

Toward a framework for Semantic Organizational Information Portal

Emanuele Della Valle and Maurizio Brioschi

CEFRIEL - Politecnico of Milano, Via Fucini 2, 20133 Milano, Italy

Abstract. Information Portals have gathered lot of attention among many organizations interested in a single point of access to their information and services. But developing portals from scratch is sometimes too expensive, so many vendors have proposed frameworks to make it affordable. Notwithstanding the frameworks the market offers seem stuck in a simplicity vs. flexibility trade off imposed by the Web technologies they are built with. Therefore we believe that a technology change is required and that Semantic Web technology can play a key role in developing a new, Semantic, generation of simpler and, at the same time, flexibler frameworks for Organizational Information Portal.

1 Introduction

Any organization¹ daily produces and consumes a great amount of information. The need for managing it goes back to ancient times but since information communication technologies have made it possible to share and store it in an inexpensive way this need seems to have grown stronger year after year. Take the last decade wide deployment of intranet solutions as an example, as they gain in popularity among organizations, 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 organizations have ended up with a huge set of repositories of structured, semi-structured and unstructured information distributed over their intranets. Such amount of information is normally *comprehensible to accustomed users*, but, occasional and novel users would have an hard time in getting to what they are looking for, because they would probably get no automatic answer to such questions as: What information do we have? Where is it? How did it get there? Who put it there? How do I get it? How can I add more? What does it mean? Can you provide context information? Can you provide more specific information?

In the last years portals have gathered lot of attention among enterprises interested in addressing these questions, due to their ability in offering the user a unique and structured access to heterogeneous information and web based services. In particular many vendors² have proposed frameworks, for facilitating

¹ by “organization” we mean “any large group of people who work together”, such as enterprises, public bodies, universities, associations, etc.

² BEA, Broad Vision, Hummingbird, IBM, Microsoft, Oracle, Plumtree and Sybase

the construction of portal solutions specific for enterprises called Enterprise Information Portals (EIPs). However, since we are facing a broader application of EIPs to organizations, rather than only to enterprises, we prefer to refer to such frameworks as Organizational Information Portal Frameworks (OIP-Fs) and we call Organizational Information Portals (OIPs) the Web applications constructed with OIP-Fs.

In a OIP we distinguish several kind of interaction times:

- *browsing time*: when generic members of the organization (we call *users*) either navigate through or search in the OIP in order to find the information or service they need to accomplish their tasks;
- *authoring time*: when authorized members (we call *editors*) publish, update or delete information or services;
- *shaping time*: when an authorized and skilled member (we call *shaper*) forges the interfaces for both users and editors;
- *administering time*: when an authorized and skilled member (we call *administrator*) decides which operations each member can perform on which information source or service.

Moreover the common set of requirements an organization usually ask an OIP-F for are: easiness (velocity and bargain rate) to develop upon; integrability with the broadest set of existing information sources and web-based services; scalability and adaptability in serving users that are accessing the OIP not only within the intranet but more and more from the extranet using portable devices; and, last but not least, long time maintainability. Most of these requirements are partially incompatible: some call for simplicity, others involve flexibility. So a common problem in developing a OIP-F is trading off between simplicity and flexibility.

We believe the market offers OIP-Fs that are stuck in a simplicity vs. flexibility trade off imposed by the Web technologies they are built with. Available OIP-Fs generally suffers from the huge amount of manual work still required for finding, extracting, representing, interpreting, and maintaining organizational information and services. 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? Right now only a technology change can enable a further step in the direction of developing a new generation of OIP-Fs. This is why we believe the innovative Semantic Web technologies will play a key role: a little semantics, provided by explicitly augmenting resources (both content and services) with metadata, the meaning of which is formally defined using ontologies, can help. It can enable both local constrains and global universality, thus many interoperable structures can be adopted underneath (RDB, XMLDB, etc.). It can endow information management in a distributed and autonomous way, making information reuse possible. Last, but not least, if a little semantics is provided then machines can, in a way, “understand” the content of a resource and its relationships with others resources; thus Semantic Web technologies can help in automatically finding, extracting, representing, interpreting, and maintaining resources. We call this new generation of OIP-Fs based on Semantic

Web technologies, Semantic Organizational Information Portal frameworks, or shortly Semantic OIP-Fs.

In the following we will introduce the four ingredients (section 2) we used in making up our concept of Semantic OIP-F (section 3). Later on, section 4 describes a prototype we have developed as an early proof of our concept. Before concluding, in section 5 we give a short survey of related works.

2 Ingredients in our concept

The innovative idea, first proposed in SEAL [1], is straightforward: can we use metadata defined by ontologies to support the construction of portals? And if so, does it help?

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 (MVC) design pattern, propose to separate data, business logic, and presentation.
- *Hypertext Architectures* and, in particular, the *WWW conceptual models* such as WebML [2], W2000 [3], HDM [4], OOHDM [5], Araneus [6], and WSDM [7], 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.

2.1 Web frameworks and Model-View-Controller approach

MVC dates back to the '80s and today it is one of the most recommended architectural design pattern for Web applications. MVC suggests to separate an application in three types of component: a model, some view and a controller. The *model* manages the data of the Web application responding to requests for information about its state (usually from the view), and to instructions to change state (usually from the controller). Each *view* provides data presentation and manages the user input. Finally, the *controller* interprets the user inputs, commanding the model and/or the views to change as appropriate.

The market provide a variety of frameworks based on MVC design patterns for Web application fast development. Just surfing the web looking for such frameworks you might run into Struts (“a flexible control layer based on standard technologies like Java Servlets, JavaBeans, ResourceBundles and XML”), Tapestry (“an alternative to scripting environments such as JavaServer Pages or Velocity) or Jetspeed (“an Open Source implementation of an Enterprise Information Portal, using Java and XML”).

2.2 WWW conceptual models

We surveyed various model-driven approaches to design Web applications available in WWW conceptual model literature [2–7]. They show it is possible, and even convenient, to model separately at least the domain information space, the navigation, the access and sometimes also the operations. So, drawing these ideas back to OIP-F development we have formulated the following definitions.

The *domain information model* is a shared understanding of the information the OIP makes available that does not change, or changes slowly, over the time. For instance, in modeling CEFRIEL (our organization) information space we can 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. The domain information model once instantiated can capture, at least in part, the organizational memory.

The *navigation models* represent the heterogeneous paths the OIP users can adopt in traversing the organizational memory. They are not necessarily shared among all users, but they are jointly employed by homogeneous categories of users. For instance, taking CEFRIEL organizational memory 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 navigation paths between researchers and their competency should be stressed, while those between researchers and units should be left in the 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 navigation paths among CEFRIEL, its units and the people who work for them and leave competency in the background instead.

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), “favorites” and so on.

The *operation models* represent both user operations and system operations. User operations are those directly visible to users. For instance, the operation “include resource in favorites”, which allows users to put a reference to the select resource in the collection of their favorites, is a typical user operation. Another common example is a “filter operation”, which allows users to specify some selection criteria other a collection in order to select a subset of elements of interest. Contrariwise, system operations are not visible to users, but are triggered by user/editor/administrator/shaper operations. For instance, the function that, triggered by “include resource in favorites”, updates the collection of favorites or the operation that “implements” the filtering, are system operations.

In order to design the domain information model WWWCM approaches use existing modeling languages: E-R diagrams in HDM [4], UML class diagrams in W2000 [3] and OOHDM [5], etc. Moreover, they explicitly provide a set of ad-hoc primitives to model navigation (e.g. entity, component, node, slot, structural link, semantic link, etc.), access (e.g. collection, collection center, etc.) and, in case they do, operation (e.g. input, pre-condition, session, etc.). On the contrary we propose to adopt ontologies and metadata.

2.3 Ontology-oriented metadata-based solutions

Metadata-based solutions have already proved to be able to 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. For instance users of a metadata-based search engine might get a smaller result set, containing more relevant resources, but once they have selected one of the retrieved resource, asking the OIP to provide context information or a more specific resource remains impossible.

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 [8] 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. This way, beside a centralized and controlled approach, a distributed environment, where autonomous entities maintain heterogeneous shared resources describing them with metadata defined by the corporate ontology, becomes feasible.

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 [9] and SHOE [10] 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 Resource Description Framework (RDF) [11], a framework for describing web resources with metadata, and Ontology Web Language (OWL) [12], a comprehensive ontology vocabulary that has just become a W3C Proposed Recommendation.

3 Making up our concept

So far we have identified the key ingredients: *Web Frameworks* are the state of the art in developing Web application; *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. But we still have to combine them to make up our concept.

3.1 Our model

We start considering that even if the global Semantic Web might still be a little bit too ambitious, however, as the standardisation of the Semantic Web technologies proceeds, some pioneering organizations would be able to build “local organizational Semantic Webs” represented by services and documents available over the intranet (or the extranet) annotated with metadata defined by a corporate ontology materialized in a sort of *organizational memory*. So the domain information model, which WWCM approaches usually explicitly ask the modeler to provide, will be already captured by the organizational memory. But, differently from WWCM, this organizational memory will be completely decoupled from the Semantic OIP design, because the OIP will be only one, among many other, application to use the information available over the organizational Semantic Web. Therefore a Semantic OIP built using our framework, cannot assume any “a priori” agreement over this domain information model except the use of a common set of primitives provided by the Semantic Web technologies (e.g. XML, RDF, OWL, ect.).

So, we don’t require the shaper to model explicitly navigation and access using a specific set of primitives as many WWW conceptual model approaches suggest. We propose, alternatively, to define a navigation and an access terminology modeling respectively navigation and access in a domain independent way, using metadata defined by ontologies and, then, we support the shaper in building navigation and access models mostly using OWL mapping terminology. In fact, we believe navigational models can be build by *mapping* some domain information terms to the navigation terminology, but in building access structure, even if mapping corporate ontology terminology to the access terminology could prove to be useful, the high variability and user specificity of such models might require to explicitly draw new relationships between resources in the corporate memory and also to add ad-hoc resources (e.g. centers of collection). Even so, we expect the latter task to be largely automatized by dynamically querying the organization memory or by profiling the interaction with the users.

Finally if we wanted to model operation using Semantic Web technologies, we could define an operation terminology as in the two previous case; but understanding that we might need to extend such terminology using rules, which are not standardized yet, we prefer to leave this task for future works.

3.2 Our controller

Decided what to model, we conceive the control operations we want to make available to each user, editor, shaper and administrator at his/her specific interaction time.

At browsing time Web users interact with the Web in many ways, but three patterns are commonly recognized: searching, navigation and bookmarking. Users do search when they know exactly what they are looking for, hence they are able to express their requests with sufficient precision. Otherwise users do navigate either when they know how to get to a resource or when they don't know what they are looking for but they understand it as they browse. In both cases users bookmark a resource when they want to store a direct entry point to a particular resource. A semantic OIP should exploit metadata and ontologies in order to improve all interaction patterns. In particular we want to improve searching by resource discovering, navigation by automatic link creation and bookmarking by personal access modeling.

At authoring time ontologies, in particular the corporate ontology, can be exploited in supporting the editorial task of adding new resources or describing existing ones. It has already been shown (e.g. in Protégé 2000 [13]) that they can be employed in automating part of process for creating editorial interfaces. But we believe most of the benefits of using Semantic Web technologies 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 project results 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.

At shaping time beside the commonly available controls that enable visual page composition, we want the shapers to be able to define both navigation and access models. Using OWL mapping terminology via a graphic interface a shaper maps the corporate terminology to the navigation terminology in order to build a navigation model for each category of user. For instance, if we define in the navigation terminology the property `related` and we determinate its presentation functionality, then a CEFRIEL shaper might model a navigation path intended for researchers by mapping the relationship between researchers and their competency to `related`.

Moreover, a shaper can build visually a shared access structure either by mapping the corporate terminology to access terminology or by querying. For instance, a CEFRIEL shaper might build an access structure by defining a query that extract the competency most frequently required in projects of the last two years.

At administering time we want to support administrators³ in authorizing: users to selectively access the resources, editors to describe and publish resources available on the organizational Semantic Web and shapers to manage navigation and access models. Administration operation don't probably get too benefit from the introduction of Semantic Web technologies, but defining permission on a concept basis instead of a resource basis might drive some improvement in managing such task, too.

3.3 Our view

We choose not to model presentation explicitly (as some WWCM have tried), 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 MVC approach. This way we aspect the same advantages, in term of visual coherence and accessibility, as modeling but at a more affordable effort.

4 Concept refinement and an early proof of concept

At the current state of the work we are refining the concept, in particular *we are investigating how to improve navigation by creating links automatically and displaying them in a navigation panel*. The rational behind is strain forward: a Semantic OIP built using our framework can, in a way, understand the meaning of the metadata used to describe a resources; so, when a user asks the Semantic OIP to retrieve a resource at a given URL, it returns the required resource inserted in a navigation panel

that automatically provides the user with links to both context information and more detailed information. In fact, it is possible to provide the user with automatically generated links to context (or more detailed) information as long as the relationship among these resources can be inferred by a *reasoner service*. The challenge is to offer the users clearly understandable metaphors, such as a landmark (the position of a well-known resource that plays the role of reference point), and orientation tools, such as a compass (a navigational instrument for finding directions).

In our vision the semantic EIP can generate, at least, three different kinds of links:

- *access point links* are generated by asking a reasoner service which resources should be reachable according to the current access model. We distinguish between *global* and *contextual access points*. The former type is always visible

³ Please note that here we are not interested in fine grain access controll to concepts and instances in the corporate ontology but to the easier problem of managing user access to the portal accessible resources.

- (such links are sort of landmarks that help users in taking their’ bearings), while the latter type depends on the retrieved resource (such links are a sort of compass that guides users locally in accessing the information).
- *categorized links* are generated by asking the reasoner service to group together, according to the current navigation model, those resources that are related via the same property or its subproperties. Links in this category are provided in order to give users an idea of both the context in which the retrieved resource is located and if more detailed information is available.
 - *metadata links* should provide the user with both a simple visualization of the metadata describing the retrieved resource and an intuitive navigation of the semantic relationships to other resources.

In order to proof this concept, we have built a first prototype of a semantic OIP-F following the presented approach and we have demonstrated its functionalities by constructing a Semantic OIP to access a “synthetic” organizational memory that describes CEFRIEL (an on-line demo is available at <http://seip.cefriel.it>). In particular we developed a servlet-based application that uses Velocity for implementing MVC pattern, Jena-2.1⁴ to manage RDF triples and RACER 1.7.15 [14] as reasoner service that can “understand” RDF and OWL-DL. An overall architectural vision of our prototype is shown in figure 1.

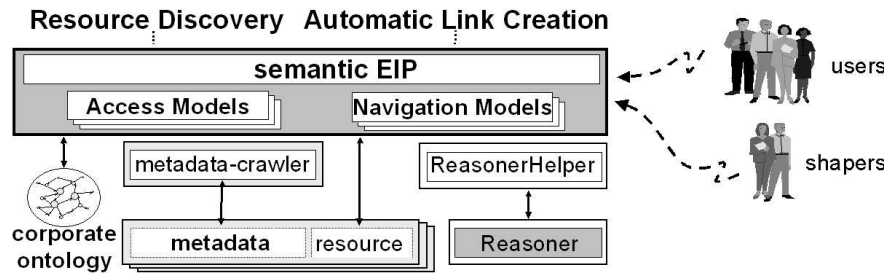


Fig. 1. Logical architecture of the proposed framework for Semantic OIP.

Moreover we have developed a first draft of the navigation and access terminology we have introduced in section 3.1. We got inspired either by common terms used in WWW conceptual modeling and in HTML 4.1 link types⁵. We kept them explicitly as simple as possible, but still rich enough to be useful in proving the concept. On the one hand, our *navigation ontology* defines four properties:

- **contains**: a transitive property to express containment (e.g. a museum contains art)
- **contained**: inverse property of contains

⁴ See <http://www.hpl.hp.com/semweb/> for details

⁵ See <http://www.w3.org/TR/html4/types.html#type-links>

- `seeAlso`: a property to express a relevant connection
- `related`: a symmetric subproperty of `seeAlso`

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

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

Then we define and code a visual behavior for each term in the two terminology in order to automatically generate and present the three different kind of links present in our navigation panel.

The prototype, “understanding” RDF and OWL-DL, can process the metadata that describe the retrieved resource, generating *metadata links* according to the following schema:

```
<resource label>[<list of labels of types>]
<property 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 a user asks to retrieve CEFRIEL home page, 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] 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 head of unit, works for CEFRIEL.

The prototype 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 links can appear in this box 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 either via a `seeAlso` or a `related` property.

As we explain in 3.1 we don't want to oblige the shaper to use directly the terms defined in the navigation and access ontology, we want instead to play on OWL mapping primitives. Thus, the shaper should not connect resources present in the organizational memory directly with **contains**, **contained**, **seeAlso** and **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 nine 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, a shaper can put in the “navigation model” the triple **has_unit rdfs:subpropertyOf contains**. This way, when CEFRIEL home page is retrieved, links to all the 9 unit of CEFRIEL are placed in the “contains” box.

In the prototype we have provided a special control, visible only at shaping time, to support the mapping task in a WYSIWYG approach. When a user is logged in as a shaper, beside each property in the “metadata links” we show the current mapping for the property. If no mapping is already present a combo box listing the available terminology is shown, otherwise a link labeled “del” give the shaper the possibility to delete the mapping.

Finally the prototype has got as *access point links* 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. 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 new relationships, between resources already present in the corporate memory, and sometimes to add ad-hoc resources. For instance, if a shaper wanted to create a “successful story” collection, we

⁶ A property of the Dublin Core metadata element set (see <http://dublincore.org/>).

should create an ad-hoc HTML document to be 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.

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 models⁷.

Finally, at the time we write this article we are investigating how *improve searching by resource discovering*. In fact if semantic metadata is available, a search engine would not be exclusively based on full text search, but it could make lever on semantics, so it could, in a way, 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. Moreover such a semantic search engine wouldnt retrieve only a list of unrelated resources, but it could organize them in a structured collections by exploiting the relationships among the retrieved resources.

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.

An approach that tries, as we do, to combine WWWCM and Semantic Web is *SHDM* [15]. SHDM is a direct extension of OOHDM [5] that use ontologies to define an application conceptual model, extending the expressive power of the original method and use RQL query language to specify flexible navigation model and access structures.

Another similar approach is *OntoWebber* [16]. It proposed to explicit model the Web sites, using ontologies as the foundation for Web portal design and semi-structured data technology for data integration. OntoWebber was the basis for creating the Semantic Web Community Portal.

An approach that presents sameness is *SEAL* [1] and its recent evolution *SEAL-II* [17] that offer a comprehensive set of industrial strength tools for building knowledge portals [18]. 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 [9]) while we build our prototype using RDF and OWL-DL.

However SHDM, OntoWebber and SEAL are still oriented to traditional Web Applications. They only take limited advantages from the Semantic Web

⁷ In case you want to try it, please remember to go back to the semantic EIP to see the differences.

technologies, considering them a set of richer formal languages to use for accomplishing their goal. Neither of them stress the necessary strong decoupling of the constructed OIP from the organizational Semantic Web. They all propose to design the organizational ontology explicitly for the portal and they assume complete controllability over it. 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 OIP is only one among many applications that can “understand” the metadata that describes resource contents. Thus we prefer to resign any “a priori” agreement on the organizational memory.

The approach that shows more similarities to our automatic link creation concept is *COHSE* [19]. 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 small groups of users) as we do.

KAON [20] 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 MVC 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.

Among the other projects we want to highlights *ODESeW* [21], a recently published ontology-based application to automatically generates and manages a knowledge portal for intranets and extranets, *SemIPort*⁸, a newly started project we share some objective with, and two really successful examples of pre-Semantic Web application we have already cited: *SHOE* [10] and *Ontobroker* [9].

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

We believe that Semantic Web technologies in the next few years will break through as the technology change that developers of OIP frameworks require for moving a step further in the direction of a better trade off between simplicity and flexibility. So we propose a novel approach to a framework for Semantic

⁸ see <http://km.aifb.uni-karlsruhe.de/semiport/partners.html/overview.html>

OIPs that, making lever on Semantic Web technologies, brings many innovation in OIP development:

- it imposes no restriction but the use of RDF 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 an Organizational Semantic Web mostly by mapping corporate terminology to the terminology known by our Semantic OIP-F;
- it provides a light weight multi-lingual support.

Furthermore in our prototype we investigate the *visual construction at shaping time of navigation and access models* that we exploit at browsing time via a *resource discovery* feature for retrieving resources (available on the intranet) and a *navigation panel* that contains one of the retrieved resources and a set of automatically generated links. Therefore, a Semantic OIP, built using the proposed approach, will give a unified view of the information present in the organizational Semantic Web, while the organization can keep developing distributed and autonomous information systems on an ad-hoc basis (as required by contingency plans) and singular departments can keep their degree of autonomy in managing such systems.

Acknowledgements

The research has been supported by the MAIS project while 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

1. Alexander Maedche, Steffen Staab, Nenad Stojanovic, Rudi Studer, York Sure: SEAL – A framework for developing SEmantic Web PortALs. Lecture Notes in Computer Science **2097** (2001)
2. Stefano Ceri, Piero Fraternali, Aldo Bongio: Web Modeling Language (WebML): a modeling language for designing Web sites. Computer Networks (Amsterdam, Netherlands: 1999) **33** (2000) 137–157
3. L. Baresi, F. Garzotto, P. Paolini, S. Valenti: HDM2000: The HDM Hypertext Design Model Revisited. Tech. report, Politecnico di Milano (Jan. 2000)

4. Franca Garzotto, Paolo Paolini, Daniel Schwabe: HDM — A model based approach to hypertext application design. *ACM Transactions on Information Systems* **11** (1993) 1–26
5. D. Schwabe, G. Rossi, S. D. J. Barbosa: Systematic hypermedia application design with OOHDm. In ACM, ed.: *Hypertext '96*, Washington, DC, March 16–20, 1996: the Seventh ACM Conference on Hypertext: Proceedings, New York, NY, USA, ACM Press (1996) 116–128
6. Giansalvatore Mecca, Paolo Atzeni, Alessandro Masci, Paolo Merialdo, 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.*, New York, ACM Press (1998) 544–546
7. O. M. F. De Troyer, C. J. Leune: WSDM: a user centered design method for Web sites. *Computer Networks and ISDN Systems* **30** (1998) 85–94
8. T. Gruber : A translation approach to portable ontology specifications. *Knowledge Acquisition* **5** (1993) 199–220
9. Dieter Fensel, Jurgen Angele, Stefan Decker and Michael Erdmann, Hans-Peter Schnurr, Steffen Staab, Rudi Studer, Andreas Witt: On2broker: Semantic-based access to information sources at the WWW. In: *WebNet* (1). (1999) 366–371
10. Jeff Hefflin, James A. Hendler: Dynamic ontologies on the web. In: *AAAI/IAAI*. (2000) 443–449
11. : Resource Description Framework (RDF) Model and Syntax Specification. (2004)
12. M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein: Owl web ontology language reference. Recommendation, World Wide Web Consortium (2004)
13. H. Eriksson, R. Ferguson, Y. Shahar, M. Musen: Automatic generation of ontology editors. In: *Proceedings of the 12th Banff Knowledge Acquisition Workshop*, Banff, Alberta, Canada (1999)
14. Volker Haarslev, Ralf Miller: Racer: A core inference engine for the semantic web. In: *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003)*, Sanibel Island, Florida, USA (2003) 27–36
15. Fernanda Lima, Daniel Schwabe: Application modeling for the semantic web. In: *LA-WEB 2003 - First Latin American Web Conference*, Santiago, Chile, IEEE-CS Press (2003) 802–817
16. Yuhui Jin, Sichun Xu, Stefan Decker, Gio Wiederhold: Managing Web sites with OntoWebber. *Lecture Notes in Computer Science* **2287** (2002) 766
17. Andreas Hotho, Alexander Maedche, Steffen Staab, Rudi Studer: SEAL-II - the soft spot between richly structured unstructured knowledge. *Journal of Universal Computer Science* **7** (2001) 566–590
18. Steffen Staab, Alexander Maedche: Knowledge portals: Ontologies at work. *The AI Magazine* **22** (2000) 63–75
19. Les Carr, Wendy Hall, Sean Bechhofer, Carole A. Goble: Conceptual linking: ontology-based open hypermedia. In: *World Wide Web*. (2001) 334–342
20. E. Bozsak, M. Ehrig, S. Handschub: Kaon – towards a large scale semantic web. In: *Proc. of the 3rd Intl. Conf. on E-Commerce and Web Technologies (EC-Web 2002)*. (2002) 304–313
21. Oscar Corcho, Asuncin Gmez-Prez, Angel Lpez-Cima, V. Lpez-Garca, Mara del Carmen Surez-Figueroa: Odesew. automatic generation of knowledge portals for intranets and extranets. In D. Fensel, ed.: *The Semantic Web, ISWC 2003*, LNCS 2870. (2003) 802–817