SEmantic portAL — The SEAL approach

[1,3]Alexander Maedche, [1,2]Steffen Staab, [1]Nenad Stojanovic, [1,2,3]Rudi Studer,
and [1]York Sure

[1]Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS
mailto:{ama, sst, nst, rst, ysu}@aifb.uni-karlsruhe.de

[2]Ontoprise GmbH, Haid-und-Neu Straße 7, 76131 Karlsruhe, Germany
http://www.ontoprise.de

[3]FZI Research Center for Information Technologies,
Haid-und-Neu Straße 10-14, 76131 Karlsruhe, Germany
http://www.fzi.de/wim

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 generic 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 discuss the role that semantic structures make for establishing communication between different agents in general. We elaborate on a number of intelligent means that make semantic web sites accessible from the outside, *viz.* semantics-based browsing, semantic querying and querying with semantic similarity, semantic personalization, and machine access to semantic information at a semantic portal. As a case study we refer to the AIFB web site — a place that is increasingly driven by Semantic Web technologies.

1 Introduction

The widely-agreed core idea of the Semantic Web is the delivery of data on a semantic basis. Intuitively the delivery of semantically apprehended data should help with establishing a higher quality of communication between the information provider and the consumer. How this intuition may be put into practice is the topic of this paper.

We discuss means to further communication on a semantic basis. For this one needs a theory of communication that links results from semiotics, linguistics, and philosophy into actual information technology. We here consider *ontologies* as a sound semantic basis that is used to define the meaning of terms and hence to support intelligent access, *e.g.* by semantic querying [6] or dynamic hypertext views [29].

Thus, ontologies constitute the foundation of our SEAL (SEmantic portAL) approach. The origins of SEAL lie in Ontobroker [6], which was conceived for semantic search of knowledge on the Web and also used for sharing knowledge on the Web [3]. It then developed into an overarching framework for search and presentation offering access at a portal site [29]. This concept was then transferred to further applications [2, 31, 34] and is currently extended into a commercial solution (*cf.* `http://www.time2research.de`).

Here, we describe how we have applied SEAL to a real-world case study, *viz.* the AIFB web site. By the history of SEAL and related projects, we have distilled a methodology for construction of ontology-based knowledge systems that has been applied for the construction of the AIFB web site (Section 3). Following the description of this methodology and the experiences we made with its application to the AIFB site, we describe the SEAL core modules and its overall architecture (Section 4). 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 5). 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. Then, we tackle the issue of semantic personalization (Section 6). The principal idea is that underlying semantics may be very useful for presenting personalized views, because they allow for content-based views onto the web site. Finally, we present mechanisms to deliver and collect machine-understandable data (Section 7). They extend previous means for better digestion of web site data by software agents. Before we conclude, we give a short survey of related work.

2 Ontology and knowledge base

For our AIFB intranet, we explicitly model relevant aspects of the domain in order to allow for a more concise communication between agents, *viz.* within the group of sotware agents, between software and human agents, and — last not least — between different human agents. In particular, we describe a way of modeling an ontology that we consider appropriate for supporting communication between human and software agents.

2.1 Ontologies for communication

Research in ontology has its roots in philosophy dealing with the nature and organisation of being. In computer science, the term ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a particular model of the world, plus a set of explicit assumptions regarding the intended meaning of the words in the vocabulary. Both, vocabulary and assumptions, serve human and software agents to reach common conclusions when communicating.

Reference and meaning. The general context of communication (with or without ontology) is described by the meaning triangle [20]. The meaning triangle defines the interaction between symbols or words, concepts and things of the world (*cf.* Figure 1).

The meaning triangle illustrates the fact that although words cannot completely capture the essence of a reference (= concept) or of a referent (= thing), there is a correspondence between them. The re-

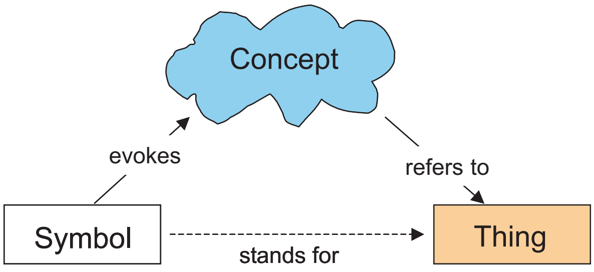


Figure 1: The Meaning Triangle

lationship between a word and a thing is indirect. The correct linkage can only be accomplished when an interpreter processes the word invoking a corresponding concept and establishing the proper linkage between his concept and the appropriate thing in the world.

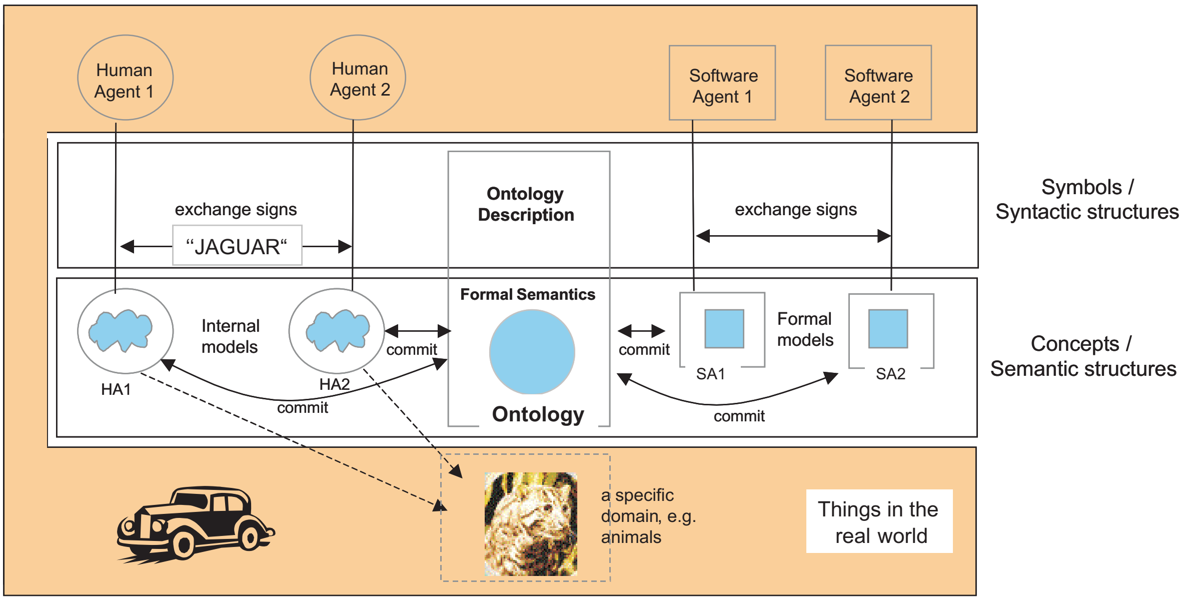


Figure 2: Communication between human and/or software agents

Logics. An ontology is a general logical theory constituted by a vocabulary and a set of statements about a domain of interest in some logic language. The logical theory specifies relations between signs and it apprehends relations with a semantics that restricts the set of possible interpretations of the signs. Thus, the ontology reduces the number of mappings from signs to things in the world that an interpreter who is committed to the ontology can perform — in the ideal case each sign from the vocabulary eventually stands for exactly one thing in the world.

Figure 2 depicts the overall setting for communication between human and software agents. We mainly distinguish three layers: First of all, we deal with things that exist in the real world, including in this example human and software agents, cars, and animals. Secondly, we deal with symbols and syntactic structures that are exchanged. Thirdly, we analyze models with their specific semantic structures.

Let us first consider the left side of Figure 2 without assuming a commitment to a given ontology.

Two human agents HA_1 and HA_2 exchange a specific sign, *e.g.* a word like “jaguar”. Given their own internal model each of them will associate the sign to his own concept referring to possibly two completely different existing things in the world, *e.g.* the animal *vs.* the car. The same holds for software agents: They may exchange statements based on a common syntax, however, they may have different formal models with differing interpretations.

We consider the scenario that both human agents commit to a specific ontology that deals *e.g.* with a specific domain, *e.g.* animals. The chance that they both refer to the same thing in the world increases considerably. The same holds for the software agents SA_1 and SA_2: They have actual knowledge and they use the ontology to have a common semantic basis. When agent SA_1 uses the term “jaguar”, the other agent SA_2 may use the ontology just mentioned as background knowledge and rule out incorrect references, *e.g.* ones that let “jaguar” stand for the car. Human and software agents use their concepts and their inference processes, respectively, in order to narrow down the choice of referents (*e.g.*, because animals do not have wheels, but cars have).

A new model for ontologies. Subsequently, we define our notion of ontology. However, in contrast to most other research about ontology languages it is not our purpose to invent a new logic language or to redescribe an old one. Rather what we specify is a way of *modeling* an ontology that inherently considers the special role of signs (mostly strings in current ontology-based systems) and references.

Our motivation is based on the conflict that ontologies are for human and software agents, but logical theories are mostly for mathematicians and inference engines. Formal semantics for ontologies is a *sine qua non*. In fact, we build our applications on a well-understood logical framework, *viz.* F-Logic [14]. However, in addition to the benefits of logical rigor, user and developer of an ontology-based system profit from ontology structures that allow to elucidate possible misunderstandings.

For instance, one might specify that the sign “jaguar” refers to the union of the set of all animals that are jaguars and the set of all cars that are jaguars. Alternatively, one may describe that “jaguar” is a sign that may either refer to a concept “animal-jaguar” or to a concept “car-jaguar”. We prefer the second way. In conjunction with appropriate GUI modules (*cf.* Sections 4ff) one may avoid presentations of ‘funny symbols’ to the user like “animal-jaguar”, while avoiding ‘funny inference’ such as may arise from artificial concepts like (‘animal-jaguar’ ∪ ‘car-jaguar’).

2.2 Ontology *vs.* knowledge base

Concerning the general setting just sketched, the term ontology is defined — more or less — as some piece of formal knowledge. However, there are several properties that warrant the distinction of knowledge contained in the ontology *vs.* knowledge contained in the so-called *knowledge base*, which are summarized in Table 1.

Table 1: Distinguishing ontology and knowledge base

	Ontology	Knowledge base
Set of logic statements	yes	yes
Theory	general theory	theory of particular circumstances
Statements are mostly	intensional	extensional
Construction	set up once	continuous change
Description logics	T-Box	A-Box

The ontology constitutes a general logical theory, while the knowledge base describes particular circumstances. In the ontology one tries to capture the general conceptual structures of a domain of interest, while in the knowledge base one aims at the specification of the given state of affairs. Thus, the ontology is (mostly) constituted by *intensional* logical definitions, while the knowledge base comprises (mostly) the *extensional* parts. The theory in the ontology is one which is mostly developed during the set up (and maintenance) of an ontology-based system, while the facts in the knowledge base may be constantly changing. In description logics, the ontology part is mostly described in the T-Box and the knowledge base in the A-Box. However, our current experience is that it is not always possible to distinguish the ontology from the knowledge base by the logical statements that are made. In the conclusion we will briefly mention some of the problems referring to some examples of following sections.

The distinctions ("general" *vs.* "specific", "intensional" *vs.* "extensional", "set up once" *vs.* "continuous change") indicate that for purposes of development, maintenance, and good design of the software system it is reasonable to distinguish between ontology and knowledge base. Also, they describe a rough shape of where to put which parts of a logical theory constraining the intended semantic models that facilitate the referencing task for human and software agents. However, the reader should note that none of these distinctions draw a clear cut borderline between ontology and knowledge base in general. Rather, it is typical that in a few percent of cases it depends on the domain, the view of the modeler, and the experience of the modeler, whether she decides to put particular entitities and relations into the ontology or into the knowledge base.

Both following definitions of ontology and knowledge base specify constraints on the way an ontology (or a knowledge base) should be modeled *in a particular logical language* like F-Logic or OIL:

Definition 1 (Ontology) *An ontology is a sign system $\mathcal{O} := (\mathcal{L}, \mathcal{F}, \mathcal{G}, \mathcal{C}, \mathcal{H}, \mathcal{R}, \mathcal{A})$, which consists of*

- *A **lexicon**: The lexicon contains a set of signs (lexical entries) for concepts, $\mathcal{L}^c$, and a set of signs for relations, $\mathcal{L}^r$. Their union is the lexicon $\mathcal{L} := \mathcal{L}^c \cup \mathcal{L}^r$.*

- *Two **reference functions** $\mathcal{F}$, $\mathcal{G}$, with $\mathcal{F} : 2^{\mathcal{L}^c} \mapsto 2^{\mathcal{C}}$ and $\mathcal{G} : 2^{\mathcal{L}^s} \mapsto 2^{\mathcal{S}}$. $\mathcal{F}$ and $\mathcal{G}$ link sets of lexical entries $\{L_i\} \subset \mathcal{L}$ to the set of concepts and relations they refer to, respectively, in the given ontology. In general, one lexical entry may refer to several concepts or relations and one concept or relation may be refered to by several lexical entries. Their inverses are $\mathcal{F}^{-1}$ and $\mathcal{G}^{-1}$.*

  In order to map easily back and forth and because there is a n to m mapping between lexicon and concepts/relations, $\mathcal{F}$ and $\mathcal{G}$ are defined on sets rather than on single objects.

- *A set $\mathcal{C}$ of **concepts**: About each $C \in \mathcal{C}$ exists at least one statement in the ontology, viz. its embedding in the taxonomy.*

- *A **taxonomy** $\mathcal{H}$: Concepts are taxonomically related by the irreflexive, acyclic, transitive relation $\mathcal{H}$, ($\mathcal{H} \subset \mathcal{C} \times \mathcal{C}$). $\mathcal{H}(C_1, C_2)$ means that C_1 is a subconcept of C_2.*

- *A set of binary **relations** $\mathcal{R}$: $\mathcal{R}$ denotes a set of binary relations.[1] They specify pairs of domain and ranges (D, R) with $D, R \in \mathcal{C}$.*

  The functions d and r applied to a binary relation Q yield the corresponding domain and range concepts D and R, respectively.

[1] Here at the conceptual level, we do not distinguish between relations and attributes.

- *A set of ontology axioms, $\mathcal{A}$.*

The reader may note that the structure we propose is very similar to the WordNet model described by Miller [19]. WordNet has been conceived as a mixed linguistic / psychological model about how people associate words with their meaning. Like WordNet, we allow that one word may have several meanings and one concept (synset) may be represented by several words. However, we allow for a seamless integration into logical languages like OIL or F-Logic by providing very simple means for definition of relations and for knowledge bases.

We define a knowledge base as a collection of object descriptions that refer to a given ontology.

Definition 2 (Knowledge Base) *We define a knowledge base as a 7-tupel* $\mathcal{KB} := (\mathcal{L}, \mathcal{J}, \mathcal{I}, \mathcal{W}, \mathcal{S}, \mathcal{A}, \mathcal{O})$*, that consists of*

- *a **lexicon** containing a set of signs for instances, $\mathcal{L}$.*
- *A **reference function** $\mathcal{J}$ with $\mathcal{J} : 2^{\mathcal{L}} \mapsto 2^{\mathcal{I}}$. $\mathcal{J}$ links sets of lexical entries $\{L_i\} \subset \mathcal{L}$ to the set of instances they correspond to.*

  Thereby, names may be multiply used, e.g. "Athens" may be used for "Athens, Georgia" or for "Athens, Greece".
- *a set of **instances** $\mathcal{I}$. About each $I_k \in \mathcal{I}, k = 1, \ldots, l$ exists at least one statement in the knowledge base, viz. a membership to a concept C from the ontology $\mathcal{O}$.*
- *A **membership function** $\mathcal{W}$ with $\mathcal{W} : 2^{\mathcal{I}} \mapsto 2^{\mathcal{C}}$. $\mathcal{W}$ assigns sets of instances to the sets of concepts they are members of.*
- ***Instantiated relations**, $\mathcal{S}$, are described, viz. $S \subseteq \{(x, y, z) | x \in \mathcal{I},\ y \in \mathcal{R},\ z \in \mathcal{I}\}$.*
- *A set of knowledge base axioms, $\mathcal{A}$.*
- *A reference to an ontology $\mathcal{O}$.*

Overall the decision to model some relevant part of the domain in the ontology *vs.* in the knowledge base is often based on gradual distinctions and driven by the needs of the application. Concerning the technical issue it is sometimes even useful to let the lexicon of knowledge base and ontology overlap, *e.g.* to use a concept name to refer to a particular instance in a particular context. In fact researchers in natural language have tackled the question how the reference function J can be dynamically extended given an ontology, a context, a knowledge base and a particular sentence.

3 Ontology engineering

The conceptual backbone of our SEAL approach is the ontology. For our intranet, we had to model the concepts relevant in this setting. As SEAL has been maturing, we have developed a methodology for setting up ontology-based knowledge systems which we sketch here. Its extended description can be found in [33]. We also describe some experiences made during the ontology development.

3.1 Methodology for ontology engineering

Until a few years ago the building of ontologies was done in a rather *ad hoc* fashion. Meanwhile there have been some few, but seminal proposals for guiding the ontology development process (*e.g.* [35, 10, 9]). For instance Guarino & Welty [10] give formal guidelines for constructing a consistent and reusable ontology. Another approach, the Methontology framework [9], includes the identification of the ontology development process, and stages through which an ontology passes during its lifetime.

In contrast to these methodologies, which mostly restrict their attention within the ontology itself, our approach (*cf.* Figure 3) focuses on the application-driven development of ontologies.

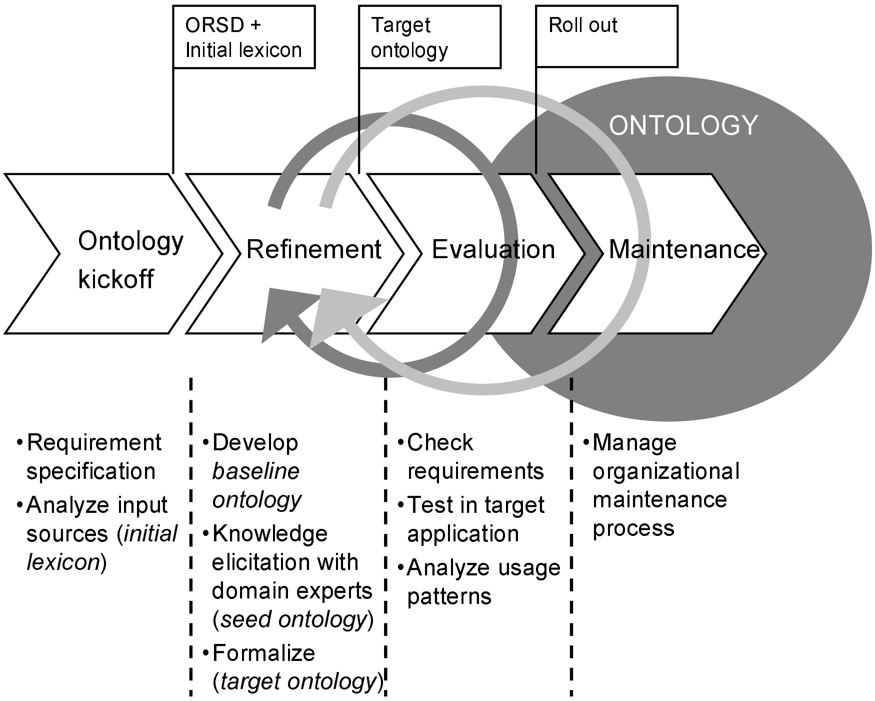

ORSD +
Initial lexicon
Target ontology
Roll out
ONTOLOGY
Ontology kickoff
Refinement
Evaluation
Maintenance
•Requirement specification
•Analyze input sources (*initial lexicon*)
•Develop *baseline ontology*
•Knowledge elicitation with domain experts (*seed ontology*)
•Formalize (*target ontology*)
•Check requirements
•Test in target application
•Analyze usage patterns
•Manage organizational maintenance process

Figure 3: Ontology Development

Kickoff phase for ontology development. The result of the kickoff phase is an ontology requirements specification document (ORSD) describing what an ontology should support, sketching the planned area of the ontology application and listing, *e.g.*, valuable input sources for the gathering of the baseline taxonomy in the refinement phase. Analysis of these input sources delivers an "initial lexicon" containing relevant lexical entries. In general, the ORSD should guide an ontology engineer to decide about inclusion and exclusion of lexical entries, their linkings to concepts / relations and the hierarchical structure of the ontology. In this early stage one should look for already developed and potentially reusable ontologies.

Refinement phase. The goal of the refinement phase is to produce a mature and application-oriented "target ontology" according to the specification given by the kickoff phase. It is divided into several subphases. First, the initial lexicon is linked to corresponding concepts / relations and the concepts are ordered in a taxonomy to form a "baseline ontology". Second, a knowledge elicitation process with domain experts based on the initial input from the baseline ontology is performed to develop a "seed ontology". There, the initial baseline ontology is modified and / or extended and axioms are added on top. Third, a formalization phase where the seed ontology is transferred into the target ontology which is expressed in formal representation languages like F-Logic [14], OIL [7] or Conceptual Graphs [27]. During the formalization phase the ontology engineer has to draw the line between ontology and knowledge base (*cf.* Section 2.2).

The usage of potentially reusable ontologies (identified during the kickoff phase) may improve the speed and quality of the development during the whole refinement phase. These ontologies might *e.g.* give useful hints for modeling decisions.

Evaluation phase. The evaluation phase serves as a proof for the usefulness of developed ontologies and their associated software environment. In a first step, the ontology engineer checks, whether the target ontology suffices the ontology requirements specification document. In a second step, the ontology is populated by adding instances to the knowledge base and tested in the target application environment — again, requirements from the ORSD serve as a base for evaluation. Feedback from beta users may be a valuable input for further refinement of the ontology. A valuable input may be the usage patterns of the ontology. The prototype system has to track the ways users navigate or search for concepts and relations. With such a "semantic logfile" (*cf.* Section 6.2) we may trace what areas of the ontology are often "used" and others which were not navigated. This phase is closely linked to the refinement phase and an ontology engineer may need to perform several cycles until the target ontology reaches the envisaged level — the "roll out" of the target ontology finishes the evaluation phase.

Maintenance phase. In the real world things are changing — and so do the specifications for ontologies. To reflect these changes ontologies have to be maintained frequently like other parts of software, too. We stress that the maintenance of ontologies is primarily an organizational process. There must be strict rules for the update-delete-insert processes within ontologies. We recommend that the ontology engineer gathers changes to the ontology and initiates the switch-over to a new version of the ontology after thoroughly testing possible effects to the application, *viz.* performing additional cyclic refinement and evaluation phases. Similar to the refinement phase, feedback from users and analysis of usage patterns may be a valuable input for identifying the changes needed. Maintenance should accompany ontologies as long as they are on duty.

3.2 Experience with ontology development

The methodology describes in general how to develop ontologies. We now describe experiences made while performing the described steps and include some aspects which were not covered by the methodology.

Kickoff phase for ontology development. Setting up requirements for the AIFB ontology we had to deal mainly with modeling the research topics done by different groups of our institute, teaching related topics and last but not least 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. By the sheer nature of these lexical entries, the ontology developers were not able to come up with all relevant lexical entries by themselves. Rather 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. We then switched to the second subphase of the refinement phase, *viz.* the development of the seed ontology through knowledge elicitation with our colleagues. There we needed three steps.

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 toward 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 very 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 would also include "politically desired concepts", because one own's ontology exhibits one's view onto the world. Thus, the ontology may become a political issue!

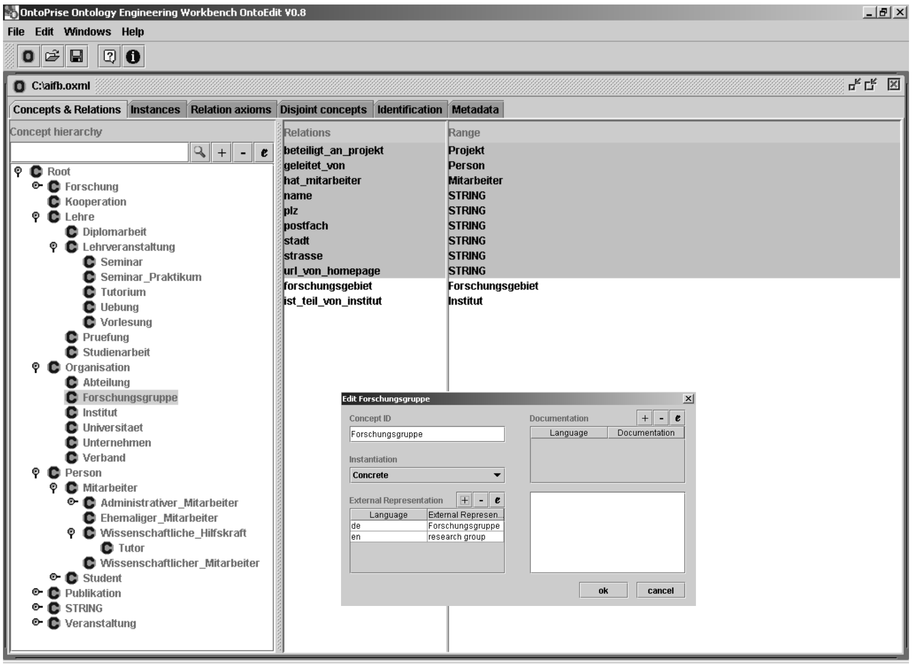
Modeling during early stages of the refinement phase was done with pen and paper, but soon we took advantage of our ontology environment OntoEdit (*cf.* Figure 4) that supports graphical ontology engineering at an epistemological level as well as formalization of the ontology. Our underlying inference engine (*cf.* Section 4) is based on F-Logic that we therefore chose as representation language for the target ontology. Like mentioned before, formalization is a non-trivial process where the ontology engineer has to draw the line between ontology and knowledge base. Our final decisions were much disputed. In the conclusion we will mention some of the intricacies that arise from excerpts of our ontology/knowledge base such as the ones given in Section 5.

Evaluation phase. After all we found that participation by users in the construction of 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 in it's version 1.0 into our application and enabled semantic logfiles (*cf.* Section 6.2) to track the usage of the ontology. On top of that we are collecting feedback from our users — basically colleagues and students from our institute. Currently we are still running the ontology version 1.0, but we expect maintenance to be a relevant topic soon.

4 SEAL infrastructure and core modules

The aim of our intranet application is the presentation of information to human and software agents taking advantage of semantic structures. In this section, we first elaborate on the general architecture for SEAL (SEmantic PortAL), before we explain functionalities of its core modules.



Relations	Range
beteiligt_an_projekt	Projekt
geleitet_von	Person
hat_mitarbeiter	Mitarbeiter
name	STRING
plz	STRING
postfach	STRING
stadt	STRING
strasse	STRING
url_von_homepage	STRING
forschungsgebiet	Forschungsgebiet
ist_teil_von_institut	Institut

Concept ID: Forschungsgruppe

Instantiation: Concrete

External Representation

Language	External Represen...
de	Forschungsgruppe
en	research group

Documentation

Language	Documentation

Figure 4: OntoEdit

4.1 Architecture

The overall architecture and environment of SEAL is depicted in Figure 5:

The *backbone* of the system consists of the *knowledge warehouse*, *i.e.* the data repository, and the *Ontobroker* system, *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.

At the front end one may distinguish between three types of *agents*: *software agents*, *community users* and *general users*. All three of them 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 at the portal over the internet. 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 modules 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* preprocessing module and/or rank retrieved results according to semantic similarity (done by the postprocessing module for *semantic ranking*). Queries also take advantage of the Ontobroker inferencing.

Third, only community users can provide data. Typical information they contribute includes personal data, information about research areas, publications, activities and other research information.

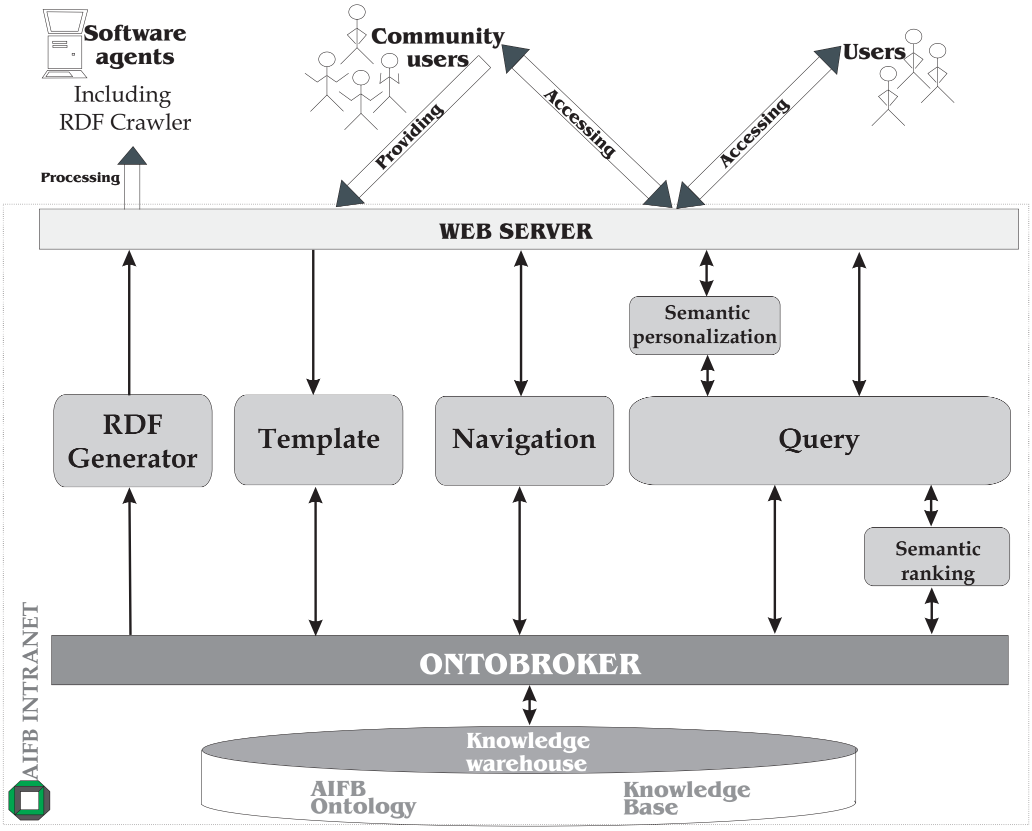


Figure 5: AIFB Intranet - System architecture

For each type of information they contribute there is (at least) one concept in the ontology. 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 modules stores the data in the knowledge warehouse.

4.2 Core modules

The core modules have been extensively described in [29]. 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 5), semantic personalization (Section 6), and semantic access by software agents (Section 7).

Ontobroker. The Ontobroker system [8] 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, in this architecture it is also used as an inference engine (server). 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 institute AIFB, he also belongs to AIFB. Thus, it is unnecessary to specify the membership to his research group *and* to AIFB. Conversely, the information seeker does not have to take care of inconsistent assignments, *e.g.* ones that specify membership to an AIFB research group, but that have erronously left out the membership to AIFB.

Knowledge warehouse. The knowledge warehouse [29] serves as repository for data represented in the form of F-Logic statements. It hosts the ontology, as well as the data proper. From the point of view of inferencing (Ontobroker) 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.

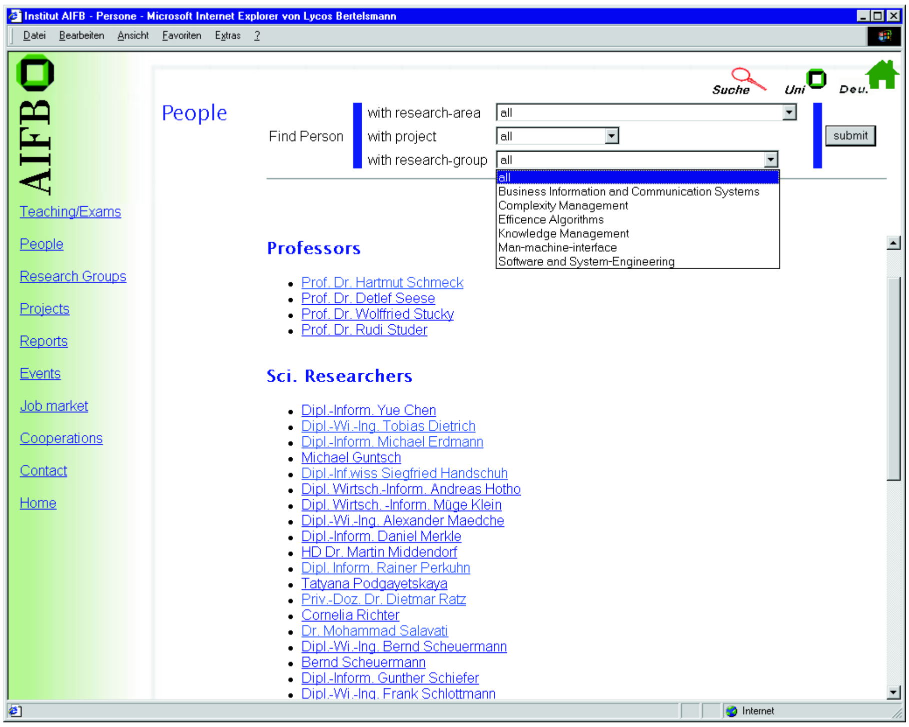
Navigation module. Beside the hierarchical, tree-based hyperlink structure which corresponds to hierarchical decomposition of domain, the navigation module enables complex graph-based semantic hyperlinking, based on ontological relations between concepts (nodes) in the domain. The conceptual approach to hyperlinking is based on the assumption that semantic 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 (*cf.* Figure 7) there are hyperlinks to web pages that describe the corresponding research groups, research areas and project web pages.

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, such as querying for projects or querying for people. For this purpose, selection lists that restrict query possibilities are offered to the user. The selection lists are compiled using knowledge from the ontology and/or the knowledge base. For instance, the query interface for persons 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 6).

Even simpler, one may apprehend 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 navigated.

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, in the AIFB intranet there is an input template (*cf.* Figure 7, upper left) generated from the concept definition of person (*cf.* Figure 7, lower left). The data is later on used by the navigation module to produce the corresponding person web page (*cf.* Figure 7, right hand side).

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 co-ordinator of a project enters information about which persons are participants of the project and this information is used when generating the person web page taking advantage of a corresponding F-Logic rule for inverse relationships. Hence, it is unnecessary to input this information in the person template.


Institut AIFB - Persone - Microsoft Internet Explorer von Lycos Bertelsmann
Datei Bearbeiten Ansicht Favoriten Extras ?
AIFB
Suche
Uni
Deu.
submit
Internet

Teaching/Exams
People
Research Groups
Projects
Reports
Events
Job market
Cooperations
Contact
Home

People

Find Person
with research-area: all
with project: all
with research-group: all

- all
- Business Information and Communication Systems
- Complexity Management
- Efficence Algorithms
- Knowledge Management
- Man-machine-interface
- Software and System-Engineering

Professors

- Prof. Dr. Hartmut Schmeck
- Prof. Dr. Detlef Seese
- Prof. Dr. Wolffried Stucky
- Prof. Dr. Rudi Studer

Sci. Researchers

- Dipl.-Inform. Yue Chen
- Dipl.-Wi.-Ing. Tobias Dietrich
- Dipl.-Inform. Michael Erdmann
- Michael Guntsch
- Dipl.-Inf.wiss Siegfried Handschuh
- Dipl. Wirtsch.-Inform. Andreas Hotho
- Dipl. Wirtsch. -Inform. Müge Klein
- Dipl.-Wi.-Ing. Alexander Maedche
- Dipl.-Inform. Daniel Merkle
- HD Dr. Martin Middendorf
- Dipl. Inform. Rainer Perkuhn
- Tatyana Podgayetskaya
- Priv.-Doz. Dr. Dietmar Ratz
- Cornelia Richter
- Dr. Mohammad Salavati
- Dipl.-Wi.-Ing. Bernd Scheuermann
- Bernd Scheuermann
- Dipl.-Inform. Gunther Schiefer
- Dipl.-Wi.-Ing. Frank Schlottmann

Figure 6: Searching form based on definition of concept Person

Ontology lexicon. The different modules described here make extensive use of the lexicon component of the ontology. The most prevalent use is the distinction between English and German (realized for presentation, though not for the template module, yet). 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[2] or SEAL[3].

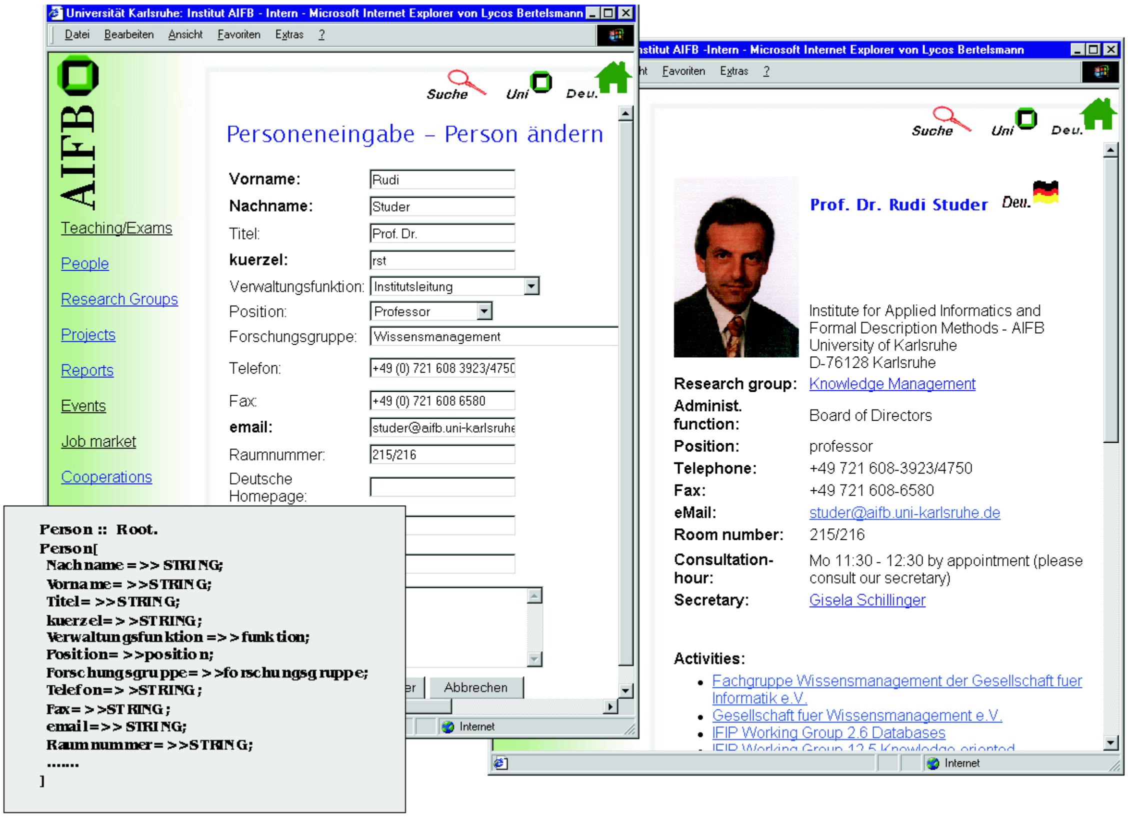
5 Semantic ranking

This section describes the architecture component "Semantic ranking" which has been developed in the context of our application. First, we will introduce and motivate the requirement for a ranking approach with a small example we are facing. 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 [16].

Let us assume the following ontology:

[2] Active server pages *vs.* active service providers.

[3] "SouthEast Asian Linguistics Conference" *vs.* "Conference on Simulated Evolution and Learning" *vs.* "Society for Evolutionary Analysis in Law" *vs.* "Society for Effective Affective Learning" *vs.* some other dozens — several of which are indeed relevant in our institute.



Personeneingabe – Person ändern

Vorname:	Rudi
Nachname:	Studer
Titel:	Prof. Dr.
kuerzel:	rst
Verwaltungsfunktion:	Institutsleitung
Position:	Professor
Forschungsgruppe:	Wissensmanagement
Telefon:	+49 (0) 721 608 3923/4750
Fax:	+49 (0) 721 608 6580
email:	studer@aifb.uni-karlsruhe
Raumnummer:	215/216
Deutsche Homepage:	

```
Person :: Root.
Person[
 Nachname =>> STRING;
 Vorname= >>STRING;
 Titel= >>STRING;
 kuerzel=> >STRING;
 Verwaltungsfunktion =>>funktion;
 Position= >>position;
 Forschungsgruppe= >>forschungsgruppe;
 Telefon=> >STRING;
 Fax= >>STRING;
 email=>> STRING;
 Raumnummer= >>STRING;
 .......
]
```

Prof. Dr. Rudi Studer

Institute for Applied Informatics and
Formal Description Methods - AIFB
University of Karlsruhe
D-76128 Karlsruhe

Research group:	Knowledge Management
Administ. function:	Board of Directors
Position:	professor
Telephone:	+49 721 608-3923/4750
Fax:	+49 721 608-6580
eMail:	studer@aifb.uni-karlsruhe.de
Room number:	215/216
Consultation-hour:	Mo 11:30 - 12:30 by appointment (please consult our secretary)
Secretary:	Gisela Schillinger

Activities:

- Fachgruppe Wissensmanagement der Gesellschaft fuer Informatik e.V.
- Gesellschaft fuer Wissensmanagement e.V.
- IFIP Working Group 2.6 Databases
- IFIP Working Group 12.5 Knowledge-oriented

Figure 7: Templates generated from concept definitions

$$
\begin{aligned}
1:&\quad \mathsf{Person} :: \mathsf{Object}[\textsc{worksIn} \Rightarrow\!\!\!\!\Rightarrow \mathsf{Project}].\\
2:&\quad \mathsf{Project} :: \mathsf{Object}[\textsc{hasTopic} \Rightarrow\!\!\!\!\Rightarrow \mathsf{Topic}].\\
3:&\quad \mathsf{Topic} :: \mathsf{Object}[\textsc{subtopicOf} \Rightarrow\!\!\!\!\Rightarrow \mathsf{Topic}].
\end{aligned}
\tag{1}
$$

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 for this Person is restricted to Project.

Let us further assume the following knowledge base:

$$
\begin{aligned}
4:&\quad \mathtt{KnowledgeManagement} : \mathsf{Topic}.\\
5:&\quad \mathtt{KnowledgeDiscovery} : \mathsf{Topic}[\textsc{subtopicOf} \twoheadrightarrow \mathtt{KnowledgeManagement}].\\
6:&\quad \mathtt{Gerd} : \mathsf{Person}[\textsc{worksIn} \twoheadrightarrow \mathtt{OntoWise}].\\
7:&\quad \mathtt{OntoWise} : \mathsf{Project}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeManagement}].\\
8:&\quad \mathtt{Andreas} : \mathsf{Person}[\textsc{worksIn} \twoheadrightarrow \mathtt{TelekomProject}].\\
9:&\quad \mathtt{TelekomProject} : \mathsf{Project}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeDiscovery}].\\
10:&\quad \textsc{forall}\ \ X,Y,Z\ \ Z[\textsc{hasTopic} \twoheadrightarrow Y] \leftarrow X[\textsc{subtopicOf} \twoheadrightarrow Y]\ \ \mathtt{and}\ \ Z[\textsc{hasTopic} \twoheadrightarrow X].
\end{aligned}
\tag{2}
$$

Definitions of instances in the knowledge base are syntactically very similar to the concept definition in F-Logic. In line 5 the instance KnowledgeDiscovery of the concept Topic is defined. Furthermore, the relation SUBTOPICOF is instantiated between KnowledgeDiscovery and

`KnowledgeManagement`. Similarly in line 6, it is stated that `Gerd` is a {concPerson working in `OntoWise`.

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

$$\texttt{FORALL}\ Y, Z \leftarrow Y[\textsc{worksIn} \twoheadrightarrow Z]\ \texttt{and}\ Z : \mathit{Project}[\textsc{hasTopic} \twoheadrightarrow \texttt{KnowledgeManagement}] \tag{3}$$

which may result in the tuples $M_1^T := (\texttt{Gerd}, \texttt{OntoWise})$ and $M_2^T := (\texttt{Andreas}, \texttt{TelekomProject})$. 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.

5.1 Reinterpreting queries

Our principal consideration builds on the definition of semantic similarity that we have first described in [16]. 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 re-interpreted 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[4]:

$$\texttt{FORALL}\ \overline{X} \leftarrow \overline{P}(\overline{X}, \overline{k}), \tag{4}$$

with $\overline{X}$ being a vector of variables $(X_1, \ldots, X_n)$, $\overline{k}$ 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 $m \times n$, 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 $\overline{X} := (Y, Z)$, $\overline{k} := (\texttt{``knowledge management''})$, $\overline{P} := (P_1, P_2)$, $P_1(a, b, c) := a[\textsc{worksIn} \twoheadrightarrow b], P_2(a, b, c) := b[\textsc{hasTopic} \twoheadrightarrow c]$ and

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

Now, we may define the query knowledge base i (QKB_i) by

$$QKB_i := \overline{P}(M_i, \overline{k}). \tag{6}$$

The similarity measure between the query knowledge base and the given knowledge base may then be computed in analogy to [16]. An adaptation and simplification of the measures described there is given in the following together with an example.

5.2 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 $\mathcal{H}$. For instance, in our example from above 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 $\mathcal{H}$, an part-whole hierarchy, or a categorization of topics. Here, we use the *upwards cotopy* (UC) defined as follows:

[4] Negation requires special treatment.

$$\text{UC}(O_i, H) := \{O_j | H(O_i, O_j) \vee O_j = O_i\} \tag{7}$$

UC is overloaded in order to allow for a set of objects M as input instead of only single objects, *viz.*

$$\text{UC}(M, H) := \bigcup_{O_i \in M} \{O_j | H(O_i, O_j) \vee O_j = O_i\} \tag{8}$$

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

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

Basically, OM reaches 1 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 superconcepts yields value 0).

Example. We here give a small example for computing UC and OM based on a given categorization of objects H. Figure 8 depicts the example scenario.

The upwards cotopy $\text{UC}(\texttt{knowledge discovery}, H)$ is given by $\text{UC}(\texttt{knowledge discovery}), H) = \{\texttt{knowledge discovery}, \texttt{knowledge management}\}$. The upwards cotopy $\text{UC}(\texttt{optimization}), H)$ is computed by $\text{UC}(\texttt{optimization}), H)) = \{\texttt{optimization}\}$.

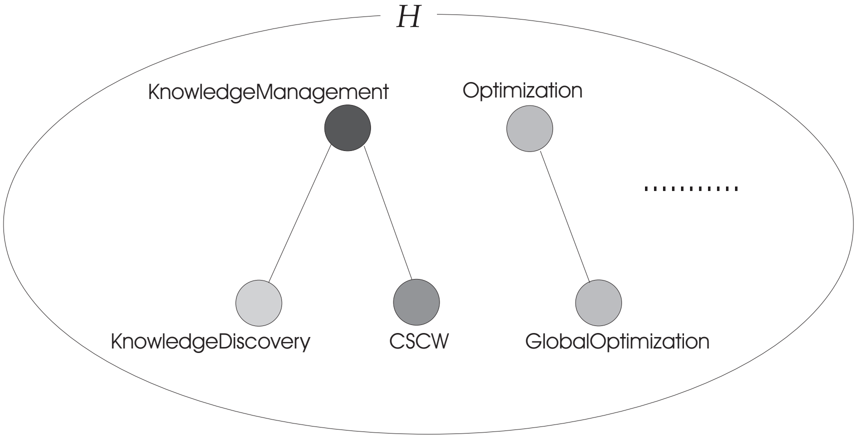


Figure 8: Example for computing UC and OM

Computing the object match OM between `KnowledgeManagement` and `Optimization` results in 0, the object match between `KnowledgeDiscovery` and `CSCW` computes to $\frac{1}{3}$.

The match introduced above may easily be generalized to relations using a relation hierarchy H_R. Thus, the predicate match (PM) for two n-ary predicate 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 — which need not be the case for the arithmetic mean:

$$\mathrm{PM}(P_1(I_1,\ldots,I_n),P_2(J_1,\ldots,J_n)) := \sqrt[n+1]{\mathrm{OM}(P_1,P_2,\mathcal{H}_R)\cdot\mathrm{OM}(I_1,J_1,H)\cdot\ldots\cdot\mathrm{OM}(I_n,J_n,H)}. \tag{10}$$

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:

$$Simil(QKB_i,KB) = Simil(\overline{P}(M_i,\overline{k}),KB) := \frac{1}{|\overline{P}|}\sum_{P_j\in\overline{P}}\max_{Q(M_i,\overline{k})\in KB.\mathcal{S}}\mathrm{PM}(P_j(M_i,\overline{k}),Q(M_i,\overline{k})). \tag{11}$$

For instance, comparing the two result tuples from our example above with the given knowledge base: First, $M_1^T := (\mathtt{Gerd},\mathtt{OntoWise})$. Then, we have the query knowledge base (QKB_1):

$$\begin{array}{l}\mathtt{Gerd}[\textsc{worksIn} \twoheadrightarrow \mathtt{OntoWise}].\\ \mathtt{OntoWise}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeManagement}].\end{array} \tag{12}$$

and its relevant counterpart predicates in the given knowledge base (KB) are:

$$\begin{array}{l}\mathtt{Gerd}[\textsc{worksIn} \twoheadrightarrow \mathtt{OntoWise}].\\ \mathtt{OntoWise}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeManagement}].\end{array} \tag{13}$$

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

Second, $M_2^T := (\mathtt{Andreas},\mathtt{TelekomProject})$. Then, we have the query knowledge base (QKB_2):

$$\begin{array}{l}\mathtt{Andreas}[\textsc{worksIn} \twoheadrightarrow \mathtt{TelekomProject}].\\ \mathtt{TelekomProject}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeManagement}].\end{array} \tag{14}$$

and its relevant counterpart predicates in the given knowledge base (KB) are:

$$\begin{array}{l}\mathtt{Andreas}[\textsc{worksIn} \twoheadrightarrow \mathtt{TelekomProject}].\\ \mathtt{TelekomProject}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeDiscovery}].\end{array} \tag{15}$$

Hence, the similarity of the first predicates indicates a perfect fit and evaluates to 1, but the congruency of $\mathtt{TelekomProject}[\textsc{hasTopic}\twoheadrightarrow\mathtt{KnowledgeManagement}]$ with $\mathtt{TelekomProject}[\textsc{hasTopic} \twoheadrightarrow \mathtt{KnowledgeDiscovery}]$ measures less than 1. The instance match of $\mathtt{KnowledgeDiscovery}$ and $\mathtt{KnowledgeManagement}$ returns $\frac{1}{2}$ in the given topic hierarchy. Therefore, the predicate match returns $\sqrt[3]{1\cdot 1\cdot\frac{1}{2}} \approx 0.79$. Thus, overall ranking of the second result is based on $\frac{1}{2}(1+0.79) = 0.895$.

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.

Thus, we may improve the interface to the underlying structures without changing the basic architecture. Of course, 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.

6 Semantic personalization

Personalization of web sites might be enabled on different levels like *e.g.* so-called ”Check-Box Personalization“ and ”Preference-based Personalization“ (*cf.* [23]). Both rely on the interaction between users and the underlying systems and aim at providing person-specific services and content to users. But they are differing the way systems are tracking a user’s interests to personalize web sites.

Check-box personalization offers configuration possibilities to users, allowing them to select from a given set of services and contents the ones they are looking for. Content is presented based on the check boxes marked by users. Preference-based personalization keeps track of users access patterns in logfiles. *E.g.* web usage mining tries to make sense of the web usage data created by surfers [28].

Considering check-box personalization, common systems rely on the selection of services and specific content areas and for preference-based personalization on monitoring of user-activities based on the sequence of clicked URLs. Our portal differs from common systems in offering semantic access. All users who access the web site use the underlying ontology for navigating, browsing and querying. To address check-box personalization users may store personalized *semantic bookmarks*. They rely on the semantic structure provided by the underlying ontology and like common bookmarks they facilitate access to regulary needed information.

While accessing the semantic web site, users leave footprints in form of clicked hyperlinks, but they also leave a trace of concepts and relations. Every query is composed from the ontology and (almost) every navigation follows underlying ontological structures. To enable preference-based personalization, we upgrade our logfiles with semantics to *semantic logfiles*. The user habits may then be semantically tracked, analyzed and used for personalization, general optimization of the web site and especially evaluation and maintenance of the ontology .

6.1 Semantic bookmarks

The AIFB mainly targets students and researchers. Both groups typically have different foci. Students tend to look for teaching-related topics and researchers tend to look for research-related topics. To satisfy their needs, we provide links to both areas from the starting page of our web site. But users sometimes need to have persistent pointers to specific information. Navigating and browsing through our web site as well as posting queries relies on the underlying ontology, therefore we have introduced *semantic bookmarks*.

Because many navigation facilities of our web site like *e.g.* hyperlinks and listboxes are created on the fly by our inference engine, semantic bookmarks basically contain predefined[5] query formulas. Upon selection bookmarks send their queries to the inference engine that reproduces the information shown to the user during his bookmarking process. Due to the fact that bookmarks contain query formulas they always produce most recent *i.e.* updated results. Semantic bookmarks are modeled in the ontology (*cf.* (16) Bookmark), their instantiations are stored in the knowledge warehouse (*cf.* (16) Bookmark1):

$$\begin{array}{ll}
\mathsf{Bookmark}[& \mathsf{Bookmark1{:}Bookmark}[\\
\text{\textsc{query}} \Rightarrow\!\!\!\!\Rightarrow \mathsf{STRING}; & \text{\textsc{query}} \twoheadrightarrow \text{“}\mathsf{FORALL\ X,N} \leftarrow \mathsf{X{:}Person[lastName} \twoheadrightarrow \mathsf{N].}\text{”}; \\
\text{\textsc{name}} \Rightarrow\!\!\!\!\Rightarrow \mathsf{STRING}; & \text{\textsc{name}} \twoheadrightarrow \text{“}\mathsf{List\ of\ all\ persons}\text{”}; \\
\text{\textsc{stylesheet}} \Rightarrow\!\!\!\!\Rightarrow \mathsf{PersonStylesheet}; & \text{\textsc{stylesheet}} \twoheadrightarrow \mathsf{PersonStylesheet1}; \\
\text{\textsc{start}} \Rightarrow\!\!\!\!\Rightarrow \mathsf{BOOLEAN}; & \text{\textsc{start}} \twoheadrightarrow \mathsf{FALSE}; \\
\text{\textsc{owner}} \Rightarrow\!\!\!\!\Rightarrow \mathsf{User}]. & \text{\textsc{owner}} \twoheadrightarrow \mathsf{User1}].
\end{array} \tag{16}$$

In addition to the QUERY formula, bookmarks contain several options for personalization. First, users may give a specific NAME for their bookmarks to describe the functionality. Second, they may choose a STYLESHEET from a given set of stylesheets for result presentation. Finally, semantic bookmarks might be marked as a START bookmark, so it will be executed as soon as users enter the starting page of our web site. This allows users to have quickest possible access to often needed informations. Every user may execute, edit and delete his own semantic bookmarks — identified by the relation OWNER.

In order to deliver personalized information the system needs to recognize users. We implemented a very simple ontology-based user administration allowing users to be recognized by their user-id. Therefore our ontology was expanded by user-specific concepts and relations (like User, ID, NAME etc.). Our system generates for each user an identifying id which is stored together with all relevant user information in our knowledge warehouse. For convenience, user-ids are also stored locally in cookies so that users are recognized automatically by the id stored on their computer. Possible pitfalls of such a simple approach are *e.g.* multi-user access from single computers or single-user access from multiple computers.

6.2 Semantic logfiles

Agents interact with our system through a web server (*cf.* Figure 5) who logs every request (*viz.* http-requests) into logfiles. Queries are processed to our inference engine through embedding query formulas into hyperlinks. Therefore all semantics are contained within logged hyperlinks.

Typically web servers track the following items: *Cookie*, *IP number*, *Timestamp*, *Request*. We take advantage of the information given. Cookies help to identify authenticated users and IP numbers in combination with timestamps help to distinguish non-authenticated users. Finally, the request string contains all concepts and relation of the ontology a user is interested in — due to the nature[6] of our ontology-enabled querying, navigation and browsing.

Currently we are working on transforming these logfiles into appropriate formats and to apply data mining methods and tools (like association rules). We are interested in answering, *e.g.*, the following questions:

[5] Experts might store any possible query into bookmarks by manually editing the queries.

[6] Our system is based on java servlets and uses URL-encoded queries.

- Does a single authenticated user have a special interest in a certain part of the ontology?
- Are there user groups, having similar interests?
- Which parts of the ontology are relevant to users, which ones are not?

Answering these questions will help us to optimize our system in iterative steps, *e.g.*, to enable ranked index lists providing each user with personalized shortcuts. Especially the last question is important for evaluation and maintenance of the ontology (*cf.* Section 3). To keep an ontology updated one might want to expand highly accessed parts, shrink rarely accessed parts and finally delete unaccessed parts. These topics are still ongoing research.

7 RDF outside — from a semantic web site to the Semantic Web

In the preceding sections we have described the development and the underlying techniques of the AIFB semantic web site. Having developed the core application we decided that RDF-capable software agents should be able to understand the content of application. 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 intranet application is "Semantic Web-ized" using RDF facts instantiated and defined according to the underlying AIFB ontology. On top of this generated and formally represented metadata, there is the RDF Crawler, a tool that gathers interconnected fragments of RDF from the internet.

7.1 RDF Generator — an example

The RDFMaker established in the Ontobroker framework (*cf.* [6]) was a starting point for building the RDF Generator. The idea of RDFMaker was, that from Ontobroker's internal data base, 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
`http://ontobroker.semanticweb.org/ontologies/aifb-onto-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 what type of data may be generated and stored on a specific homepage of a researcher:

```
<rdf:RDF
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:aifb = "http://ontobroker.semanticweb.org/aifb-2001-01-01.rdfs#">

       <aifb:PhDStudent rdf:ID="person:ama">
         <aifb:name>Alexander Maedche</aifb:name>
         <aifb:email>ama@aifb.uni-karlsruhe.de</aifb:email>
         <aifb:phone>+49-(0)721-608 6558</aifb:phone>
         <aifb:fax>+49-(0)721-608 6580</aifb:fax>
         <aifb:homepage>http://www.aifb.uni-karlsruhe.de/WBS/ama</aifb:homepage>
         <aifb:supervisor
           rdf:resource = "http://www.aifb.uni-karlsruhe.de/Staff/studer.html#person: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.

7.2 RDF CRAWLER

The RDF CRAWLER[7] 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 whole AIFB (its researcher, its projects, its publications, ...) becomes easy using the RDF CRAWLER and machine-processable RDF data currently defined on AIFB‘s web. We here shortly describe the underlying techniques of our RDF CRAWLER and the process of building a knowledge base. 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[8] that can deal with different embeddings of RDF described above.

One problem of crawling is the applied filtering mechanism: Baseline crawlers are typically restricted by a given depth value. Recently several new research work on so-called *focused crawling* has been published (*e.g. cf.* [4]). In their approach, they use a set of predefined documents associated with topics in a Yahoo like taxonomy to built a focused crawler. Two hypertext mining algorithms constitute the core of their approach. A classifier evaluates the relevance of a hypertext document with respect to the focus topics and a distiller identifies hypertext nodes that are good access points to many relevant pages within a few links. In contrast, our approach uses ontological background knowledge to judge the relevance of each page. If a page is highly relevant, the crawler may follow the links on the particular web site. If RDF data is available on a page, we judge relevance with respect to the quantity and quality of available data and by the existing URI's.

Example: Erdoes numbers. As mentioned above we here give a small example of a nice application that may be easily built using RDF metadata taken from AIFB using the RDF CRAWLER. The so-called *Erdoes numbers* have been a part of the folklore of mathematicians throughout the world for many years[9].

Scientific papers are frequently published with co-authors. Based on information about collaboration one may compute the Erdoes number (denoted $PE(R)$) for a researcher R. In the AIFB web site the RDF-based metadata allows for computing estimates of Paul Erdoes numbers of AIFB members. The numbers are defined recursively:

1. $PE(R) = 0$, iff R is Paul Erdoes
2. $PE(R)$ =min$\{PE(R_1) + 1\}$ else, where R_1 varies over the set of all researchers who have collaborated with R, *i.e.* have written a scientific paper together.

To put this into work, we need lists of publications annotated with RDF facts. The lists may be automatically generated by the RDF GENERATOR. Based on the RDF facts one may crawl relevant information into a central knowledge base and compute these numbers from the data.

[7] RDF CRAWLER is freely available for download at http://ontobroker.semanticweb.org/rdfcrawler.

[8] RDF-API is freely available at http://www-db.stanford.edu/˜melnik/rdf/api.html.

[9] The interested reader may have a look at http://www.oakland.edu/˜grossman/erdoshp.html for an overall project overview.

8 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 areas of personalization and semantic ranking of query results.

Related work on Knowledge Portals. One of the well-established web portals on the web is Yahoo[10]. 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. Personalization is limited to check-box personalization. We get rid of these shortcomings since our portal is built upon a rich ontology enabling the portal to give integrated answers to queries. Furthermore, our semantic personalization features provide more flexible means for adapting the portal to the specific needs of its users.

A portal that is specialized for a scientific community has been built by the Math-Net project [5]. At http://www.math-net.de/ the portal for the (German) mathematics community is installed that makes distributed information from several mathematical departments available. This information is accompanied by meta-data according to the Dublin Core [11] Standard [36]. The Dublin Core element "Subject" is used to classify resources as conferences, as research groups, as preprints etc. A finer classification (*e.g.* via attributes) is not possible except for instances of the publication category. Here the common MSC-Classification[12] is used that resembles a light-weight ontology of the field of mathematics. With respect to our approach Math-Net lacks a rich ontology that could enhance the quality of search results (esp. via inferencing), and the smooth connection to the Semantic Web world that is provided by our RDF generator.

The Ontobroker project [6] 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. Therefore, we compare our system against approaches that are similar to Ontobroker.

The approach closest to Ontobroker is SHOE [11]. 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 neither provides means for a semantic ranking of query results nor for a semantic personalization feature. A more detailed comparison to other portal approaches and underlying methods may be found in [29].

Related work on Personalization. Personalization is a feature of portals that attracts more and more interest, both from a research as well as from a commercial point of view. In contrast to our semantic personalization approach that exploits the conceptualization as offered by the ontology and the inferencing capabilities as provided by our ontobroker component, preference-based personal-

[10] http://www.yahoo.com

[11] http://www.purl.org/dc

[12] *cf.* Mathematical Subject Classification; http://www.ams.org/msc/

ization approaches typically rely on some notion of web mining exploiting the content of hypertext documents (web content mining), analyzing the hypertext links structure (web structure mining) or server/browser logs (web usage mining) [15]. *I.e.*, the organization and presentation of the web site might be optimized over time based on the analysis of such logfiles. While users are accessing the web sites, their habits are compared to previous users' behaviors to offer them personalized content like *e.g.* ranked index-lists (*cf.* [22]).

From a marketing point of view the analysis of customer behaviour might be used to relate the interaction of customers with the web site to aspects that are important for customer relationship management. *E.g.*, Easyminer [1] is a web intelligence tool that allows to analyze the customer behaviour with respect to aspects like the clicks-to-close value, *i.e.* how easily customers can find the information they are interested in and how this value can be improved through personalization. However, in contrast to our semantic personalization approach the data mining algorithms used in Easyminer are analyzing the syntactical link structures.

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 [18], we have not found any discussion about comparing two knowledge bases covering the same domain that corresponds to our semantic ranking approach. Similarity measures for ontological structures have been investigated in areas like cognitive science, databases or knowledge engineering (*cf. e.g.*, [25, 24, 26, 13]). 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 [25] 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 aspect 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, that are 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 (axioms and relations between concepts) in measuring the similarity, which expands our approach to a methodology for comparing knowledge bases.

From our point of view, our community portal system is rather unique with respect to the collection of methods used and the functionality provided. We have extended our community portal appraoch that provides flexible means for providing, integrating and accessing information [29] by semantic personalization features, semantic ranking of generated answers and a smooth integration with the evolving Semantic Web. All these methods are integrated into one uniform system environment, the SEAL framework.

9 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 considered the ontological foundation of SEAL and the methodological aspects of building ontologies for knowledge systems. There, we have made the experience that there are many big open issues that have hardly been dealt with so far. In particular, the step of formalizing the ontology raises very principal problems. The issue of where to put relevant concepts, *viz.* into the ontology *vs.* into the knowledge base, is an important one that deeply affects organization and application. However, there exist no corresponding methodological guidelines to base the decision upon so far. For instance, we have given the example ontology and knowledge base in (1) and (2). Using description logics terminology, we have equated the ontology with the "T-Box" and we have put the topic hierachy into the knowledge base ("A-Box"). An alternative could have been to formalize the topic hierarchy as an `isA`-hierarchy, which however it isn't and put it into the T-Box. We believe that both alternatives exhibit an internal fault, *viz.* the ontology should not be equated with the T-Box, but rather should its scope be independent from an actual formalization with particular logical statements. Its scope should to a large extent depend on soft issues, like "Who updates a concept?" and "How often does a concept change?" such as already indicated in Table 1.

Second, we have described the general architecture of the SEAL approach, which is also used for our real-world case study, the AIFB web site. The architecture integrates a number of components that we have also used in other applications, like Ontobroker, navigation or query module.

Third, we have extended our semantic modules to include a larger diversity of intelligent means for accessing the web site, *viz.* semantic ranking, personalization and machine access by crawling. Thus, we have shown a comprehensive approach to meet many of the challenges put forth in [21].

For the future, we see a number of new important topics appearing on the horizon. For instance, we consider approaches for ontology learning [17] 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 [32]. Given a particular conceptualization, we envision that one wants to be able to use a multitude of different inference engine taking advantage of different inferencing capabilities (temporal, non-monotonic, high scalability, etc.). Then, however, one needs means to change from one representation paradigm to the next [30].

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 [12].

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. We thank Jürgen Angele, Kalvis Apsitis (now: RITI Riga Information Technology Institute), Nils Braeunlich, Stefan Decker (now: Stanford University), Michael Erdmann, Dieter Fensel (now: VU Amsterdam), Siegfried Handschuh, Andreas Hotho, Mika Maier-Collin, Daniel Oberle, and Hans-Peter Schnurr. 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] S.S. Anand, M. Baumgarten, A. Buechner, and M. Mulvenna. Gaining insights into web customers using web intelligence. In *Proceedings of ECAI'2000*, pages 681–685, Berlin, Germany, August 2000.

[2] J. Angele, H.-P. Schnurr, S. Staab, and R. Studer. The times they are a-changin' — the corporate history analyzer. In D. Mahling and U. Reimer, editors, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland, October 30-31, 2000*, 2000. http://www.research.swisslife.ch/pakm2000/.

[3] V. Richard Benjamins and Dieter Fensel. Community is knowledge! $(KA)^2$. In *Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling, and Management (KAW '98), Banff, Canada, April 1998*, 1998.

[4] S. Chakrabarti, M. van den Berg, and B. Dom. Focused crawling: a new approach to topic-specific web resource discovery. In *Proceedings of WWW-8*, 1999.

[5] W. Dalitz, M. Grötschel, and J. Lügger. Information Services for Mathematics in the Internet (Math-Net). In A. Sydow, editor, *Proceedings of the 15th IMACS World Congress on Scientific Computation: Modelling and Applied Mathematics*, volume 4 of *Artificial Intelligence and Computer Science*, pages 773–778. Wissenschaft und Technik Verlag, 1997.

[6] 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.

[7] S. Decker, D. Fensel, F. van Harmelen, I. Horrocks, S. Melnik, M. Klein, and J. Broekstra. Knowledge representation on the web. In *Proceedings of the 2000 International Workshop on Description Logics (DL2000), Aachen, Germany*, 2000.

[8] D. Fensel, S. Decker, M. Erdmann, and R. Studer. Ontobroker: The Very High Idea. In *Proceedings of the 11th International Flairs Conference (FLAIRS-98), Sanibel Island, Florida, May*, 1998.

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

[10] N. Guarino and C. Welty. Identity, unity, and individuality: Towards a formal toolkit for ontological analysis. *Proceedings of ECAI-2000*, August 2000. available from http://www.ladseb.pd.cnr.it/infor/ontology/Papers/OntologyPapers.html.

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

[12] A. Hotho and G. Stumme, editors. *Semantic Web Mining — Workshop at ECML-2001 / PKDD-2001, Freiburg, Germany*, 2001.

[13] E. Hovy. 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.

[14] M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42:741–843, 1995.

[15] R. Kosala and H. Blockeel. Web mining reseach: A survey. *SIGKDD Explorations*, 2(1):1–15, 2000.

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

[17] A. Maedche and S. Staab. Ontology learning for the semantic web. *IEEE Intelligent Systems*, 16(2), 2001.

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

[19] G. Miller. Wordnet: A lexical database for English. *CACM*, 38(11):39–41, 1995.

[20] C.K. Odgen and I.A. Richards. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Routledge & Kegan Paul Ltd., London, 10 edition, 1923.

[21] M. Perkowitz and O. Etzioni. Adaptive web sites: an AI challenge. In *Proceedings of the 15th International Joint Conference on AI (IJCAI-97)*, pages 16–23, Nagoya, Japan, August 23-29 1997.

[22] M. Perkowitz and O. Etzioni. Adaptive web sites: Conceptual cluster mining. In *Proceedings of the 16th International Joint Conference on AI (IJCAI-99)*, pages 264–269, 1999.

[23] personalization.com. Frequently asked questions: Are there different types of personalization? http://www.personalization.com/basics/faq/faq2.asp observed at Feb 14, 2001, 2001.

[24] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1), 1989.

[25] P. Resnik. Knowledge maintenance: The state of the art. In *Proceedings of IJCAI-95*, pages 448–453, Montreal, Canada, 1995.

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

[27] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1992.

[28] J. Srivastava, R. Cooley, M. Deshpande, and P. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[29] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic community web portals. *Proc. of WWW9 / Computer Networks*, 33(1-6):473–491, 2000.

[30] S. Staab, M. Erdmann, and A. Maedche. Semantic patterns. Technical report, Institute AIFB, University of Karlsruhe, 2001.

[31] S. Staab and A. Maedche. Knowledge portals — ontologies at work. *AI Magazine*, 21(2), Summer 2001.

[32] S. Staab, A. Maedche, and S. Handschuh. An annotation framework for the semantic web. In *Proceedings of the First Workshop on Multimedia Annotation, Tokyo, Japan, January 30-31, 2001*, 2001.

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

[34] Y. Sure, A. Maedche, and S. Staab. Leveraging corporate skill knowledge - From ProPer to OntoProper. In D. Mahling and U. Reimer, editors, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management. Basel, Switzerland, October 30-31, 2000*, 2000. http://www.research.swisslife.ch/pakm2000/.

[35] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *Knowledge Sharing and Review*, 11(2), June 1996.

[36] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. *Dublin Core Metadata for Resource Discovery*. Number 2413 in IETF. The Internet Society, September 1998.