sindoni. The ARANEUS web-base management system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD-98)*, volume 27,2 of *ACM SIGMOD Record*, pages 544–546, New York, June 1–4 1998. ACM Press.

[RDF, 1999] Resource Description Framework (RDF) Model and Syntax Specification. Recommendation, World Wide Web Consortium, 1999.

[RDF, 2002] RDF Vocabulary Description Language 1.0: RDF Schema. Working draft, World Wide Web Consortium, 2002.

[Schwabe *et al.*, 1996] D. Schwabe, G. Rossi, and S. D. J. Barbosa. Systematic hypermedia application design with OOHDM. In ACM, editor, *Hypertext '96, Washington, DC, March 16–20, 1996: the Seventh ACM Conference on Hypertext: Proceedings*, pages 116–128, New York, NY, USA, 1996. ACM Press.

[Staab and Maedche, 2000] Steffen Staab and Alexander Maedche. Knowledge portals: Ontologies at work. *The AI Magazine*, 22(2):63–75, 2000.

[Sure, 2002] Y. Sure. A tool-supported methodology for ontology-based knowledge management, 2002.