

# Table of Contents

<b>1. The Knowledge Portal "OntoWeb"</b>	
Daniel Oberle and Peter Spyns .....	1
1.1 Introduction .....	1
1.2 The OntoWeb Project .....	2
1.3 DOGMA – The core idea .....	4
1.4 SEAL – The core approach .....	4
1.5 Applying SEAL to OntoWeb .....	6
1.6 Content Provision .....	7
1.6.1 Content Syndication .....	8
1.6.2 Content Objects .....	8
1.7 Content Access .....	11
1.7.1 Browsing .....	12
1.7.2 Typed term-based Querying .....	12
1.7.3 Combining browsing and free term-based Querying ...	13
1.8 Content Presentation .....	15
1.9 Related work .....	15
1.10 Future Work and Conclusion .....	17
References .....	18

# 1. The Knowledge Portal "OntoWeb"

Daniel Oberle<sup>1</sup> and Peter Spyns<sup>2</sup>

<sup>1</sup> Institute AIFB, University of Karlsruhe  
76128 Karlsruhe, Germany

email: {last name}@aifb.uni-karlsruhe.de

<sup>2</sup> STAR Lab, Vrije Universiteit Brussels,  
B-1050 Brussels, Belgium

email: {first.lastname}@vub.ac.be

The recent years have seen a tremendous progress in managing semantically heterogeneous data sources. Core to this semantic reconciliation between the different sources is a rich conceptual model that the various stake-holders agree on, an *ontology*. Similarly, in recent years the information system community has successfully strived to reduce the effort for managing complex web sites. The core to these different web site management approaches also is a rich conceptual model that allows for accurate and flexible access to data. SEAL (SEmantic PortAL), a framework to building community web sites, has been developed to use ontologies as key elements for managing community web sites and web portals. In addition, semantic data stores underpinning (community) web sites have to be scalable, re-usable and interoperable. DOGMA (Developing Ontology Guided Mediation for Agents) provides a robust framework making use of database technology that copes with these issues. This chapter presents a combination of the SEAL and DOGMA frameworks and elaborately illustrates our approach with examples from the OntoWeb community portal of the EU thematic network with the same name.

## 1.1 Introduction

Supporting communities in sharing and exchanging knowledge is an important aspect of Knowledge Management. This holds e.g for communities of practice being organized within enterprises or being organized by a collection of cooperating enterprises or for scientific communities that are spread all over the world and thus urgently need support in sharing their knowledge. In that context, knowledge portals [1.9] play a part in offering means for providing and accessing knowledge on a semantic level. In essence, knowledge portals exploit ontologies for achieving a conceptual foundation for all functionalities that are offered by the portal. The SEAL framework for developing and managing knowledge portals exploits Semantic Web technologies to offer mechanisms for acquiring, structuring, integrating, sharing and accessing distributed knowledge between human and/or machine agents [1.23, 1.5]. Up to now, SEAL puts emphasis on supporting the acquiring and structuring of knowledge by semantic annotation [1.20] and the automatic generation of navigational views and mixed ontology and content-based presentation.

The topic of this article is the application and extension of SEAL and the combination with DOGMA for realizing the OntoWeb community portal (<http://www.ontoweb.org>). OntoWeb is a EU IST thematic network that propagates ontologies in the context of eBusiness and Knowledge Management and that currently has more than one hundred members from research and industry. The knowledge portal that will be used as a case study throughout the article is a joint effort between the Free University of Brussels - STAR Lab - and the University of Karlsruhe - Institute AIFB. Each lab has its particular contribution to the realisation of this knowledge portal. On the one hand, the process of knowledge provisioning and publishing has to be supported by an appropriate workflow. Therefore, the AIFB SEAL framework has been extended by methods and tools for defining and handling a publishing workflow, realized by a comprehensive content management system (CMS). On the other hand, access to the knowledge is equally important, and the information extraction technology of the portal has to take advantage of the underlying ontology to come up with (more) relevant answers than traditional search or query mechanisms. Therefore, for the purposes of the OntoWeb portal the STAR Lab ontology server (called DOGMA server see [1.21] ) has been modified and extended with a query facility that processes user information requests via a graphical interface that exploits the underlying ontology [1.30].

The article is structured as follows. In section 1.2 we first talk about the OntoWeb project and the aims of its portal in general. We then briefly introduce the DOGMA initiative 1.3, followed by a description of the main components and functionalities of the SEAL framework in section 1.4. Section 1.5 outlines the scenario that is set up by the OntoWeb portal. The following sections elaborate more on the specialties of the OntoWeb portal. We focus on how content can be provided in section 1.6, viz. within the portal itself utilizing the content management facilities and implicitly by the syndication of metadata. Section 1.7 talks about how the content (or community knowledge) can be accessed, viz. by browsing, template-based querying and their combination. Finally, section 1.8 briefly sketches data presentation. We conclude with a discussion of related work (section 1.9) and an outline of open research problems (section 1.10).

## 1.2 The OntoWeb Project

The EU thematic network “OntoWeb – Ontology-based information exchange for knowledge management and electronic commerce” aims at bringing together researcher and industrials to “enable the full power of ontologies”. The project aims at improving information exchange in areas such as: information retrieval, knowledge management, electronic commerce, and bio-informatics. It will also strengthen the European influence on standardization efforts in ar-

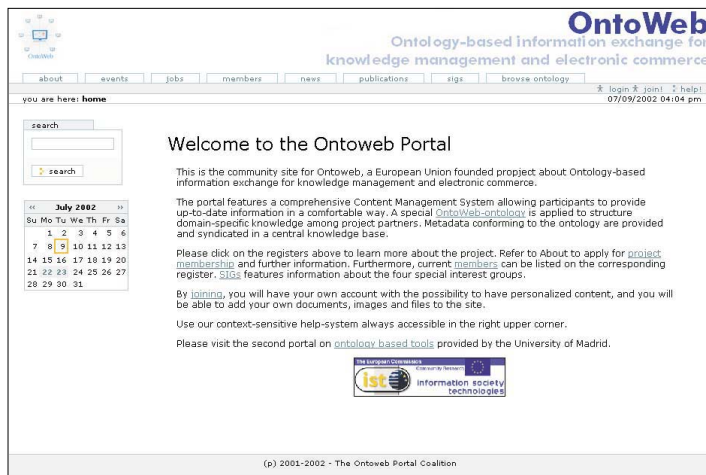


Fig. 1.1. www.ontoweb.org – The OntoWeb portal

eas such as web languages (RDF, XML), upper-layer ontologies, and content standards such as catalogues in electronic commerce” (cf. [1.14]).

One of the tasks of the OntoWeb partners was to create a portal serving as a platform for communication between partners and also between partners and other members of the World Wide Web. Through this portal knowledge can be gathered, stored, secured and accessed by the OntoWeb community. It is an open community, i.e. new members can join at any time. The positive effects of the existence of such a portal are multiple. Only the most important ones will be mentioned. At a first stage, the portal serves as an inventory of knowledge available in the community. In the case of an Internet portal, knowledge has been made available outside of the organization of the original producer or owner. E.g., members of the community get a good overview of the skills and profiles of the various community members. In the case of an intranet, it may stimulate the communication between departments of a same company and support the local (technology) innovation management process.

Turning information into knowledge that suits the above mentioned situation, requires a shared conceptualization of the domain in question. In the present case, the domain spans a conceptualization of the OntoWeb organization (e.g., companies, research institutions, special interest groups etc.), of various kinds of documents (e.g., meeting minutes, deliverables, papers etc.), of events and their organizations (e.g., conferences, workshops, internal meetings etc.), of scientific results and material (e.g., cases, programs, etc.), and so forth. A formal version of such a shared conceptualization is commonly called an ontology [1.26]. When relating specific terms to concepts, a controlled vocabulary or some other common terminological framework can be created.

### 1.3 DOGMA – The core idea

In this section we shortly present the DOGMA initiative for a formal ontology engineering framework - more details in [1.21], [1.29]. The initiative is based on the double articulation of an ontology: we decompose an ontology into an ontology base, which holds (multiple) intuitive conceptualisation(s) of a domain, and a layer of ontological commitments, where each commitment holds a set of domain rules. We adopt a classical database model-theoretic view in which conceptual relationships are separated from domain rules. They are moved - conceptually - to the application "realm". This distinction may be exploited effectively by allowing the explicit and formal semantical interpretation of the domain rules in terms of the ontology. Experience shows that agreement on the domain rules is much harder to reach than one on the conceptualisation [15]. The ontology base consists of sets of intuitively "plausible" domain fact types, represented and organised as sets of context-specific binary conceptual relations, called lexons. The layer of ontological commitments mediates between the ontology base and its applications. Each ontological commitment corresponds to an explicit instance of an (intensional) first order interpretation of a task in terms of the ontology base. Each commitment consists of rules that specify which lexons from the ontology base are visible for usage in this commitment and the rules that constrain this view (= commits it ontologically). As a result, (re-)usability, shareability, interoperability and reliability of the knowledge will be enhanced. In short, the DOGMA approach takes agreed semantical knowledge out of an IT application that makes use of an external ontology. This is done in much the same way that "classical" databases take data structures out of these applications. Likewise, ontologies built in accordance with the principle of the double articulation achieve a form of semantical independence for IT applications. A modified version of the DOGMA server functions as the central knowledge repository of the OntoWeb portal.

### 1.4 SEAL – The core approach

The recent decade has seen a tremendous progress in managing semantically heterogeneous data sources. Core to the semantic reconciliation between the different sources is a rich conceptual model that the various stake-holders agree on, an *ontology* [1.2]. The conceptual architecture developed for this purpose now generally consists of a three layer architecture comprising (cf. [1.10]) (i) heterogeneous **data sources** (e.g., databases, XML, but also data found in HTML tables), (ii) **wrappers** that lift these data sources onto a common data model (e.g. OEM [1.7] or RDF [1.6]), (iii) integration modules (**mediators** in the dynamic case) that reconcile the varying semantics of the different data sources. Thus, the complexity of the integration/mediation task could be greatly reduced.

Similarly, in recent years the information system community has successfully strived to reduce the effort for managing complex web sites [1.11, 1.13, 1.18, 1.24]). Previously ill-structured web site management has been structured with process models, redundancy of data has been avoided by generating it from database systems and web site generation (including management, authoring, business logic and design) has profited from recent, also commercially viable, successes [1.11]. Again we may recognize that core to these different web site management approaches is a rich conceptual model that allows for accurate and flexible access to data. Similarly, in the hypertext community conceptual models have been explored that im- or explicitly exploit ontologies as underlying structures for hypertext generation and use (e.g. [1.1]).

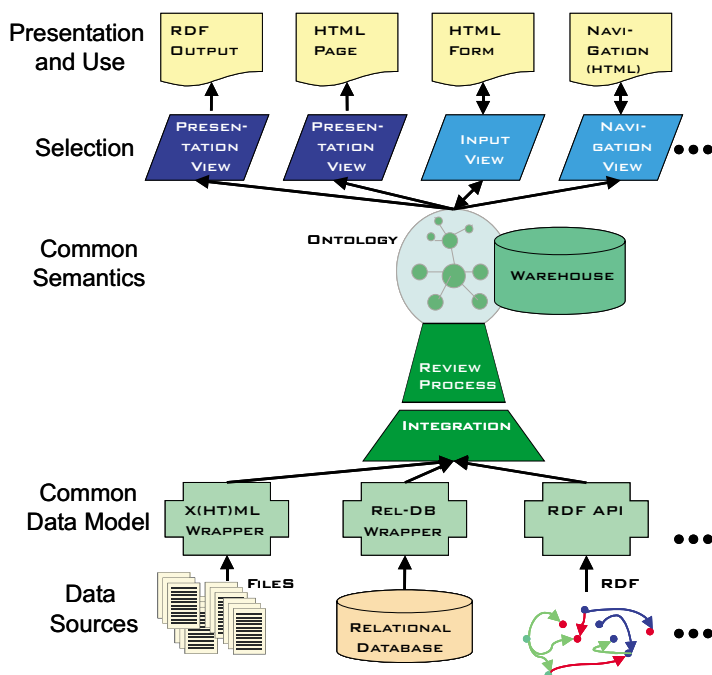


Fig. 1.2. Extended conceptual SEAL architecture

SEAL<sup>1</sup>, the AIFB framework to building community web sites, has been developed to use ontologies as key elements for managing community web sites and web portals. The ontology supports queries to multiple sources (a task also supported by semi-structured data models [1.18]), but beyond that it also includes the intensive use of the schema information itself al-

<sup>1</sup> Cf. [1.23] on the history of SEAL.

lowing for automatic generation of navigational views<sup>2</sup> and mixed ontology and content-based presentation. The core idea of SEAL is that Semantic Portals for a community of users that contribute *and* consume information [1.8] require web site management *and* web information integration. In order to reduce engineering and maintenance efforts SEAL uses an ontology for semantic integration of existing data sources as well as for web site management and presentation to the outside world. SEAL exploits the ontology to offer mechanisms for acquiring, structuring and sharing information between human and/or machine agents. Thus, SEAL combines the advantages of the two worlds briefly sketched above.

## 1.5 Applying SEAL to OntoWeb

The OntoWeb portal (cf. Figure 1.1) is structured according to an ontology that serves as a shared basis for supporting communication between humans and machines. The general goal of our approach is the semi-automatical construction of a community portal using the community’s metadata to enable information provision, querying and browsing of the portal. For this purpose we could reuse the framework as explained in Section 1.4, but we also had to provide new modules for content management resulting in the extended architecture depicted in Figures 1.2 and 1.3. In the following, we explain how SEAL is applied to OntoWeb (paragraphs *Integration* and *Presentation*) and talk about the specific extension of the portal in the form of a content management system.

*Integration.* One of the core challenges when building a data-intensive web site is the integration of heterogeneous information on the WWW. The recent decade has seen a tremendous progress in managing semantically heterogeneous data sources [1.10, 1.18]. The general approach SEAL pursues is to “lift” all the different input sources onto a common data model, in our case RDF. Additionally, the ontology acts as a semantic model for the heterogeneous input sources. As mentioned earlier and visualized in our conceptual architecture in Figure 1.2, we consider different kinds of **Web data sources** as input. However, to a large part the Web consists of static HTML pages, often semi-structured, including tables, lists, etc. In our case, we had to integrate the DOGMA Server that serves as a knowledge base for syndicated metadata like further discussed in section 1.6.1. Besides, there is the Zope Object Database (ZODB), containing content added manually by visitors of the portal (cf. section 1.6.2). The object oriented Zope is used as the central web and content server of the portal like explained in section 1.6.

---

<sup>2</sup> Examples are navigation hierarchies that appear as **has-part-trees** or **has-subtopic trees** in the ontology.

*Presentation.* Based on the integrated data in the warehouse we define user-dependent **presentation views**. First, we render HTML pages for human agents. Typically *queries for content* of the warehouse define presentation views by selecting content, but also *queries for schema* might be used, e.g. to label table headers. Second, as a contribution to the Semantic Web, our architecture is dedicated to satisfy the needs of software agents and produces machine understandable RDF.

To maintain a portal and keep it alive its content needs to be updated frequently not only by information integration of different sources but also by additional inputs from human experts. The **input view** is defined by *queries to the schema*, i.e. queries to the ontology itself. Similar to [1.3] we support the knowledge acquisition task by generating forms out of the ontology. The forms capture data according to the ontology in a consistent way which are stored afterwards in the warehouse. To navigate and browse the warehouse we automatically generate navigational structures, i.e. **navigation views**, by using *combined queries for schema and content*.

*Extensions to SEAL.* During the development of the OntoWeb portal we recognized rather soon that the process of knowledge provisioning and publishing has to be supported by an appropriate content management system in order to be able to control what content is put into the portal by whom. Only then, the high quality of content that is expected by the OntoWeb users can be guaranteed. Therefore, the SEAL framework has been extended by methods and tools for defining and handling a publishing workflow. Such a workflow represents an important constituent of the overall approach for managing a running knowledge portal to make user focussed access to the OntoWeb portal maintainable. In Figure 1.2 this is depicted as "Review Process", further discussed in section 1.6.2.

Besides content management, ontology-based annotation of the community information is a prerequisite in order to offer the possibility of knowledge retrieval and extraction (also known as conceptual or intelligent search — cf. [1.22] as an example). The usage of well-defined semantics allows for the knowledge exchange between different OntoWeb community members. Members are encouraged to publish annotated information on the web, which is then crawled by a syndicator and stored in the portal knowledge base. This mechanism can be considered as another extension to SEAL and is discussed in section 1.6.1.

## 1.6 Content Provision

Basically, there are two ways of providing content to the OntoWeb portal. First, there is the syndication mechanism, automatically gathering metadata from participating sites. Second, the portal allows for content provision itself. Both possibilities are discussed in subsections 1.6.1 and 1.6.2, respectively.



### 1.6.1 Content Syndication

The portal allows centralized access to distributed information that has been provided by participants on their own sites. To facilitate this, participants can enrich resources located outside of the portal with metadata according to the shared OntoWeb ontology. This annotation process can be supported semi-automatically by the Ontomat Annotizer tool [1.20] for instance.

As depicted in Figure 1.3, syndicating information from participants is done by replicating their metadata. The information finds its way in the DOGMA Server [1.21] that exploits a relational DBMS for storing and can be queried by users (cf. section 1.7 for a detailed discussion). Within the portal, authenticated users may generate content objects on their behalf (cf. subsection 1.6.2). As we use Zope<sup>3</sup> as underlying technology, such objects are stored in its respective database (called ZODB). Besides, metadata, both conforming to Dublin Core [1.31] as well as to the Ontoweb ontology, are generated for all the portal's objects. This can be achieved easily as all metadata are stored within Zope's own database. When adding new content to the portal, users have the possibility to supply metadata accordingly. Comparing this technical architecture to the conceptual one depicted in Figure 1.2, we find that Zope is both used for presentation as well as for storage. Besides, the DOGMA Server provides additional storage capabilities in our case. The ontology always forms the central part for the structuring of knowledge.

### 1.6.2 Content Objects

We acknowledge the fact that some members might not be able to publish data on the web on their own due to corporate restrictions or other reasons. Therefore OntoWeb participants staff members are provided with a personal space to create and manage content for the portal. To facilitate this, the portal includes a fully-fledged content management system. Additionally, all content created within the portal is automatically associated with the predefined OntoWeb design to achieve an integrated visual experience with a consistent appearance. In the personal space people can provide the following types of content: HTML-documents, arbitrary files and folders, and selected predefined content types based on ontological concepts: Publications, News, Events, Scientific Events, Jobs, etc.

If a member chooses to create new content based on the predefined content types, appropriate metadata is automatically generated. Second, all content is associated with standard Dublin Core metadata to keep track of publishing information such as date of creation, last modification, authorship and subject classification.

<sup>3</sup> cf. <http://www.zope.org>

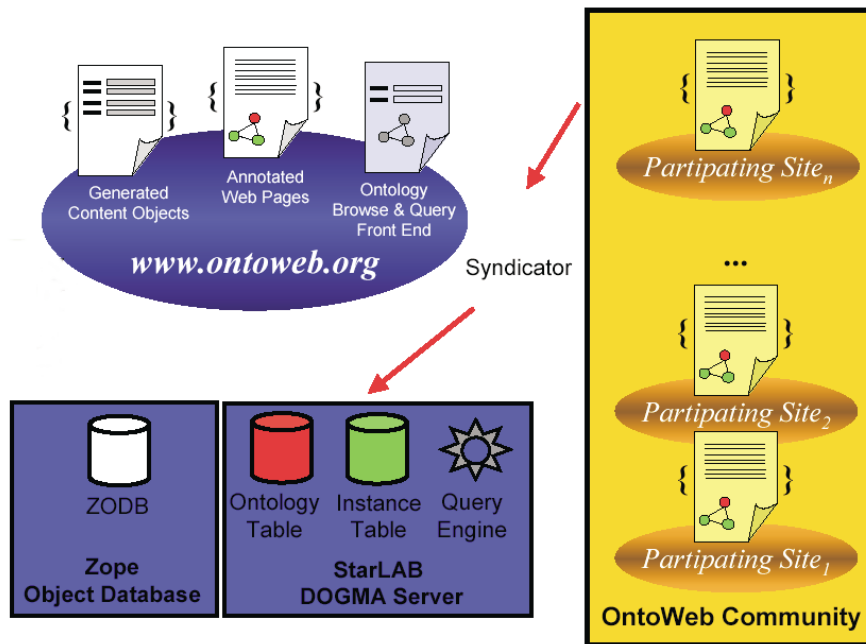


Fig. 1.3. Content Syndication

**Process Model for Publishing Workflows.** As mentioned in section 1.2, OntoWeb is an open community posing additional constraints since data that is (re)published through the portal could be provided by arbitrary people. In order to guarantee quality of data in such an environment, an additional model regulating the publishing process is required, which prevents foreseeable misuses. To support this requirement the established portal architecture was extended with a workflow component which regulates the publishing process. In the following we will begin with introducing the concept of a publishing workflow in general. Afterwards we explain how we instantiated this generic component in OntoWeb.

A publishing workflow is the series of interactions that should happen to complete the task of publishing data. Business organizations have many kinds of workflow. Our notion of workflow is centered around tasks. Workflows consist of several tasks and several transitions between these tasks. Additionally, workflows have the following characteristics: (i) they might involve several people, (ii) they might take a long time, (iii) they vary significantly in organizations and in the computer applications supporting these organizations respectively, (iv) sometimes information must be kept across states, and last but not least, (v) the communication between people must be supported in order to facilitate decision making. Thus, a workflow component must be

customizable. It must support the assignment of tasks to (possibly multiple) individual users. In our architecture these users are grouped into roles. Tasks are represented within a workflow as a set of transitions which cause state changes. Each object in the system is assigned a state, which corresponds to the current position within the workflow and can be used to determine the possible transitions that can validly be applied to the object. This state is persistent supporting the second characteristic mentioned above. Due to the individuality of workflows within organizations and applications we propose a generic component that supports the creation and customization of several workflows. In fact, each concept in the ontology, which – as you might recall – is used to capture structured data within a portal, can be assigned a different workflow with different states, transitions and task assignments. As mentioned above, sometimes data is required to be kept across states<sup>4</sup>. To model this behavior, the state machine underlying our workflow model needs to keep information that “remembers” the past veto. Thus, variables are attached to objects and used to provide persistent information that transcends states. Within our approach, variables also serve the purpose of establishing a simple form of communication between the involved parties. Thus, each transition can attach comments to support the decision made by future actors. Also metadata like the time and initiator of a transition is kept within the system.

**Workflows in OntoWeb.** Figure 1.4 depicts the default workflow within OntoWeb. There are three states: private, pending, and published. In the private state the respective object is only visible to the user himself, the pending state makes it visible to reviewers. In the published state, a given object is visible to all (possibly anonymous) users of the portal. If a user creates a new object<sup>5</sup>, it is in private state. If the user has either a reviewer or a manager role the published state is immediately available through the publish transition. For normal users such a transition is not available. Instead, the object can only be sent for a review leading to the pending state. In the pending state either managers or reviewers can force the transition into the published state (by applying the transition “publish”) or retract the object leading back to the private state. The reject transition deletes the object completely. When an object is in the private state, only the user who created it and users with manager roles can view and change it. Once an object is in published state, the modification by the user who created it resets the object into pending state, thus the modification must be reviewed again. This does not apply to modifications by site managers.

<sup>4</sup> For example, envision the process of passing bills in legislature, a bill might be allowed to be revised and resubmitted once it is vetoed, but only if it has been vetoed once. If it is vetoed a second time, it is rejected forever.

<sup>5</sup> Currently only within the portal, the content syndicated from other OntoWeb member web sites and within the databases is “trusted”. We assume that this kind of data already went through some kind of review.

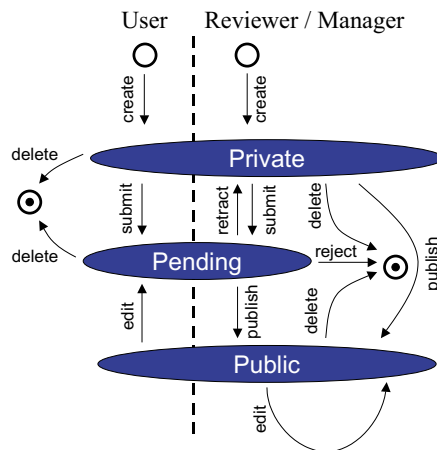


Fig. 1.4. OntoWeb Publishing workflow

## 1.7 Content Access

As has already been mentioned in section 1.2, the OntoWeb semantic portal offers an ontology-based browse and query facility. It has been developed as a highly generic system that allows to explore the available information at a conceptual level. Stated otherwise, the searches are performed on meta-data and not actual data. Currently, a (human) user can access information in the OntoWeb semantic portal in three different ways (see also [[1.8]:p.476]):

- *browsing*: a user doesn't know the vocabulary he needs to search with and/or is rather unfamiliar with the domain <sup>6</sup>
- *querying*: a user is quite familiar with the domain and its vocabulary
- a *combination* of the above: a user has some insights in the structure of the domain and is vaguely aware of the vocabulary (s)he needs to access the information (s)he needs

Any attempt to access information necessarily starts with an initial selection of a concept from the IsA hierarchy (displayed at the left hand side of Figure 1.5). As a result, a reduction of the search space is achieved. The system performs queries for content and schema in order to generate navigation views (see section 1.2). The main distinctions to keep in mind are the ones between sub- and superconcepts and between literal and non-literal properties of the various concepts or instances. Note that the user interface is still work in progress. E.g., a hyperbolic view [[1.8]:p.482] or a landscape view [1.28] would be an alternative way of displaying search results for subsequent selection. Also note that clicking the OntoWeb symbol (root of the IsA hierarchy) restarts the search process from scratch.

<sup>6</sup> See section 1.2 for a description of the OntoWeb domain.

### 1.7.1 Browsing

An ontology, as it is by definition a shared agreement on an intended conceptualization of a domain [1.4], represents how a (majority of members of a) user community "sees" the structure of an application domain <sup>7</sup>. Therefore, a visualization of the domain model can be considered as shared "mental roadmap" that helps users in locating and finding the desired information more rapidly. An expandable tree representation of the IsA hierarchy (see Figure 1.5), combined with an overview of semantic relationships for a selected concept instance, provides a local and partial view on the domain conceptualization. Note that the hierarchy supports multiple inheritance (nodes can have more than one parent in the "tree").

A user can view instances associated to a concept of the IsA hierarchy (= the tree in the left pane) by expanding its nodes and selecting the concept of interest. The instances of this concept will then be displayed (in the middle part of the right pane). Moving up or down the concept hierarchy corresponds to performing a generalized or specialized look-up of corresponding instances. By selecting a subtype of the current concept, the look-up precision should improve since the instances of the supertype (i.e. the concept originally selected), including all the instances of its subtypes that do not belong to the subtype newly selected, are excluded from the search space. A independent (but partial) validation of this hypothesis can be found in [1.19].

Generalization (i.e. moving up one level in the hierarchy or clicking on the supertype displayed) on the other hand broadens the scope of the query, exploiting the concept hierarchy to expand the query to all instances of the siblings (and their subtypes) of the concept originally of interest to the user (cf. also [1.12]).

### 1.7.2 Typed term-based Querying

A user is presented with a search form (see the lower part of Figure 1.5), containing text boxes in which attribute values (literals) can be specified. In addition, additional buttons labelled with a concept name allow to refine the query by imposing restrictions on related concept instances. Clicking on a such "concept button" leads to another form that allows to specify related instances (= entering new attribute values) of the concept newly selected (i.e. the one now shown in bold on the titlebar of the form). There are as many buttons as there are relationships associated with the concept originally selected. This means that the forms are generated dynamically on basis of the underlying ontology. The terms are typed - but no explicit type-checking is done yet - since each attribute has a meaningful range (e.g., it doesn't

---

<sup>7</sup> Note that an application domain transcends a data model for a single application - see [1.29] for more details



Fig. 1.5. Ontology-based searching for information

make sense to enter a date in the address attribute field). The range for the associated relationships is restricted to a single concept, namely the one of which the label is displayed on the corresponding button. One can enter attribute values for several cascaded forms.

The form-based specifies a query path across the ontology (displayed in the titlebar of the form). For each node in the path, a user can add restrictions on the attribute values. All the attributes values filled in in the various forms constitute a complex query on the instance base. This can be considered as a form of interactive query refinement. However, this kind of query refinement is guided by the ontology before the actual look-up process is activated and not on basis of (intermediary) search results as it is usually the case. The hypothesis is that recall should improve since underlying semantic properties are taken into account instead of only the formal appearance of a character string. Also the precision should improve as the semantic properties have a higher discriminative power (compared to pattern matching that is the basic traditional search mechanism) to rule out non relevant search results.

### 1.7.3 Combining browsing and free term-based Querying

The portal also offers a keyword based search on attribute values of instances without the need to specify the attribute. A user may opt at any moment to enter one or more search terms in the search box that is displayed at the top of the page. When a query is executed, the search space only includes instances

of the concept selected earlier (and its subtypes). This search strategy is useful when a user only has a rough idea of what (s)he is looking for. Some of the characteristics (= attributes) of the item to be looked for are known but exactly how these characteristics relate to the item being searched for (in opposition to form based querying) are not known to an end-user.

The instances retrieved are presented to the user grouped by links pointing to the instance details page. The right hand side of the screen now displays the most specific concepts corresponding to instances that match the user query (see Figure 1.6). Clicking on a concept in the "tree" equals to selecting a particular view on the results <sup>8</sup>.

**OntoWeb**

- ★ ALL(8)
- ★ Organization(1)
- ★ FullProfessor(1)
- ★ ResearchGroup(1)
- ★ Institute(1)
- ★ Employee(4)

**Search results:**  
If your search did not return the results you are looking for, you can also search document contents.

**Group>>Organization>>**  
Hoofdstedelijke Openbare Bibliotheek - Brussel

**Person>>Employee>>AcademicStaff>>FacultyMember>>FullProfessor>>**  
Robert Meersman

**Group>>Organization>>ResearchGroup>>**  
Systems Technology and Applications Research Laboratory

**Group>>Organization>>Institute>>**  
the Vlaamse Gemeenschapscommissie

**Person>>Employee>>**  
Ben Majer

**Person>>Employee>>**  
Jan De Bo

**Person>>Employee>>**  
Jarrar, Mustafa

**Person>>Employee>>**  
Zheng Jijuan

**Fig. 1.6.** search results for "STARLab" combined with the "Person" concept

When an end-user enters multiple keywords, the engine searches for conceptual paths between instances that have these keywords as their attribute values and, if found, displays paths and the related instances (see Figure 1.7). Notice that this particular feature enables a user to discriminate between meaningful and meaningless combinations, and in addition, helps him/her to select these meaningful combinations that are relevant for him/her. The strength of a semantically-based search engine is fully exploited and valorized in this situation. Traditional search engines, lacking underlying semantics, simply cannot offer a similar powerful feature.

<sup>8</sup> Of course, "ALL" is not a concept, but merely represents a exhaustive view of the result list.

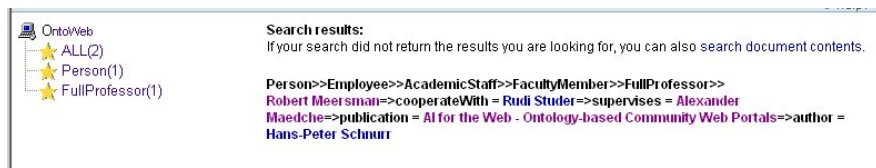


Fig. 1.7. search results for "Robert Peter" combined with the "Person" concept

## 1.8 Content Presentation

What strategy (as has been described in the previous section) a user has applied to specify his/her search request, eventually (s)he selects a particular instance. When displaying the detailed information for that instance, a distinction is made between the attributes and relationships. Attribute values - modelled in the ontology as literal properties - provide a user with specific instance information, while "relationship values" are shown as hyperlinks, enabling a user to jump to instances of related concepts. The distinction between attributes and relationships is decided by the ontology modeler<sup>9</sup>.

Attributes are displayed at the upper part of the page. These concern e.g., in the case of a person, the name, telephone number and email... An overview of all the relevant conceptual relationships is displayed in the middle part of the page. The "relationship values" are presented at the lower part of the page grouped by relationship. They point to instances of related concepts (cf. Figure 1.8). Again, the screen is dynamically generated: only those attributes and relationships are shown for which instance data is stored in the instance base. Whenever relevant (depending on the ontology) and/or applicable (depending on the instance data), also a URL is displayed that brings the user to the web site that contains the original data. Remember that the portal basically contains meta-data (except for the content object as described in section 1.6.2) that has been crawled by the syndicator (see section 1.6.1).

## 1.9 Related work

The SEAL approach offers a comprehensive conceptual framework for Web information integration and Web site management. The ontological foundation of SEAL is the main distinguishing feature when comparing SEAL with approaches from the information systems community.

Using an ontology to support the access of content has been discussed before. E.g., the so-called Yahoo-a-lizer [1.16] transforms a knowledge base

<sup>9</sup> The range of an attribute is "STRING", while the range of a relationship is a concept of the domain ontology.



**Concept Instance label****FullProfessor : Robert Meersman**

<b>Attributes</b>	
address	Department of Computer Science, Vrije Universiteit Brussel Bldg. G-10, Pleinlaan 2, B-1050 Brussels Belgium
email	robert.meersman@vub.ac.be
fax	++32-(0)2-629-3525
name	Robert Meersman
phone	++32-(0)2-629-3308
photo	<a href="http://www.starlab.vub.ac.be/images/robert.gif">http://www.starlab.vub.ac.be/images/robert.gif</a>
<b>Attribute values (literals)</b>	
<div style="display: flex; justify-content: space-between;"> <span>affiliation</span> <span>cooperateWith</span> <span>headOf</span> <span>headOfGroup</span> <span>organizerOrChairOf</span> </div>	
<b>Relationships (overview)</b>	
<b>Relationships</b>	
affiliation	<a href="#">Systems Technology and Applications Research Laboratory</a>
cooperateWith	<a href="#">Rudi Studer</a>
headOf	<a href="#">DOGMA</a>
headOfGroup	<a href="#">Systems Technology and Applications Research Laboratory</a>
organizerOrChairOf	<ul style="list-style-type: none"> <li>▶ <a href="#">DOA</a></li> <li>▶ <a href="#">IFCIS</a></li> </ul>
supervises	<ul style="list-style-type: none"> <li>▶ <a href="#">Ben Majer</a></li> <li>▶ <a href="#">Jarrar Mustafa</a></li> <li>▶ <a href="#">Zheng Jijuan</a></li> </ul>
worksAtProject	<a href="#">DOGMA</a>
<b>Instances (hyperlinks)</b>	
<b>Instance URI</b>	
URI-reference :	<a href="http://starpc14.vub.ac.be:8000/OntoWeb/Browse/allInstances#RM">http://starpc14.vub.ac.be:8000/OntoWeb/Browse/allInstances#RM</a>

Fig. 1.8. Content presentation

into a set of XML pages that are structured like the term hierarchy of Yahoo. These XML-files are translated via an XSL-stylesheet into ordinary HTML. Within Ontobroker-based web portals [1.15], a Hyperbolic View Applet allows for graphical access to an ontology and its knowledge base. Another related work is KAON Portal<sup>10</sup> which takes an ontology and creates a standard Web interface out of it. The OntoSeek [1.19] prototype uses a linguistic ontology and structured content representations to search yellow pages and product catalogs. The importance of conceptual indexing for information retrieval has been acknowledged since quite some time in the medical information processing field [1.17, 1.27, 1.32]. However, from our point of view the

<sup>10</sup> cf. <http://kaon.semanticweb.org/Portal>

OntoWeb portal is rather unique with respect to the collection of methods used and the functionality provided.

Given the difficulties with managing complex Web content, several papers tried to facilitate database technology to simplify the creation and maintenance of data-intensive web-sites. Systems such as ARANEUS [1.24] and AutoWeb [1.13], take a declarative approach, i.e. they introduce their own data models and query languages, although all approaches share the idea to provide high-level descriptions of web-sites by distinct orthogonal dimensions. The idea of leveraging mediation technologies for the acquisition of data is also found in approaches like Strudel [1.18] and Tiramisu [1.11], they propose a separation according to the aforementioned task profiles as well. Strudel does not concern the aspects of site maintenance and personalization. It is actually only an implementation tool, not a management system. It provides a clear separation of three tasks that are important for building up a data-intensive Web site: (i) accessing and integrating the data available in the Web site, (ii) building up the structure and content of the site, and (iii) generating the HTML representation of the site pages. Basically, STRUDEL relies on a mediator architecture where the semi-structured OEM data model is used at the mediation level to provide a homogeneous view on the underlying data sources. STRUDEL then uses so-called 'site definition queries' to specify the structure and content of a Web site. When compared to our SEAL approach STRUDEL lacks the semantic level that is defined by the ontology. Furthermore, within SEAL the ontology offers a rich conceptual view on the underlying sources that is shared by the Web site users and that is made accessible at the user interface for e.g. browsing and querying.

The Web Modelling Language WebML [1.13] provides means for specifying complex Web sites on a conceptual level. Aspects that are covered by WebML are a.o. descriptions of the site content, the layout and navigation structure as well as personalization features. Thus, WebML addresses functionalities that are offered by the presentation and selection layer of the SEAL conceptual architecture. Whereas WebML provides more sophisticated means for e.g. specifying the navigation structure, SEAL offers more powerful means for accessing the content of the Web site, e.g. by semantic querying.

## 1.10 Future Work and Conclusion

For the future, we see a number of new important topics appearing on the horizon. For instance, we consider approaches for ontology learning in order to semi-automatically adapt to changes in the world and to facilitate the engineering of ontologies. Currently, we work on providing intelligent means for providing semantic information, *i.e.* we elaborate on a semantic annotation framework that balances between manual provisioning from legacy texts (*e.g.* web pages) and information extraction. Finally, we envision that once semantic web sites are widely available, their automatic exploitation may be

brought to new levels. Semantic web mining considers the level of mining web site structures, web site content, and web site usage on a semantic rather than at a syntactic level yielding new possibilities, *e.g.* for intelligent navigation, personalization, or summarization, to name but a few objectives for semantic web sites.

A next important step to take is to enter a significantly large amount of real life data in the instance base so that a truly useful knowledge base is created. Before doing that, an update of the ontology is foreseen as well. As a general consideration, the user interface will be refined as well. After these steps, a large-scale assessment on the strengths and flaws (also as perceived by end-users) of the OntoWeb portal becomes possible.

In this article we have shown the combination of two frameworks (SEAL and DOGMA) for building a knowledge portal. In particular, we have focused on three issues. First, we have described the general architecture of the SEAL framework. Second, we have presented our real world case study, the OntoWeb portal. Third, to meet the requirements of the OntoWeb portal, we extended our initial conceptual architecture by publishing workflows to make user focussed access to the portal maintainable. In addition, we created a specific semantically driven user interface to improve the information retrieval process.

## Acknowledgements

We want to thank several colleagues at the Free University of Brussels - STAR Lab, headed by Robert Meersman, and at the University of Karlsruhe - Institute AIFB, headed by Rudi Studer, for their contribution to this work. This includes York Sure, Raphael Volz, Steffen Staab and Jens Hartmann at AIFB as well as Jijuan Zheng, Mustafa Jarrar and Ben Majer for STAR Lab. This work has been funded under the EU IST-2001-29243 project "OntoWeb".

## References

- 1.1 M. Crampes and S. Ranwez. Ontology-supported and ontology-driven conceptual navigation on the world wide web. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia, May 30 - June 3, 2000, San Antonio, TX, USA*, pages 191–199. ACM Press, 2000.
- 1.2 D. Fensel, J. Angele, S. Decker, M. Erdmann, H.-P. Schnurr, R. Studer, and A. Witt. Lessons learned from applying AI to the web. *International Journal of Cooperative Information Systems*, 9(4):361–382, 2000.
- 1.3 E. Grosso, H. Eriksson, R. W. Fergerson, S. W. Tu, and M. M. Musen. Knowledge modeling at the millennium: the design and evolution of PROTEGE-2000. In *Proceedings of the 12th International Workshop on Knowledge Acquisition, Modeling and Management (KAW-99)*, Banff, Canada, October 1999.

- 1.4 N. Guarino, and P. Giaretta, (1995). Ontologies and Knowledge Bases: Towards a Terminological Clarification. N. Mars (editor), *Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press, Amsterdam, 1995.
- 1.5 A. Hotho, A. Maedche, S. Staab, and R. Studer. SEAL-II - The soft spot between richly structured and unstructured knowledge. *Universal Computer Science (J.UCS)*, 7(7):566–590, 2001.
- 1.6 O. Lassila and R. Swick. Resource Description Framework (RDF). Model and syntax specification. Technical report, W3C, 1999. <http://www.w3.org/TR/REC-rdf-syntax>.
- 1.7 Y. Papakonstantinou, H. Garcia-Molina, and J. Widom. Object exchange across heterogeneous information sources. In *Proceedings of the IEEE International Conference on Data Engineering, Taipei, Taiwan, March 1995*, pages 251–260, 1995.
- 1.8 S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. In *WWW9 / Computer Networks (Special Issue: WWW9 - Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands, May, 15-19, 2000)*, volume 33, pages 473–491. Elsevier, 2000.
- 1.9 S. Staab and A. Maedche. Knowledge portals - ontologies at work. *AI Magazine*, 21(2), 2001.
- 1.10 G. Wiederhold and M. Genesereth. The conceptual basis for mediation services. *IEEE Expert*, 12(5):38–47, Sep.-Oct. 1997.
- 1.11 C. R. Anderson, A. Y. Levy, and D. S. Weld. Declarative web site management with Tiramisu. In *ACM SIGMOD Workshop on the Web and Databases - WebDB99*, pages 19–24, 1999.
- 1.12 A. Aronson and T. Rindflesch. Query expansion using the UMLS. In R. Masys, editor, *Proceedings of the AMIA Annual Fall Symposium 97, JAMIA Suppl*, pages 485–489. AMIA, 1997.
- 1.13 S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (WebML): a modeling language for designing web sites. In *WWW9 Conference, Amsterdam, May 2000*, 2000.
- 1.14 OntoWeb Consortium. Ontology-based information exchange for knowledge management and electronic commerce - IST-2000-25056. <http://www-ontoweb.org>, 2001.
- 1.15 S. Decker, M. Erdmann, D. Fensel, and R. Studer. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. In R. Meersman et al., editors, *Database Semantics: Semantic Issues in Multimedia Systems*, pages 351–369. Kluwer Academic Publisher, 1999.
- 1.16 Michael Erdmann. *Ontologien zur konzeptuellen Modellierung der Semantik von XML*. Isbn: 3831126356, University of Karlsruhe, 10 2001.
- 1.17 D. Evans, D. Rothwell, I. Monarch, R. Lefferts, and R. Côté. Towards representations for medical concepts. *Medical Decision Making*, 11 (supplement):S102–S108, 1991.
- 1.18 M. F. Fernandez, D. Florescu, A. Y. Levy, and D. Suciu. Declarative specification of web sites with Strudel. *VLDB Journal*, 9(1):38–55, 2000.
- 1.19 N. Guarino, C. Masolo, and G. Vetere. Ontoseek: Content-based access to the web. *IEEE Intelligent Systems*, May-June4-5:70–80, 1999.
- 1.20 S. Handschuh and S. Staab. Authoring and annotation of web pages in CREAM. In *The Eleventh International World Wide Web Conference (WWW2002), Honolulu, Hawaii, USA 7-11 May, 2002*. To appear.

- 1.21 M. Jarrar and R. Meersman. Formal Ontology Engineering in the DOGMA Approach. In R. Meersman, Z. Tari and al., editors, *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE; Confederated International Conferences CoopIS, DOA, and ODBASE 2002 Proceedings*, pages 1238–1254. LNCS 2519, Springer Verlag, 2002.
- 1.22 H. Lowe, I. Antipov, W. Hersh, and C. Arnott Smith. Towards knowledge-based retrieval of medical images. the role of semantic indexing, image content representation and knowledge-based retrieval. In C. Chute, editor, *Proceedings of the 1998 AMIA Annual Fall Symposium*, pages 882–886. AMIA, Henley & Belfus, Philadelphia, 1998.
- 1.23 A. Maedche, S. Staab, R. Studer, Y. Sure, and R. Volz. Seal — tying up information integration and web site management by ontologies. *IEEE Data Engineering Bulletin*, 25(1):10–17, March 2002.
- 1.24 G. Mecca, P. Merialdo, P. Atzeni, and V. Crescenzi. The (short) Araneus guide to web-site development. In *Second Intern. Workshop on the Web and Databases (WebDB'99) in conjunction with SIGMOD'99*, May 1999.
- 1.25 R. Meersman. Semantic Web and Ontologies: Playtime or Business at the Last Frontier in Computing ? NSF-EU Workshop on Database and Information Systems Research for Semantic Web and Enterprises. pages 61–67, 2002.
- 1.26 R. Meersman and M. Jarrar. Scalability and reusable in ontology modeling. In *Proceedings of the International conference on Infrastructure for e-Business, e-Education, e-Science, and e-Medicine (SSGRR2002s)*, 2002. (only available on CD-ROM).
- 1.27 T. Rindflesch and A. Aronson. Semantic processing in information retrieval. In C. Safran, editor, *Seventeenth Annual Symposium on Computer Applications in Medical Care (SCAMC 93)*, pages 611–615. McGraw-Hill Inc., New York, 1993.
- 1.28 V. Sabol, W. Kienreich, M. Granitzer, J. Becker, K. Tochtermann, K. Andrews. Applications of a Lightweight, Web-based Retrieval, Clustering, and Visualisation Framework. In D. Karagiannis and U. Reimer (editors), *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM02)*. pages 359–369. LNAI 2569, Springer Verlag, 2002
- 1.29 P. Spyns, R. Meersman and J. Jarrar. Data modelling versus Ontology engineering. In A. Sheth and R. Meersman (editors), *SIGMOD Record Special Issue on Semantic Web, Database Management and Information Systems*, [forthcoming].
- 1.30 P. Spyns, D. Oberle, R. Volz, J. Zheng, M. Jarrar, Y. Sure, R. Studer, R. Meersman. OntoWeb - A Semantic Web Community Portal. In D. Karagiannis and U. Reimer (editors), *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management (PAKM02)*. pages 189–200. LNAI 2569, Springer Verlag, 2002
- 1.31 S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. *Dublin Core Metadata for Resource Discovery*. Number 2413 in IETF. The Internet Society, September 1998.
- 1.32 P. Zweigenbaum, J. Bouaud, B. Bachimont, J. Charlet, B. Séroussi, and J.-F. Boisvieux. From text to knowledge: a unifying document-oriented view of analyzed medical language. *Methods of Information in Medicine*, 37(4-5):384–393, 1998.