SEAL — A Framework for Developing SEmantic PortALs

Nenad Stojanovic
Institute AIFB
University of Karlsruhe
D-76128 Karlsruhe, Germany
nst@aifb.uni-karlsruhe.de

Alexander Maedche
FZI Research Center for Information Technologies
Haid-und-Neu Straße 10-14
D- 76131 Karlsruhe, Germany
maedche@fzi.de

Steffen Staab♣, Rudi Studer♦, York Sure
Institute AIFB
University of Karlsruhe
D-76128 Karlsruhe, Germany
staab,studer,sure@aifb.uni-karlsruhe.de

Abstract

The core idea of the Semantic Web is to make information accessible to human and software agents on a semantic basis. Hence, Web sites may feed directly from the Semantic Web exploiting the underlying structures for human and machine access. We have developed a domain-independent approach for developing semantic portals, *viz.* SEAL (SEmantic portAL), that exploits semantics for providing and accessing information at a portal as well as constructing and maintaining the portal. In this paper we focus on semantics-based means that make semantic Web sites accessible from the outside, i.e. semantics-based browsing, semantic querying, querying with semantic similarity, and machine access to semantic information. In particular, we focus on methods for acquiring and structuring community information as well as methods for sharing information.

As a case study we refer to the AIFB portal — a place that is increasingly driven by Semantic Web technologies. We also discuss lessons learned from the ontology development of the AIFB portal.

Keywords

Ontology, Knowledge portal, Semantic Web

INTRODUCTION

The widely-agreed core idea of the Semantic Web is the delivery of data on a semantic basis. Intuitively the delivery of semantically processable data should help with establishing a higher quality of communication between the information provider and the consumer. The vison of the Semantic Web is closely related to ontologies as a sound semantic basis that is used to define the meaning of terms and hence to support intelligent providing and access to information on the Web.

♣ Also Ontoprise GmbH, Haid-und-Neu Str. 7, D-76131 Karlsruhe

♦ Also Ontoprise GmbH and FZI Research Center for Information Technologies

K-CAP'01, October 22-23, 2001, Victoria, British Columbia, Canada.

The topic of this paper is a framework for developing ontology-based portal applications, namely SEAL (SEmantic PortAL) and its semantic mechanism for acquiring, structuring and sharing community information between human and/or machine agents. Ontologies constitute the foundation of our SEAL approach. The origins of SEAL lie in Ontobroker [3], which was conceived for semantic search of knowledge on the Web and also used for sharing knowledge on the Web [2]. It then developed into an overarching framework for search and presentation offering access at a portal site [17]. This concept was then transferred to further applications [1] and is currently extended into a commercial solution (*cf.* http://www.time2research.de). We here describe the SEAL core modules and its overall architecture (Section *SEAL Infrastructure and core modules*). As a case study we refer to the AIFB portal (*cf.* http://www.aifb.uni-karlsruhe.de). Thereafter, we go into several technical details that are important for human and machine access to a semantic portal.

In particular, we describe a general approach for semantic ranking (Section *Semantic Ranking*). The motivation for semantic ranking is that even with accurate semantic access, one will often find too much information. Underlying semantic structures, *e.g.* topic hierarchies, give an indication of what should be ranked higher on a list of results. Also, we present mechanisms to deliver and collect machine-understandable data (Section *RDF Outside*) and discuss how this approach establishes the road to the Semantic Web. These mechanisms extend previous means for better digestion of Web site data by software agents. Finally, we describe some experiences made during the development of the ontology for our AIFB portal (Section *Experience with ontology engineering*). Before we conclude, we give a short survey of related work.

SEAL INFRASTRUCTURE AND CORE MODULES

In this section, we first elaborate on the general architecture for SEAL (SEmantic PortAL) and then we explain functionalities of its core modules. As a running example we refer to the AIFB portal, which aims at presenting information to human and software agents taking advantage of semantic structures.

Architecture

The overall architecture and environment of SEAL is depicted in Figure 1. The *backbone* of the system consists of the *knowledge warehouse*, *i.e.* the ontology and knowledge base, and the *Ontobroker* system [3], *i.e.* the principal inferencing mechanism. The latter functions as a kind of middleware run-time system, possibly mediating between different information sources when the environment becomes more complex than it is now.

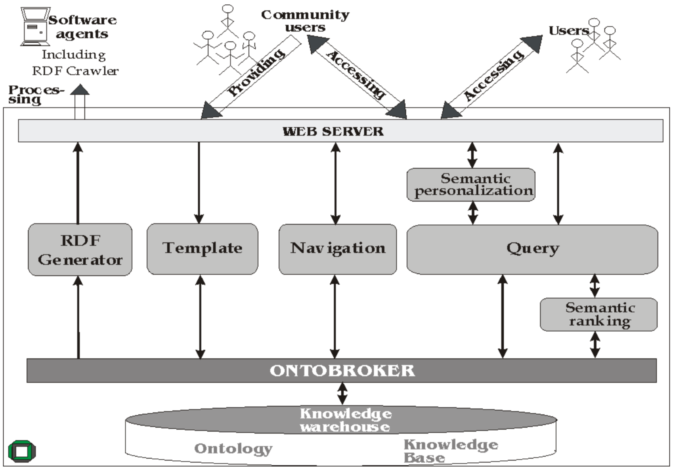

Figure 1: SEAL - System architecture

At the front end one may distinguish between three types of *agents*: *software agents*, *community users* and *general users*. All three communicate with the system through the *Web server*. The three different types of agents correspond to three primary modes of interaction with the system.

First, remote applications *(e.g.* software agents) may process information stored in the portal. For this purpose, the *RDF generator* presents RDF facts through the Web server. Software agents with *RDF crawlers* may collect the facts and, thus, have direct access to semantic knowledge stored at the Web site.

Second, community users and general users can access information contained at the Web site. Two forms of accessing are supported: navigating through the portal by exploiting hyperlink structure of documents and searching for information by posting queries. The hyperlink structure is partially given by the portal builder, but it may be extended with the help of the *navigation* module. The navigation module exploits inferencing capabilities of the inference engine in order to construct conceptual hyperlink structures. Searching and querying is performed via the *query* module. In addition, the user can personalise the search interface using the *semantic personalization* module and/or rank retrieved results according to semantic similarity (done by the module for *semantic ranking*). Queries also take advantage of the Ontobroker inferencing capabilities.

Third, only community users can provide data. In the AIFB portal application, typical information community user contribute include personal data, information about research areas, publications and other research information. For each type of information they may contribute there is (at least) one concept in the ontology. By retrieving parts of the ontology, the *template* module may semi-automatically produce suitable HTML forms for data input. The community users fill in these forms and the template module stores the data in the knowledge warehouse.

Core modules

The core modules have been extensively described in [17]. In order to give the reader a compact overview we here shortly survey their function. In the remainder of the paper we delve deeper into those aspects that have been added or considerably extended recently, *viz.* semantic ranking (Section *Semantic Ranking*), and semantic access by software agents (Section *RDF Outside*).

Ontobroker

The Ontobroker system [3] is a deductive, object-oriented database system operating either in main memory or on a relational database (via JDBC). It provides compilers for different languages to describe ontologies, rules and facts. Beside other usage, it is also used as an inference engine (server) within SEAL. It reads input files containing the knowledge base and the ontology, evaluates incoming queries, and returns the results derived from the combination of ontology, knowledge base and query. The possibility to derive additional factual knowledge from given facts and background knowledge considerably facilitates the life of the knowledge providers and the knowledge seekers. For instance, one may specify that if a person belongs to a research group of the institute AIFB, he also belongs to AIFB. Thus, it is unnecessary to specify the membership to a research group *and* to AIFB. Conversely, the info seeker does not have to take care of inconsistent assignments, *e.g.* ones that specify membership to an AIFB research group, but that have erroneously left out the membership to AIFB.

Knowledge warehouse

The knowledge warehouse [17] serves as repository for data represented in the form of F-Logic statements [6]. It hosts the ontology, as well as the data proper. From the point of view of inferencing the difference is negligible, but from the point of view of maintaining the system the difference between ontology definition and its instantiation is useful. The knowledge warehouse is organised around a relational database, where facts and concepts are stored in a reified format. It states relations and concepts as first-order objects and it is therefore very flexible with regard to changes and amendments of the ontology.

Navigation module

Beside the hierarchical, tree-based hyperlink structure which corresponds to the hierarchical decomposition of the domain, the navigation module enables complex graph-based semantic hyperlinking, based on ontological relations between concepts (nodes) in the domain. The conceptual approach to hyperlinking is based on the assumption that semantically relevant hyperlinks from a Web page correspond to conceptual relations, such as `memberOf` or `hasPart`, or to attributes, like `hasName`. Thus, instances in the knowledge base may be presented by automatically generating links to all related instances. For example, on personal Web pages there are, among others, hyperlinks to Web pages that describe the corresponding research groups, secretary and professional activities (*cf.* Figure 2, higher part).

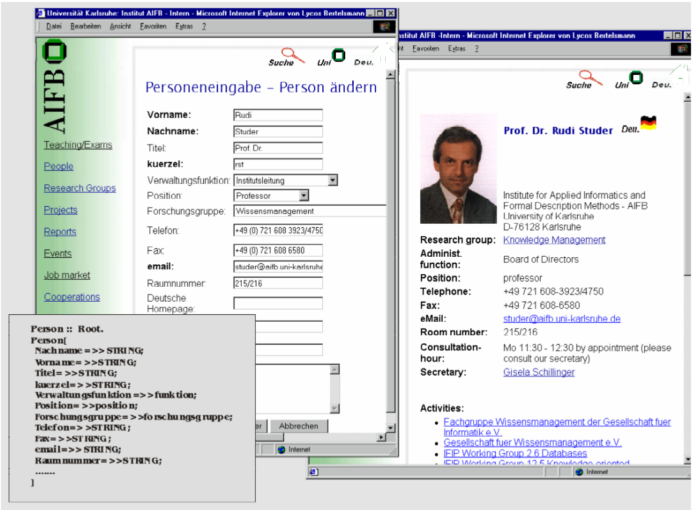

Figure 2: Templates generated from concept definitions

Query module

The query module puts an easy-to-use interface on the query capabilities of the F-Logic query interface of Ontobroker. The portal builder models Web pages that serve particular query needs. For this purpose, selection lists that restrict query possibilities are offered to the user.

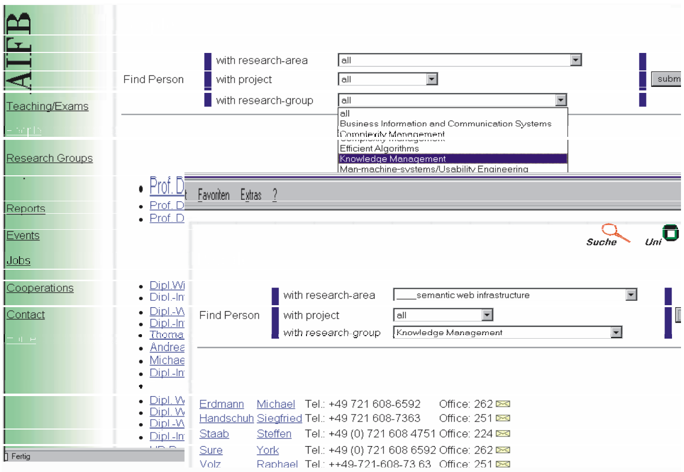

Figure 3: Query form based on definition of concept Person

The selection lists are compiled using knowledge from the ontology and/or the knowledge base. For instance, the query interface for persons on the AIFB portal, allows to search for people according to research groups they are members of. The list of research groups is dynamically filled by an F-Logic query and presented to the user for easy choice by a drop-down list (*cf.* snapshot in Figure 3). Even simpler, one may associate a hyperlink with an F-Logic query that is dynamically evaluated when the link is hit. More complex, one may construct an `isA`, a `hasPart`,or a `hasSubtopic` tree, from which query events are triggered when particular nodes in the tree are selected.

Template module

In order to facilitate the contribution of information by community users, the template module generates an HTML form for each concept that a user may instantiate. For instance, the AIFB portal includes an input template (*cf.* Figure 2, left upper part) generated from the concept definition of person (*cf.* Figure 2, lower left). The data is later on used by the navigation module to produce the corresponding person Web page (*cf.* Figure 2, right part). In order to reduce the data required for input, the portal builder specifies which attributes and relations are derived from other templates. For example, in our case the portal builder has specified that project membership is defined in the project template. The coordinator of a project enters information which persons are participants of the project and this info is used when generating the person Web page taking advantage of a corresponding inverse relationship, between relations `WorksIn` and `memberOf`.

Ontology lexicon

The different modules described here make extensive use of the lexicon component of the ontology (*cf.* *Section Experience with ontology engineering*). The most prevalent use is the distinction between English and German. In the future we envision that one may produce more adaptive Web sites making use of the explicit lexicon. For instance, we will be able to produce short descriptions when the context is sufficiently narrow, *e.g.* working with ambiguous acronyms like ASP (e.g. active server pages *vs.* active service providers).

SEMANTIC RANKING

This section describes the architecture component „Semantic Ranking“ which has been developed in the context of our framework. First, we will introduce and motivate the requirement for a ranking approach with a small example. Second, we will show how the problem of semantic ranking may be reduced to the comparison of two knowledge bases. Query results are reinterpreted as „query knowledge bases“ and their similarity to the original knowledge base without axioms yields the basis for semantic ranking. Thereby, we reduce our notion of similarity between two knowledge bases to the similarity of concept pairs.

Let us assume the following ontology:

```
1:  Person::Object [worksIn =>> Project].
2:  Project::Object[hasTopic =>> Topic].
3:  Topic::Object[subtopicOf =>> Topic].      (1)
4:  FORALL X,Y,Z   Z[hasTopic ->>Y] <-
    X[subtopicOf ->>Y] and Z[hasTopic ->>X].
```

To give an intuition of the semantic of the F-Logic statements, in line 1 one finds a concept definition for a `Person` being an `Object` with a relation `worksIn`. The range of the relation is restricted to `Project`.

Ontology axioms like the one given in line 4 (1) use this syntax to describe regularities. Line 4 states that if some `Z` has topic `X` and `X` is a subtopic of `Y` then `Z` also has topic `Y`. Let us further assume the following knowledge base:

```
5:  KnowledgeManagement:Topic.
6:  KnowledgeDiscovery:
    Topic[subtopicOf ->>KnowledgeManagement].
7:  Gerd:Person[worksIn ->>OntoWise].          (2)
8:  OntoWise:
    Project[hasTopic ->>KnowledgeManagement].
9:  Andreas:Person[worksIn ->>TelekomProject].
10: TelekomProject:
    Project[hasTopic ->>KnowledgeDiscovery].
```

Definitions of instances in the knowledge base are syntactically very similar to the concept definition in F-Logic. In line 6 the instance `KnowledgeDiscovery` of the concept `Topic` is defined. Furthermore, the relation `subtopicOf` is instantiated between `KnowledgeDiscovery` and `KnowledgeManagement`. Similarly in line 7, it is stated that `Gerd` is a `Person` working in the project `OntoWise`.

Now, an F-Logic query may ask for all people who work in a knowledge management project by:

```
FORALL Y,Z <- Y[worksIn ->> Z] and          (3)
Z:Project[hasTopic ->> KnowledgeManagement]
```

which may result in the tuples `(Gerd, OntoWise)` and `(Andres, TelekomProjet)`. Obviously, both answers are correct with regard to the given knowledge base and ontology, but the question is, what would be a plausible ranking for the correct answers. This ranking should be produced from a given query without assuming any modification of the query.

Reinterpreting queries

Our principal consideration builds on the definition of semantic similarity that we have first described in [10]. There, we have developed a measure for the similarity of two knowledge bases. Here, our basic idea is to reinterprete possible query results as a „query knowledge base" and compute its similarity to the original knowledge base while abstracting from semantic inferences. The result of an F-Logic query may be reinterpreted as a *query knowledge base* (***QKB***) by the following approach.

An F-Logic query is of the form or can be rewritten into the form (*cf.* negation requires special treatment):

$$\text{FORALL } \overline{X} \leftarrow \overline{P}(\overline{X},\overline{k}) \qquad (4)$$

With $\overline{X}$ being a vector of variables $(X_1,...,X_n)$, $\overline{k}$ being a vector of constants, and $\overline{P}$ being a vector of conjoined predicates. The result of a query is a two-dimensional matrix M of size *mxn*, with *n* being the number of result tuples and *m* being the length of $\overline{X}$ and, hence, the length of the result tuples. Hence, in our example above: $\overline{X}$:= *(Y, Z)*, $\overline{k}$:= ("`KnowledgeManagement`"), $\overline{P}$:= (P_1, P_2), $P_1(a,b,c)$:= a [`worksIn ->>` b], $P_2(a,b,c)$:= b [`hasTopic ->>` c] and

$$M := (M_1, M_2) = \begin{pmatrix} \text{Gerd} & \text{Andreas} \\ \text{OntoWise} & \text{TelekomProjekt} \end{pmatrix}. \qquad (5)$$

Now, we may define the query knowledge base $i(QKB_i)$ by $QKB_i := \overline{P}(M_i, \overline{k})$. (6)

Similarity of knowledge bases

The similarity between two objects (concepts and or instances) may be computed by considering their relative place in a common hierarchy H. H may, but need not be a taxonomy H. For instance, in above example we have a categorization of research topics, which is not a taxonomy.

Our principal measures are based on the cotopies of the corresponding objects as defined by a given hierarchy H, *e.g.* an ISA hierarchy H, a part-whole hierarchy, or a categorization of topics. Here, we use the *upwards cotopy* (*UC*) defined as follows:

$$UC(O_i,H) := \{O_j \mid H(O_i,O_j) \vee O_j = O_i\} \qquad (7)$$

Concepts are taxonomically related by the irreflexive, acyclic, transitive relation $H, (H \subset C \times C)$. $H(C_1,C_2)$ means that C_1 is a subconcept of C_2. *UC* is overloaded in order to allow for a set of objects M as input, *viz.*

$$UC(M,H) := \bigcup_{O_i \in M} \{O_j \mid H(O_i,O_j) \vee O_j = O_i\} \qquad (8)$$

Based on the definition of the upwards cotopy (*UC*) the object match (*OM*) is defined by:

$$OM(O_1,O_2,H) := \frac{|UC(O_1,H) \cap UC(O_2,H)|}{|UC(O_1,H) \cup UC(O_2,H)|} \qquad (9)$$

Basically, *OM* reaches when two concepts coincide (number of intersections of the respective upwards cotopies and number of unions of the respective cotopies is equal); it degrades to the extent to which the discrepancy between intersections and unions increases (a *OM* between concepts that do not share common super-concepts yields value 0).

The match introduced above may easily be generalized to relations using a relation hierarchy H_R into account. Hence, it may also be generalized to instantiated relations. Thus, the predicate match (PM) for two n-ary predicates P_1, P_2 is defined by a mean value. Thereby, we use the geometric mean in order to reflect the intuition that if the similarity of one of the components approaches 0 the overall similarity between two predicates should approach 0 as well — which need not be the case for the arithmetic mean:

$$PM(P_1(I_1,...,I_n), P_2(J_1,J_n)) := \qquad (10)$$

$$\sqrt[n+1]{OM(P_1,P_2,H_R) \cdot OM(I_1,J_1,H) \cdot ... \cdot OM(I_n,J_n,H)}$$

This result may be averaged over an array of predicates. We here simply give the formula for our actual needs, where a query knowledge base is compared against a given knowledge base KB:

$$Simil(QKB_i, KB) = Simil(\overline{P}(M_i,\overline{k}), KB) := \qquad (11)$$

$$\frac{1}{|\bar{P}|}\sum_{P_j\in\bar{P}} \max_{Q(M_i,\bar{k})\in KB} PM(P_j(M_i,k),Q(M_i,\bar{k})).$$

Example. We here give a small example for computing UC and OM based on a given categorization of objects H. Figure 4 depicts the example scenario.

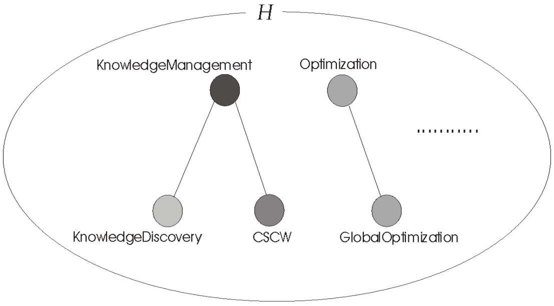

Figure 4: Example for computing UC and OM

The upwards cotopy UC(`KnowledgeDiscovery`, H) is given by {`KnowledgeDiscovery`, `KnowledgeManagement`}. The upwards cotopy UC(`Optimization`, H) computes to {`Optimization`}. Computing the object match OM between `KnowledgeManagement` and `Optimization` results in 0, the object match between `CSCW` and `KnowledgeDiscovery` computes to 1/3 .

For instance, we compare the two result tuples from our example above with the given knowledge base: Our first result tuple is $M_1^T :=$ (`Gerd, OntoWise`). Then, we have the query knowledge base (QKB_1) :

```
Gerd[worksIn ->>OntoWise].                       (12)
OntoWise[hasTopic ->>KnowledgeManagement].
```

Its relevant counterpart predicates in the given knowledge base (KB) are

```
Gerd[worksIn->>OntoWise].                        (13)
OntoWise[hasTopic->>KnowledgeManagement]
```

This is a perfect fit. $Simil(QKB_1, KB)$ computes to 1.

Our second result tuple is $M_2^T :=$ (`Andreas, TelekomProject`). Then, we have the query knowledge base (QKB_2) :

```
Andreas:Person[worksIn ->>TelekomProject]. (14)
TelekomProject[hasTopic->>KnowledgeManagement].
```

Its relevant counterpart predicates in the given knowledge base (KB) are

```
Andreas[worksIn ->>TelekomProject].          (15)
TelekomProject[hasTopic ->>KnowledgeDiscovery].
```

Hence, the similarity of the first predicate indicates a perfect fit and evaluates to 1, but the congruency of `TelekomProject[hasTopic->>KnowledgeManagement]` with `TelekomProject[hasTopic->>KnowledgeDiscovery]` measures less than 1. The instance match of `KnowledgeDiscovery` and `KnowledgeManagement` returns ½ in the given topic hierarchy. Therefore, the predicate match PM returns $\sqrt[3]{1\cdot 1\cdot \frac{1}{2}} \approx 0.79$. Thus, overall ranking of the second result is based on ½(1+0.79)=0.895. Therefore, the AIFB portal will display (`Gerd, OntoWise`) as the first result and (`Andreas, TelekomProject`) as the second one.

Remarks on semantic ranking. The reader may note some basic properties of the ranking: *(i)* similarity of knowledge bases is an asymmetric measure, *(ii)* the ontology defines a conceptual structure useful for defining similarity, *(iii)* the core concept for evaluating semantic similarity is cotopy defined by a dedicated hierarchy. The actual computation of similarity depends on which conceptual structures (*e.g.* hierarchies like taxonomy, part-whole hierarchies, or topic hierarchies) are selected for evaluating conceptual nearness. Thus, similarity of knowledge bases depends on the view selected for the similarity measure. Ranking of semantic queries using underlying ontological structures is an important means in order to allow users a more specific view onto the underlying knowledge base. The method that we propose is based on a few basic principles:

- Reinterprete the combination of query and results as query knowledge bases that may be compared with the explicitly given information.
- Give a measure for comparing two knowledge bases, thus allowing rankings of query results.

The reader should be aware that our measure may produce some rankings for results that are hardly comparable. For instance, results may differ slightly because of imbalances in a given hierarchy or due to rather random differences of depth of branches. In this case, ranking may perhaps produce results that are not better than unranked ones — but the results will not be any worse either

RDF OUTSIDE — FROM A SEMANTIC WEB SITE TO THE SEMANTIC WEB

In the preceding sections we have described the components and the underlying techniques of the SEAL framework and its instantination in the AIFB portal. Since we want to embed SEAL-based portals into the Semantic Web, we have developed means for RDF-capable software agents to process the portal data. Therefore, we have built an automatic RDF GENERATOR that dynamically generates RDF statements on each of the static and dynamic pages of the semantic knowledge portal.

Our current AIFB portal is „Semantic Web-ized" using RDF facts instantiated and defined according to the underlying AIFB ontology. This means, that *e.g.* for each person from the institute, contact information (telephone, fax, e-mail address) as well as professional information (research-area, research-group, projects involved in) are available for processing from world-wide software agents which understand this form of metadata representation. The RDFMAKER established in the Ontobroker framework (*cf.* [3]) was a starting point for building the RDF GENERATOR. The idea of RDFMAKER was, that from Ontobroker's internal knowledge warehouse RDF statements are generated.

RDF GENERATOR follows a similar approach and extends the principal ideas. In a first step it generates an

RDF(S)-based ontology that is stored on a specific XML namespace, *e.g.* in our concrete application:

```
xmlns:aifb= "http://ontobroker.semanticweb.org/
                          ontologies/aifb-2001-01-01.rdfs#"
```

Additionally, it queries the knowledge warehouse. Data, *e.g.* for a person, is checked for consistency, and, if possible, completed by applying the given F-Logic rules. We here give a short example of information, namely name, phone, fax, e-mail and supervisor, which may be generated and stored on a specific home-page of a researcher in the position of PhD student :

```
<rdf:RDF  xmlns:rdf="http://www.w3.org/1999/02/22-
rdf-syntax-ns#"
          xmlns:aifb="http://ontobroker.
semanticweb.org/ontologies/aifb-2001-01-01.rdfs#">
    <aifb:PhDStudent rdf:ID="per:ama">
       <aifb:name>Alexander Maedche</aifb:name>
       <aifb:phone>+49-(0)721-608 558</aifb:phone>
       <aifb:fax>+49-(0)721-608 6580</aifb:fax>
        <aifb:email>maedche@fzi.de</aifb:email>
        <aifb:supervisor rdf:resource="http://www.
aifb.uni-karlsruhe.de/studer.html#per:rst"/>
    </aifb:PhDStudent>
</rdf:RDF>
```

RDF GENERATOR is a configurable tool, in some cases one may want to use inferences to generate materialized, complete RDF descriptions on a home page, in other cases one may want to generate only ground facts of RDF. Therefore, RDF GENERATOR allows to switch axioms on and off in order to adopt the generation of results to varying needs. In order to collect RDF annotateted information from dedicated sources, software agents have to crawl that portion of the Web – by using RDF CRAWLER.

The RDF CRAWLER (*cf.* RDF CRAWLER free downloadable at http://ontobroker.semanticweb.org /rdfcrawler) is a tool which downloads interconnected fragments of RDF from the internet and builds a knowledge base from this data. Building an external knowledge base for the AIFB portal (its researcher, its projects, its publications, ...) becomes easy using the RDF CRAWLER and the machine-processable RDF data currently defined on AIFB portal. In general, RDF data may appear in Web documents in several ways. We distinguish between pure RDF (files that have an extension like „*.rdf"), RDF embedded in HTML and RDF embedded in XML. Our RDF CRAWLER uses RDF-API (*cf.* RDF-API free downloadable at http://www-db.stanford.edu /˜melnik/rdf/api.html) that can deal with the different embeddings of RDF as described above.

EXPERIENCES WITH ONTOLOGY ENGINEERING

The conceptual backbone of our SEAL approach is an ontology. For our AIFB portal application, we had to model concepts relevant in this setting. As SEAL has been maturing, we have developed a methodology for setting up ontology-based knowledge systems [18]. Our approach (*cf.* Figure 5) is mainly based on [16] and [5] but focuses on the application-driven development of ontologies. We here describe some experiences made during the ontology development for our AIFB portal.

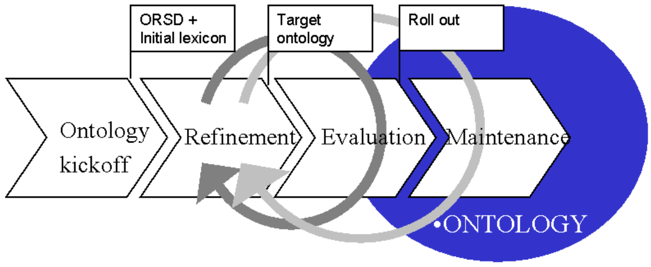

Figure 5: Ontology Development

Kickoff phase for ontology development

Setting up requirements for the AIFB ontology we had to deal mainly with modeling the research and teaching topics addressed by different groups of our institute and personal information about members of our institute. We took ourselves as an „input source" and collected a large set of lexical entries for research topics, teaching related topics and personal information, which represent the lexicon component of the ontology. By the sheer nature of these lexical entries, the ontology developers were not able to come up with all relevant lexical entries by themselves. It was necessary to go through several steps with domain experts *(viz.* our colleagues) in the refinement phase.

Refinement phase

We started to develop a baseline taxonomy that contained a heterarchy of research topics identified during the kickoff phase. An important result for us was to recognize that categorization was not based on an `isA`-taxonomy, but on a much weaker `HasSubtopic` relationship. *E.g.* „KDD" is a subtopic of „Knowledge Management", which means that it covers some aspects of „Knowledge Management" — but it does not reflect inheritance provided by an `isA`-taxonomy. It then took us three steps to model the currently active ontology. In the first step, lexical entries were collected by all members from the institute. Though we had already given the possibility to provide a rough categorization, the categories modeled by non-knowledge engineers were not oriented towards a model of the world, but rather towards the way people worked in their daily routine. Thus, their categorization reflected a particular rather than a shared view onto the domain. A lesson learned from this was that people need an idea about the nature of ontologies to make sound modeling suggestions. It was helpful to show existing prototypes of ontology-based systems to the domain experts.

In the second step, we worked towards a common understanding of the categorization and the derivation of implicit knowledge, such as „someone who works in logic also works in theoretical computer science" and inverseness of relations, *e.g.* „an author has a publication" is inverse to „a publication is written by an author".

In the third step, we mapped the gathered lexical entries to concepts and relations and organized them at a middle level. Naturally, this level involved the introduction of more generic concepts that people would usually not use when characterizing their work (such as „optimization“), but it also included „politically desired concepts“, because one own's ontology exhibits one's view onto the world. Thus, the ontology may become a political issue.

Modeling during early stages of the refinement phase was done with pen and paper, but soon we took advantage of our ontology environment OntoEdit (*cf.* free downloadable at http://www.ontoprise.de/) that supports graphical ontology engineering at an epistemological level as well as formalization of the ontology. Like in other ontology engineering projects, the formalization of the ontology is a non-trivial process where the ontology engineer has to draw the line between ontology and knowledge base. Therefore our final decisions were much disputed.

Evaluation phase

After all we found that participation by users in the constructionof the ontology was very good and met the previously defined requirements, as people were very interested to see their work adequately represented. Some people even took the time to learn about OntoEdit. However, the practical problem we had was that our environment does not yet support an ontology management module for cooperative ontology engineering. We embedded the ontology into our AIFB portal. It contains around 170 concepts (including 110 research topics) and 75 relations. This version is still running, but we expect maintenance to be a relevant topic soon. Therefore we are collecting feedback from our users - basically colleagues and students from our institute.

RELATED WORK

This section positions our work in the context of existing Web portals and also relates our work to other basic methods and tools that are or could be deployed for the construction of community Web portals, especially to related work in the area of semantic ranking of query results.

Related Work on Knowledge Portals. One of the well-established Web portals on the Web is Yahoo (cf. http://www.yahoo.com). In contrast to our approach Yahoo only utilizes a very light-weight ontology that solely consists of categories arranged in a hierarchical manner. Yahoo offers keyword search (local to a selected topic or global) in addition to hierarchical navigation, but is only able to retrieve complete documents, *i.e.* it is not able to answer queries concerning the contents of documents, not to mention to combine facts being found in different documents or to include facts that could be derived through ontological axioms. We get rid of these shortcomings since our portal is built upon a rich ontology enabling the portal to give integrated answers to queries.

The Ontobroker project [3] lays the technological foundations for the AIFB portal. On top of Ontobroker the portal has been built and organizational structures for developing and maintaining it have been established.

The approach closest to Ontobroker is SHOE [8]. In SHOE, HTML pages are annotated via ontologies to support information retrieval based on semantic information. Besides the use of ontologies and the annotation of Web pages the underlying philosophy of both systems differs significantly: SHOE uses description logic as its basic representation formalism, but it offers only very limited inferencing capabilities. Ontobroker relies on Frame-Logic and supports complex inferencing for query answering. Furthermore, the SHOE search tool does not provide means for a semantic ranking of query results. A more detailed comparison to other portal approaches may be found in [17].

Related Work on Semantic Similarity. Since our semantic ranking is based on the comparison of the query knowledge base with the given ontology and knowledge base, we relate our work to the comparison of ontological structures and knowledge bases (covering the same domain) and to measuring the similarity between concepts in a hierarchy. Although there has been a long discussion in the literature about evaluating knowledge-bases [12], we have not found any discussion about comparing two knowledge bases covering the same domain that corresponds to our ranking approach. Similarity measures for ontological structures have been investigated in areas like cognitive science, databases or knowledge engineering (*cf. e.g.*, [14, 13, 15, 9]). However, all these approaches are restricted to similarity measures between lexical entries, concepts, and template slots within one ontology.

Closest to our measure of similarity is work in the NLP community, named semantic similarity [14] which refers to similarity between two concepts in a `isA`-taxonomy such as the WordNet or CYC upper ontology. Our approach differs in two main aspects from this notion of similarity: Firstly, our similarity measure is applicable to a hierarchy which may, but not need be a taxonomy and secondly it is taking into account not only commonalties but also differences between the items being compared, expressing both in semantic-cotopy terms. This second property enables the measuring of self-similarity and subclass-relationship similarity, which are crucial for comparing results derived from the inferencing processes, executed in the background.

Conceptually, instead of measuring similarity between isolated terms (words), that does not take into account the relationship among word senses that matters, we measure similarity between „words in context“, by measuring similarity between Object-Attribute-Value pairs, where each term corresponds to a concept in the ontology. This enables us to exploit the ontological background knowledge (relations between concepts) in measuring the similarity, which expands our approach to a methodology for comparing knowledge bases.

From our point of view, our SEAL framework is rather unique with respect to the collection of methods used and the functionality provided. We have extended our community portal approach that provides flexible means for providing, integrating and accessing information [17], semantic ranking of generated answers and a smooth integration with the evolving Semantic Web. All these methods are integrated into one uniform environment, the SEAL framework.

CONCLUSION

In this paper we have shown our comprehensive approach SEAL for building semantic portals. In particular, we have focused on three issues.

First, we have described the general architecture of the SEAL framework, which is also used for our real-world case study, the AIFB portal. The architecture integrates a number of components that we have also used in other applications, like Ontobroker, the navigation and query module. Second, we have extended our semantic modules to include a larger diversity of intelligent means for accessing the Web site, *viz.* semantic ranking and machine access by crawling. Third, we have presented some experiences made during the ontology development for our case study - AIFB portal.

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 [11].

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 [4], [7].

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.

Acknowledgements The research presented in this paper would not have been possible without our colleagues and students at the Institute AIFB, University of Karlsruhe, and Ontoprise GmbH. Research for this paper was partially financed by Ontoprise GmbH, Karlsruhe, Germany, by US Air Force in the DARPA DAML project "OntoAgents", by EU in the IST-1999-10132 project "On-To-Knowledge" and by BMBF in the project "GETESS" (01IN901C0).

REFERENCES

[1]Angele, J., Schnurr, H.-P., Staab, S., and Studer, R. The times they are changing- the corporate history analyzer. In D. Mahling, U. Reimer, editors, *Proceedings of the Third International PAKM Conference*, pages 1.1-1.11, Basel, October, 2000.

[2] Benjamins, V.R., Fensel, D., Decker, S., and Perez, A.G. (KA): Building ontologies for the internet. *International Journal of Human-Computer Studies*, 51(1):687–712, 1999

[3] Decker, S., Erdmann, M., Fensel, D., and Studer, R. 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.

[4] Erdmann, M., Maedche, A., Schnurr H.-P., and Staab, S. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. *ETAI Journal - Section on Semantic Web*, 6(2001).

[5] Gomez-Perez, A. A framework to verify knowledge sharing technology. *Expert Systems with Application*, 11(4):519–529, 1996.

[6] Kifer, M., Lausen, G., Wu, J. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM* , Vol. 42, 741-883, 1995

[7] Handschuh, S., Staab, S., Maedche, A. CREAM - Creating relational metadata with a component-based, ontology-driven annotation framework. *ACM K-CAP 2001*. October, Vancouver, in press

[8] Heflin, J. and Hendler, J. Searching the Web with SHOE. In *Artificial Intelligence for Web Search*. AAAI Workshop.WS-00-01,pages 35–40. AAAI Press, 2000.

[9] Hovy, E. Combining and standardizing large-scale, practical ontologies for machine translation and other uses. In *Proc. Of the First Int. Conf. on Language Resources and Evaluation (LREC*), 1998.

[10] Maedche, A. and Staab, S. Discovering conceptual relations from text. In *Proceedings of ECAI-2000*, pages 321-324 IOS Press, Amsterdam, 2000.

[11] Maedche, A., Staab, S. Ontology Learning for the Semantic Web. *IEEE Intelligent Systems*, 16(2):72–79, 2001.

[12] Menzis, T.J. Knowledge maintenance: The state of the art. *The Knowledge Engineering Review*, 10(2), 1998.

[13] Rada, R., Mili, H., Bicknell, E. and Blettner, M. Development and application of a metric on semantic nets. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(1), pages 17-30, 1989.

[14] Resnik, P. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of IJCAI-95*, pages 448-453, Montreal, Canada, 1995.

[15] Richardson, R., Smeaton, A.F. and Murphy, J. Using Wordnet as knowledge base for measuring semantic similarity between words. Technical Report CA-1294, Dublin City University, School of Computer Applications, 1994.

[16] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R, Shadbolt, N., Van de Velde, W., and Wielinga, B. *Knowledge Engineering and Management — The CommonKADS Methodology*. The MIT Press, 1999.

[17] Staab, S., Angele, J., Decker, S., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R. and Sure, Y. AI for the Web — ontology-based community Web portals. In *AAAI 2000/IAAI 2000*, pages 1034-1039, AAAI Press/MIT Press, 2000.

[18] Staab, S., Schnurr, H.-P., Studer, R. and Sure, Y. Knowledge processes and ontologies. *IEEE Intelligent Systems*, 16(1):26–34, 2001.