Journal of Universal Computer Science, vol. 7, no. 7 (2001), 566-590
submitted: 1/5/01, accepted: 6/6/01, appeared: 28/7/01 © Springer Pub. Co.

SEAL-II — The Soft Spot between Richly Structured and Unstructured Knowledge

Andreas Hotho
(Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
hotho@aifb.uni-karlsruhe.de)

Alexander Maedche
(FZI Research Center for Information Technologies
at the University of Karlsruhe, 76131 Karlsruhe, Germany
maedche@fzi.de)

Steffen Staab
(Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
and
Ontoprise GmbH, Haid-und-Neu Straße 7, 76131 Karlsruhe, Germany
staab@aifb.uni-karlsruhe.de)

Rudi Studer
(Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany
and
FZI Research Center for Information Technologies
at the University of Karlsruhe, 76131 Karlsruhe, Germany
and
Ontoprise GmbH, Haid-und-Neu Straße 7, 76131 Karlsruhe, Germany
studer@aifb.uni-karlsruhe.de)

Abstract: Recently, the idea of semantic portals on the Web or on the intranet has gained popularity. Their key concern is to allow a community of users to present and share knowledge in a particular (set of) domain(s) via semantic methods. Thus, semantic portals aim at creating high-quality access — in contrast to methods like information retrieval or document clustering that do not exploit any semantic background knowledge at all. However, by way of this construction semantic portals may easily suffer from a typical knowledge management problem. Their initial value is low, because only little richly structured knowledge is available. Hence the motivation of its potential users to extend the knowledge pool is small, too.

We here present SEAL-II, a methodology for semantic portals that extends its previous version, by providing a range of ontology-based means for hitting the soft spot between unstructured knowledge, which virtually comes for free, but which is of little use, and richly structured knowledge, which is expensive to gain, but of tremendous possible value. Thus, we give the portal builder tools and techniques in an overall framework to start the knowledge process at a semantic portal. SEAL-II takes advantage of the ontology in order to initiate the portal with knowledge, which is more usable than unstructured knowledge, but cheaper than richly structured knowledge.

Key Words: Knowledge Portal, Ontology
Category: H.0

1 Introduction

With SEAL (Staab et al., 2000; Staab & Maedche, 2001; Maedche et al., 2001) we have presented a comprehensive architecture for a semantic portal offering a broad range of tools for improving the benefit–effort ratio of semantic portals. This technology, viz. the easy and adequate presentation and exchange of information based on ontologies in conjunction with little additional editing effort, offers itself to knowledge management tasks like corporate history analysis (Angele et al., 2000) or skill management (Sure et al., 2000).

The life cycle of such a semantic portal spans three intertwined subcycles (cf. (Staab et al., 2001)): First, in a so-called *knowledge meta process* the domain of the application is modelled in an ontology. Second, in an initial *knowledge instantiation phase* one tries to scrape whatever knowledge is available from legacy systems, such as database contents. Thus, the portal may be started with sufficient knowledge to motivate employees to use the system. Third, in the *knowledge process* proper, i.e. the process undertaken by all users of the system, knowledge on the portal is used and new knowledge is contributed to the portal.

With SEAL-II we aim at a substantial extension of SEAL working on two important issues:

1. **Motivation:** People tend to use systems on a tit-for-tat basis. They tend not to invest work when they cannot recognize immediate benefit from it. Thus, semantic portals, or KM systems in general, that really start from scratch are easily ignored, as no one leaves the trap in this prisoners' dilemma.

2. **Grey-shades along the benefit–effort scale:** Knowledge items in SEAL are forced into one of the two categories *valuable* or *useless*. There is little balance between not having knowledge items at all (you could still do information retrieval) and investing efforts to have them in the ontology-based KM system. In addition, however, one would like to offer means that allow to trade-off more finely on the scale of benefit divided by effort.

SEAL-II tackles the soft spot between unstructured knowledge and richly structured knowledge. First, it provides new possibilities to create knowledge when instantiating the semantic portal while taking full advantage of the ontology created in the knowledge meta process. Thus, one may motivate people to actually use the system and contribute to it. Second, it accounts for the gray shades of knowledge regions that live too fast (e.g. are updated too often) to be solidified into ontology-based facts. The latter type of facts should not only be accessible by information retrieval techniques, but also by exploratory means.

Thus, on a scale between richly and unstructured knowledge we place a set of (partially new) techniques between the two extremes (cf. Figure 1). At the unstructured end, one finds techniques such as information retrieval, document clustering and keyword matches. At the other extreme, we have developed a set of techniques for providing richly structured knowledge (Staab et al., 2000; Staab & Maedche, 2001). This paper is about techniques for filling the void between the two extremes and about an architecture that allows for flexible scaling on the degree of structure of information. The overall approach presented here has been instantiated as **FZI-Broker**, a knowledge broker for the FZI knowledge management research group.

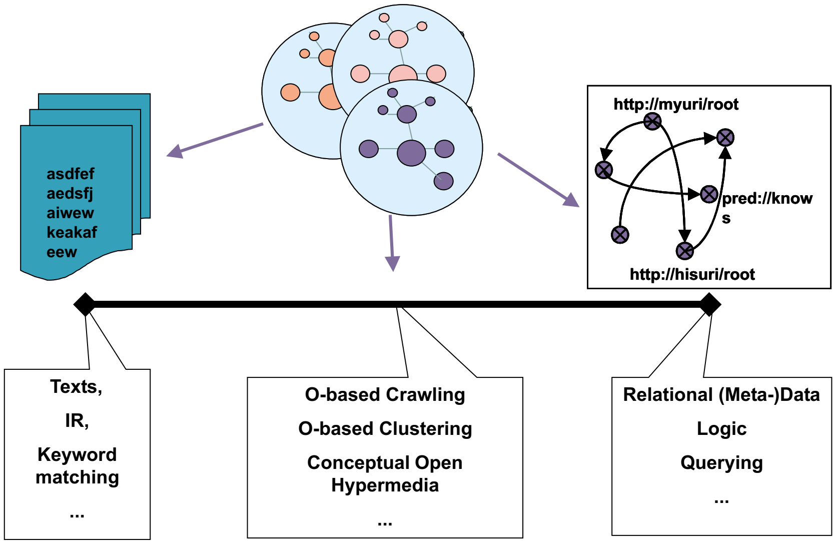


Figure 1: Coping with different needs: KM techniques for structured and unstructured information

In the following, we will first describe the architecture of SEAL-II — modifying slightly the original proposal. Thereafter, we sketch the initialization process of SEAL-II, i.e. the setup of the general approach into a particular application (Section 3). Section 4 outlines the underlying representations in the knowledge warehouse. We continue with a description of the ontology-based crawler, a tool that allows to grow the knowledge base from a number of seed HTML pages by crawling HTML texts and metadata. Section 6 elaborates ontology-based clustering, a technique that we have developed in order to provide subjective, concept-based views onto crawled documents. Conceptual open hypermedia extends conventional linking by ontology-construed relations (Section 7). Before we conclude we shortly describe our instantiations of SEAL-II, the FZI-Broker and the HR-TopicBroker, and give a short survey of related work.

2 Architecture

We here show how the SEAL architecture has been extended to tackle the "soft spot between unstructured knowledge and richly structured knowledge". In this section, we elaborate on the general architecture of SEAL-II that extends the SEAL architecture presented in (Maedche et al., 2001) and explain the functionalities of its core components in detail in the subsequent sections. Figure 2 depicts the overall architecture that underlies SEAL-II. The following main components are contained in the SEAL-II architecture:

- **Knowledge Warehouse:** The components of SEAL-II are built around the ontology that is stored in the knowledge warehouse. In general, the function of the knowledge warehouse is to store a variety of different types of structured and unstructured knowledge. The major parts of the warehouse are ontologies, fact knowledge, and document representations. The knowledge warehouse does not distinguish between schema and non-schema data as commonly known from typical relational databases. A detailed description of the different kinds of data that are stored in the knowledge warehouse is given in section 4.
- **Inference Engine:** The Ontobroker system (Decker et al., 1999) is a deductive, object-oriented database system. It is based on F-Logic allowing to describe ontologies, rules and facts. Beside other usage, it is used as an inference engine to derive new knowledge based on the existing fact knowledge contained in the knowledge warehouse.
- **Extractor:** The extractor analyzes natural language texts in order to recognize concepts. In our current implementation we use a very simple processing mechanism for recognizing concepts. The porter stemming algorithm (Porter, 1980) is used to compute word stems from a given document. The lexicon (cf. section 4) maps word stems to concepts. Thus, a look-up in the lexicon retrieves the concepts that are referred by a specific word stem. The reader may note that we are aware that a more complex processing strategy based on shallow linguistic processing techniques[1] may improve the overall quality of our system.
- **Ontology-focused Crawler:** An important component in the architecture is the ontology-focused document and (meta-)data crawler. The crawler takes the ontology from the knowledge warehouse and uses it for judging relevancy of documents in order to control the search in the web space. Thus, each document is represented by so-called concept vectors. Additionally, the crawler is able to collect relational

[1] The GETESS (German Text Exploitation and Search System) project pursues the tight integration between linguistic processing and ontological background knowledge (cf. (Staab et al., 1999)).

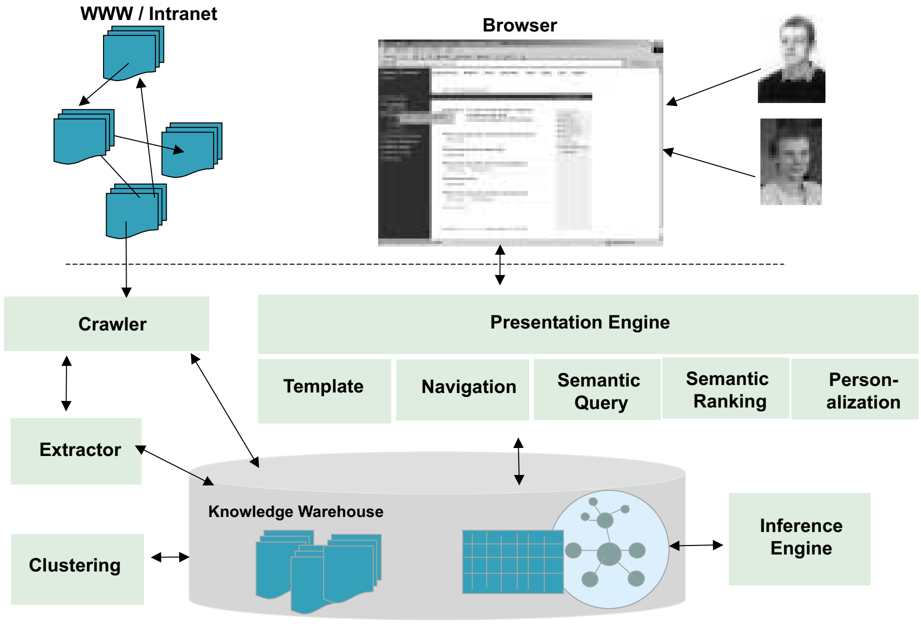


Figure 2: Architecture

metadata that have been generated by community members, e.g. by using a semantic annotation tool (cf. (Staab et al., 2000)). The ontology-focused crawler is further discussed in section 5.

- **Ontology-based Clustering:** A specific feature of the document part of SEAL-II is the ontology-based clustering component described in section 6. Ontology-based clustering computes structure for knowledge available in unstructured documents. It clusters similar documents into groups while taking advantage of background knowledge from the ontology.

- **Presentation:** At the front end we use a simple web application that is based on the idea of conceptual hypermedia. The interface is automatically generated by exploiting facts from the knowledge warehouse (cf. section 7). Thereby, the presentation component accesses the following modules available in SEAL-II:

  - **Template:** As mentioned earlier SEAL-II allows the contribution of information by users. The template module generates an HTML form for each concept that a user may instantiate and relate with other concepts. In general, users may add information in the following different ways: *First*, they may add documents (refering to them by URLs) to the knowledge warehouse. These document serve as input for the relevancy search for further documents using the crawler. *Second*, they may define formal instances of concepts contained in the ontology, e.g. they may define a concrete PROJECT like `OntoKnowledge` as fact knowledge described in a given document. *Third*, they may define attributes of and relations between instances, e.g. it may be interesting to define the attribute URL of the instance `OntoKnowledge`, namely `http://www.ontoknowledge.org` or to define the relation PARTICIPATES between the instances `AIFB` and `OntoKnowledge`.

- **Semantic Query & Navigation:** 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 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.

- In addition to the presentation modules described above, SEAL-II accesses the two components **personalization** and **semantic ranking** that are described in detail in (Maedche et al., 2001).

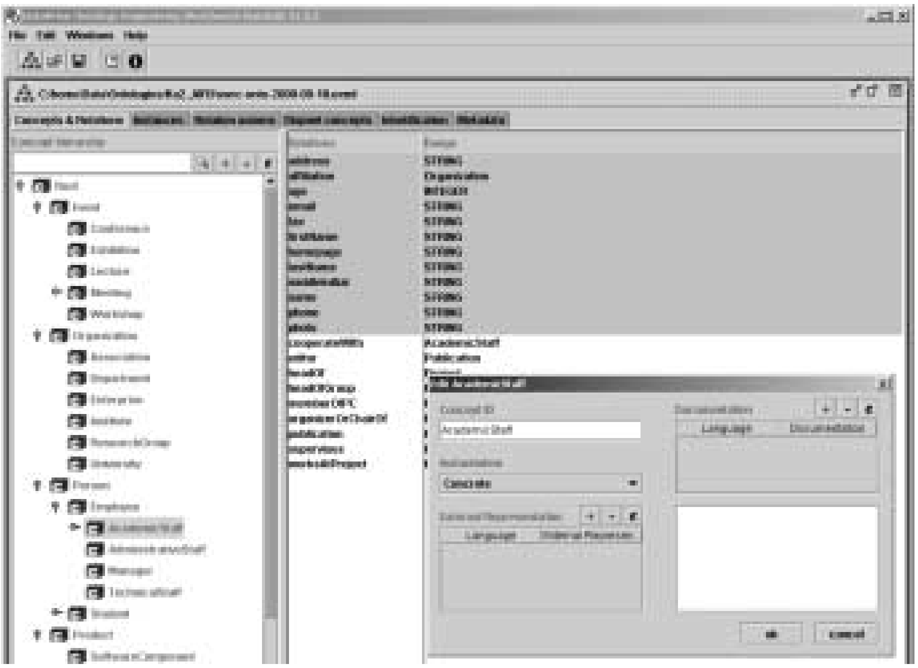
3 Make Application

The previous section shows how the different ontology-based modules contained in SEAL-II interact. In this section, we want to sketch how one may configure and instantiate the SEAL-II architecture. The basic idea of SEAL-II is to extend basic services such as given, e.g., with ZOPE[2]. ZOPE is an web application server that provides basic mechanisms for web site management including, e.g., user administration, undo functions, database integration, user administration, HTML programming environment, plug-in API, etc. This is an extensible list, which may be extended by programmers that package their modules into so-called “products”. People that instantiate the application server simply select the appropriate products and add dynamic HTML pages, in particular HTML layout. The ultimate goal of SEAL-II is to provide such configuration facilities not only on the “product” level, i.e. the level of standard website processes, but also on the content level. For the latter, the configuration and selection of ontologies play the crucial role that determines the structuring of content *within* the different “products”.

[2] cf. http://www.zope.org

3.1 Engineer Ontology

The conceptual backbone of our SEAL-II approach is the ontology. For instantiating SEAL-II, one has to model the concepts and relations relevant in a specific domain. As SEAL-II has been maturing, we have developed a methodology for setting up ontology-based knowledge systems (cf. (Staab et al., 2001)). This methodology is supported to a large extent by our ontology engineering environment ONTOEDIT (cf. Figure 3) and its submodule ONTOKICK.



Relations	Range
address	STRING
affiliation	Organization
age	INTEGER
email	STRING
fax	STRING
firstName	STRING
homepage	STRING
lastName	STRING
middleInitial	STRING
name	STRING
phone	STRING
photo	STRING
cooperatesWith	AcademicStaff
editor	Publication
headOf	Project
headOfGroup	
memberOfPC	
organizerOrChairOf	
publication	
supervises	
worksAtProject	

Concept ID: AcademicStaff

Instantiation: Concrete

Language	Documentation

Language	External Resources

Figure 3: OntoEdit Screenshot with SWRC ontology

ONTOEDIT provides the basic means for building concept and relation hierarchies as well as describing inference rules. For instance, in our snapshot one may recognize that in the Semantic Web Research Community Ontology[3], which serves as the basis for our FZI-Broker, ACADEMICSTAFF is a subclass of EMPLOYEE. ACADEMICSTAFF has, e.g., the relation COOPERATESWITH. Some of its relations — the ones in the grey shade — are inherited from EMPLOYEE. The submodule ONTOKICK allows to align the ontology specification documents with their corresponding concepts and relations. It allows to investigate domain texts in order to facilitate the discovery of concepts and

[3] cf. http://ontobroker.semanticweb.org/ontos/swrc.html

relations from the texts by the ontology engineer.

3.2 Make Ontology-based Applications

Essentially, we aim at promoting the shift from programming towards modeling and reuse of existing software. In particular on semantic portals many of the core processes — knowledge contribution, knowledge sharing, etc. — are nearly standard. Similar to knowledge acquisition methodologies and tools like Protege (Eriksson et al., 1994; Grosso et al., 1999), which have been used to (semi-)automatically generate fact acquisition interfaces from ontologies, ontologies may be used to (semi-)automatically present to and acquire facts from the users of a semantic portal. In addition, one must only add:

- Metadata about the ontology: In order to select important concepts for different views one may describe meta information about ontology concepts and relations, e.g. the relative importance of a concept for presentation purposes. Alternatively, one may have several ontologies for different products or different views onto the same ontology for varying products.
- Layout: The more that knowledge processes and configuration steps become standard, the larger is the share of overall efforts dedicated to layout concerns.

Some part of this overall vision has already been realized. Though, we still have to go a considerable distance for a ready-to-go solution that incorporates the full potential of "ontology application servers", we may now rather quickly instantiate SEAL-II — just by engineering a new ontology and by instantiating basic parameters like seed pages for crawling.

4 Knowledge Warehouse

The function of the knowledge warehouse is to store a variety of different types of structured and unstructured knowledge. The major parts of the warehouse are:

- *Ontologies*: The transparent storing of ontologies and corresponding data allows to combine both in intelligent ways such as outlined below.
- *Fact knowledge*: Fact knowledge is available in our warehouse like in an object-oriented database with the ontology as the corresponding schema. In addition, however, we allow facts to have dual nature, as fact knowledge may also be sometimes used as ontology knowledge. This is helpful, e.g. for describing concepts on the ontological level, but also to assign to them meta-statements that describe, e.g., their relative importance for presentation to the user.
- A *Document representation* is factual knowledge arranged in a manner suitable for retrieval or other kinds of processing. We employ three major types of document representations, as described below.

4.1 Ontology

Following Tom Gruber, we understand ontologies as a formal specification of a shared conceptualization of a domain of interest to a group of users (cf. (Gruber, 1993)). Similar to the term "information system in a wider sense" this notion of ontologies contains many aspects, which either cannot be formalized at all or which cannot be formalized for practical reasons. For instance, it might be too cumbersome to model the interests of all the persons involved. Concerning the formal part of ontologies (or "ontology in the narrow sense"), we employ a two-part structure. The first part (cf. Definition 1) describes the structural properties that virtually all ontology languages exhibit. Note that we do not define any syntax here, but simply refer to these structures as a least common denominator for many logical languages, such as OIL (Fensel et al., 2001) or F-Logic (Kifer et al., 1995):

Definition 1. Let $\mathcal{L}$ be a logical language having a formal semantics in which inference rules can be expressed. An *abstract ontology* is a structure $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ consisting of

- two disjoint sets C and R whose elements are called *concepts* and *relations*, resp.,
- a partial order $\leq_C$ on C, called *concept hierarchy* or *taxonomy*,
- a function $\sigma \colon R \to C \times C$ called *signature*,
- a partial order $\leq_R$ on R where $r_1 \leq_R r_2$ implies $\sigma(r_1) \leq_{C \times C} \sigma(r_2)$, for $r_1, r_2 \in R$, called *relation hierarchy*.
- and a set IR of inference rules expressed in the logical language $\mathcal{L}$.

The function $\mathrm{dom} \colon R \to C$ with $\mathrm{dom}(r) := \pi_1(\sigma(r))$ gives the *domain* of r, and the function $\mathrm{range} \colon R \to C$ with $\mathrm{range}(r) := \pi_2(\sigma(r))$ gives its *range*.

At the interface level we use an explicit representation of the lexical level.[4] Therefore, we define a lexicon for our abstract ontology $\mathcal{O}$ as follows:

Definition 2. A *lexicon* for an abstract ontology $\mathcal{O} := (C, \leq_C, R, \sigma, \leq_R, IR)$ is a structure $Lex := (S_C, S_R, Ref_C, Ref_R)$ consisting of

- two sets S_C and S_R whose elements are called *signs (lexical entries) for concepts* and *relations*, resp.,
- and two relations $Ref_C \subseteq S_C \times C$ and $Ref_R \subseteq S_R \times R$ called *lexical reference assignments for concepts/relations*, resp.

[4] Our distinction of lexical entry and concept is similar to the distinction of word form and synset used in WordNet (Fellbaum, 1998). WordNet has been conceived as a mixed linguistic / psychological model about how people associate words with their meaning.

Based on Ref_C, we define, for $s \in S_C$,

$$\mathit{Ref}_C(s) := \{c \in C \mid (s,c) \in \mathit{Ref}_C\}$$

and, for $c \in C$,

$$\mathit{Ref}_C^{-1}(c) := \{s \in S \mid (s,c) \in \mathit{Ref}_C\} \ .$$

Ref_R and Ref_R^{-1} are defined analogously.

The abstract ontology is made concrete through naming. Thus:

Definition 3. A (concrete) *ontology* (in the narrow sense) is a pair $(\mathcal{O}, \mathit{Lex})$ where $\mathcal{O}$ is an abstract ontology and Lex is a lexicon for $\mathcal{O}$.

4.2 Fact knowledge

Fact knowledge may be represented in a variety of ways, such as by tuples from relational database tables, by F-Logic statements, or in the resource description format (RDF)[5], the W3C standard for describing metadata on the WWW. For actual storage, we have the possibility to either store them directly in conventional object-relational databases or in a reified format which builds a layer on top of relational databases.

4.3 Document Representation

We employ three types of document representations distinguishing between term vectors, concept vectors and metadata representations of documents.

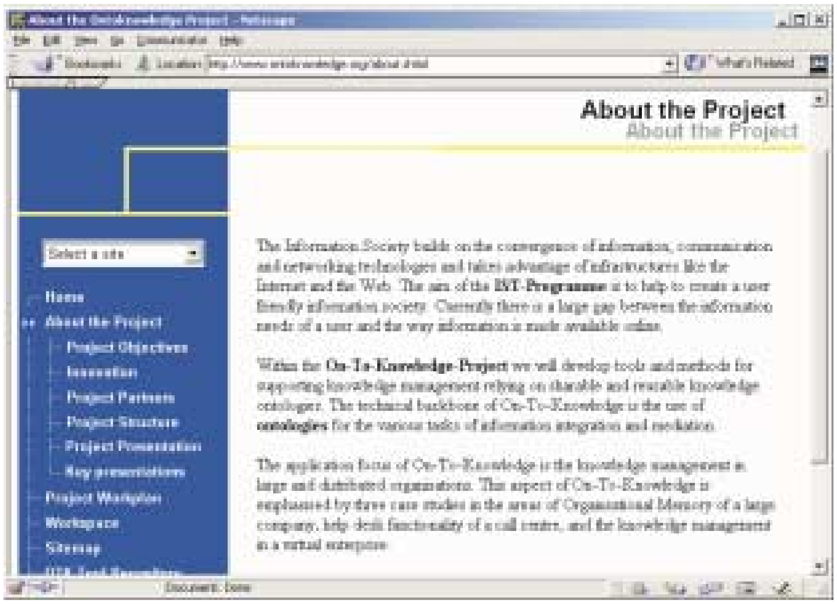
4.3.1 Term vectors

Term vectors (frequently called “bag of words”) describe a document like shown in Figure 4 as a bag of document terms. I.e. one represents the terms that appear in a document together with their frequency in this document. Given the document in Figure 4, the corresponding vector $\mathbf{t} := (1, 2, 0 \ldots)^T$ means that the terms “distributed”, “organization”, and “publication” appear once, twice, and zero times, respectively, in the given document. Thereby, as is standard, stop terms like “the” and “and” or HTML markup like “<title>” or “<p>” are filtered out.

4.3.2 Concept vectors

Term vectors are ideally suited for standard information retrieval methods, such as known from search engines like AltaVista. In restricted domains, however, ontologies may give additional power to retrieval and other tasks by employing available background knowledge. For this purpose, the lexicon is used to map terms to concepts.

[5] http://www.w3.org/RDF/



About the Project

The Information Society builds on the convergence of information, communication and networking technologies and takes advantage of infrastructures like the Internet and the Web. The aim of the **IST-Programme** is to help to create a user friendly information society. Currently there is a large gap between the information needs of a user and the way information is made available online.

Within the **On-To-Knowledge-Project** we will develop tools and methods for supporting knowledge management relying on sharable and reusable knowledge ontologies. The technical backbone of On-To-Knowledge is the use of **ontologies** for the various tasks of information integration and mediation.

The application focus of On-To-Knowledge is the knowledge management in large and distributed organizations. This aspect of On-To-Knowledge is emphasized by three case studies in the areas of Organizational Memory of a large company, help desk functionality of a call centre, and the knowledge management in a virtual enterprise.

Figure 4: Sample of crawled website

Thereby, synonyms may be resolved to refer to a common unique concept. However, word sense ambiguities may make it necessary to represent multiple options, e.g. ones that let “school” stand for the organization vs. the building. Word sense disambiguation methods may be used, however, the current state-of-the-art tools are still rather imprecise.

4.3.3 Metadata representations of documents

Instead of representing document contents, one may gather metadata that *describe* the contents. The standard format for metadata on the Web is the resource description framework (RDF). Essentially, metadata may simply be stored as fact knowledge that is indexed by a document identifier.

5 Ontology-Focused Crawling

A crawler is a program that retrieves Web pages, commonly for use by a search engine (Pinkerton, 1994) or a Web cache. Roughly, a crawler starts off with the URL for an initial page P_0. It retrieves P_0, extracts any URLs in it, and adds them to a queue of URLs to be scanned. Then the crawler gets URLs from the queue (in some order), and repeats the process. Every page that is scanned is given to a client that saves the pages, creates an index for the pages, or summarizes or analyzes the content of the pages.

Our application heavily depends on the detection of relevant data on the web or an intranet. With the rapid growth of the world-wide web new challenges for general-purpose crawlers are given (cf. (Chakrabarti et al., 1999) for a comprehensive framework). Therefore, we have extended classical general-purpose crawlers as currently used in two directions:

1. We follow a focused crawling approach similar to (Chakrabarti et al., 1999). The explicit focus for crawling is given by the ontology.
2. We extend the document crawlers with a component that allows the crawling of relational metadata that may be defined in documents following a given ontology.

5.1 Ontology-focused Document Crawling

The ontology-focused document crawler builds on the general crawling mechanism described above. It extends general crawling by using ontological background knowledge to focus the search in the web space. It takes as input a user-given set A of seed documents (in the form of URLs), a core ontology $\mathcal{O}$, a maximum depth level d_{max} to crawl and a minimal document relevance value r_{min}. The resulting output of the crawler is a set of focused documents D. The crawler downloads each document contained in the set A of start documents. Based on the results of the extraction mechanisms we compute for each document a relevancy measure $r(d)$. In its current implementation this relevancy measure is equal to the overall number of concepts referenced in one document, defined as follows:

Definition 4. Let $L_d := \{l \in S_C \mid l \in d\}$ and $C_d := \{c \in C \mid \exists l \in L_d : (l,c) \in Ref_c\}$. The document relevance value for a document $d \in D$ is given by

$$r(d) = |C_d|. \tag{1}$$

If the relevancy $r(d)$ exceeds the user defined threshold r_{min}, the specific document will be added to the set of focused documents D[6]. All hyperlinks starting from a document d are recursively analyzed. In addition the crawling process is restricted with a maximum depth level d_{max} for a given start document d, i.e. from a seed document maximally d_{max} recursions are followed for crawling.

5.2 Crawling relational metadata

As already mentioned above, our crawler should also extract fact knowledge in the form of relational metadata if available on web pages. Therefore, we have developed

[6] The reader may note that this strategy for measuring relevancy may be further refined, e.g. with normalized counts or with the inclusion of ontological background knowledge, e.g. contained in the concept taxonomy.

the RDF Crawler[7], a basic tool that gathers interconnected fragments of RDF from the Web and builds a local knowledge base from this data. The RDF crawler builds on RDF. In general, RDF data may appear in Web documents in several ways. We distinguish between *(i)* pure RDF (files that have an extension like "*.rdf"), *(ii)* RDF embedded in HTML and *(iii)* RDF embedded in XML. Our RDF Crawler relys on Melnik's RDF-API[8] that can deal with the different embeddings of RDF described above.

6 Ontology-based Clustering

The objective of document clustering is to present the user a high-level structured view for navigation through mostly unknown terrain. Thus, he may find associations between seemingly unrelated documents and get an intuition about the structure of the document repository that has not been crafted manually. Thereby, the document clustering algorithms work by determining (dis-)similarity of documents, e.g. based on the Euclidean or based on the cosine distance of document term vectors or document concept vectors. More similar documents are put into the same cluster and documents that appear in different clusters are considered to be rather dissimilar. Together, these features, i.e. unsupervised structuring of a document repository, which in our case has been crawled from the Web, are highly desirable to acquaint the user of the portal with the rather unstructured document knowledge.

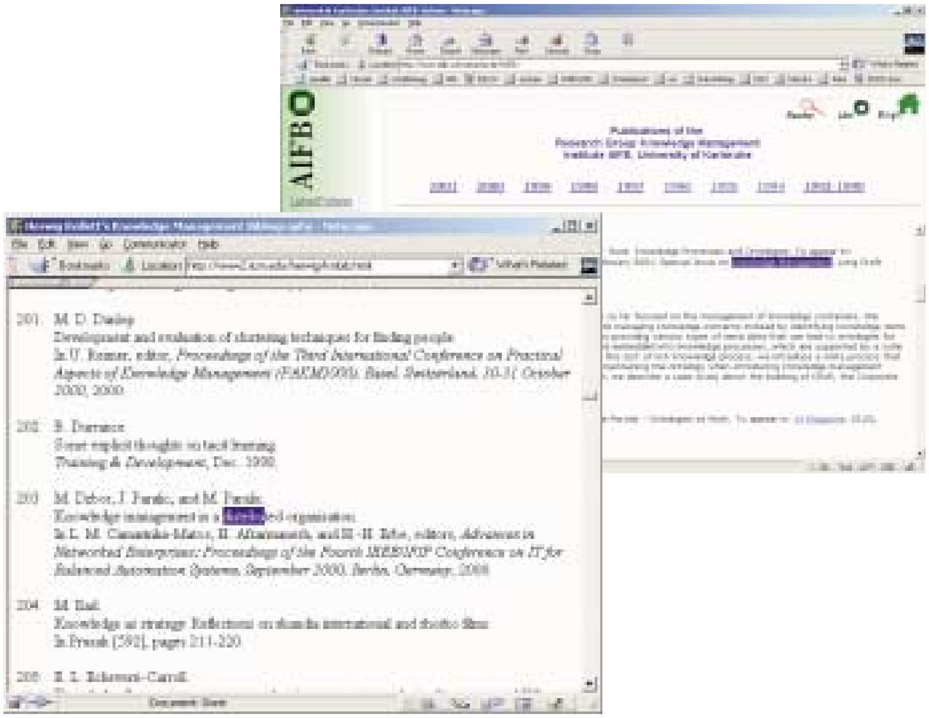
Though standard mechanism for text clustering are well known, they typically suffer from several inherent problems. First, the term/concept vectors are too large. Therefore clustering takes place within a high-dimensional vector space leading to undesirable mathematical consequences, viz. all document pairs are similarly (dis-)similar. Thus, clustering becomes impossible and yields no recognizable results (Beyer et al., 1999). Second, it is hard for the user to understand the differences between clusters. Third, background knowledge does not influence the interpretation of structures found by the clustering algorithms. Therefore, we have developed an ontology-based clustering approach that (partially) solves these problems (Hotho et al., 2001).

Document #	1 ("OTK")	2 ("AIFB Publications")	3 ("IICM Publications")
Publication	0	1	1
Knowledge Management	2	2	1
Distributed Organization	1	0	1

Figure 5: Concept vector representations for 3 sample documents (cf. Figs. 4 and 6)

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

[8] http://www-db.stanford.edu/∼melnik/rdf/api.html


AIFB
Publications of the
Research Group Knowledge Management
Institute AIFB, University of Karlsruhe
2001 2000 1999 1998 1997 1996 1995 1994 1993-1990
Knowledge Management Bibliography - Netscape
File Edit View Go Communicator Help
Bookmarks Location
What's Related
Document: Done

201 M. D. Dunlop
Development and evaluation of clustering techniques for finding people
In U. Reimer, editor, *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management (PAKM2000), Basel, Switzerland, 30-31 October 2000*, 2000.

202 B. Durrance
Some explicit thoughts on tacit learning
Training & Development, Dec. 1998.

203 M. Debor, J. Fardo, and M. Fardo
Knowledge management in a distributed organization
In L. M. Camarinha-Matos, H. Afsarmanesh, and H.-H. Erbe, editors, *Advances in Networked Enterprises: Proceedings of the Fourth IEEE/IFIP Conference on IT for Balanced Automation Systems, September 2000, Berlin, Germany*, 2000.

204 M. Earl
Knowledge as strategy: Reflections on skandia international and shorko films
In Prusak [392], pages 211-220.

205 E. L. Eckerson-Carroll

Figure 6: Sample web pages

In the following we give a detailed example. In Figure 5 you find a sample of (abbreviated) concept vectors representing the web pages depicted in Figures 4 and 6. In Figure 7 you see the corresponding concepts highlighted in an excerpt of the FZI-Broker ontology (also cf. section 8 on FZI-Broker). Our simplifying example shows the principal problem of vector representations of documents: The tendency that spurious appearance of concepts (or terms) rather strongly affects the clustering of documents. The reader may bear in mind that our simplification is so extensive that practically it does not appear in such tiny settings, but only when one works with large representations and large document sets. In our simplifying example the appearance of concepts PUBLICATION, KNOWLEDGE MANAGEMENT, and DISTRIBUTED ORGANIZATION is spread so evenly across the different documents that all document pairs exhibit (more or less) the same similarity. Corresponding squared Euclidian distances for the example document pairs (1,2), (2,3), (1,3) leads to values of 2, 2, and 2, respectively, and, hence, to no clustering structure at all.

When one reduces the size of the representation of our documents, e.g. by projecting into an subspace, one focuses on particular concepts and one may focus on the significant differences that documents exhibit with regard to these concepts. For

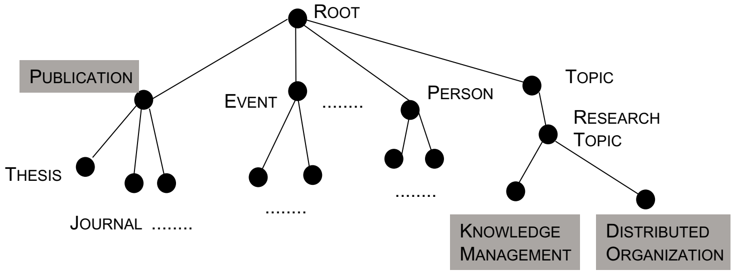


Figure 7: A sample ontology

instance, when we project into a document vector representation that only considers the two dimensions PUBLICATIONS and KNOWLEDGE MANAGEMENT, we will find that document pairs (1,2), (2,3), (1,3) have squared Euclidean distances of 1, 1, and 2. Thus, axis-parallel projections like in this example may improve the clustering situation. In addition, we may exploit the ontology. For instance, we select views from the taxonomy, choosing, e.g., RESEARCH TOPIC instead of its subconcepts KNOWLEDGE MANAGEMENT and DISTRIBUTED ORGANIZATION. Then, the entries for KNOWLEDGE MANAGEMENT and DISTRIBUTED ORGANIZATION are added into one vector entry resulting in squared Euclidean distances between pairs (1,2), (2,3), (1,3) of 2, 0, and 2, respectively. Thus, documents 2 and 3 can be clustered together, while document 1 falls into a different cluster.

In (Hotho et al., 2001) we have developed an algorithm ”GenerateConceptViews” as a preprocessing step for clustering. GenerateConceptViews chooses a set of axis-parallel and ontology-based projections leading to modified document representations. Conventional clustering algorithms like K-Means may work on these modified representations producing improved clustering results. Because the size of the vector representation is reduced, it becomes easier for the user to track the decisions made by the clustering algorithms. Because there are a variety of projections, the user may choose between views. For instance, there are projections such that publication pages are clustered together and the rest is set aside or projections such that web pages about KNOWLEDGE MANAGEMENT are clustered together and the rest is left in another cluster. The choice of concepts from the taxonomy thus determines the output of the clustering result and the user may use a view like Figure 7 in order to select and understand differences between clustering results.

Concluding this section, we may claim that the particular merits of ontology-based clustering are the improvement of clustering results (as can be proved by standard evaluation measures), the explanation of results, and the consideration of background knowledge. The general merits of ontology-based clustering in SEAL-II are its capabilities to relate the ontology with unstructured knowledge. Thus, it provides the portal with abilities for unsupervised, automatic structuring of portal contents. Users of SEAL-II may approach large document sets more conveniently and efficiently select views that correspond to concepts of actual concern to the user.

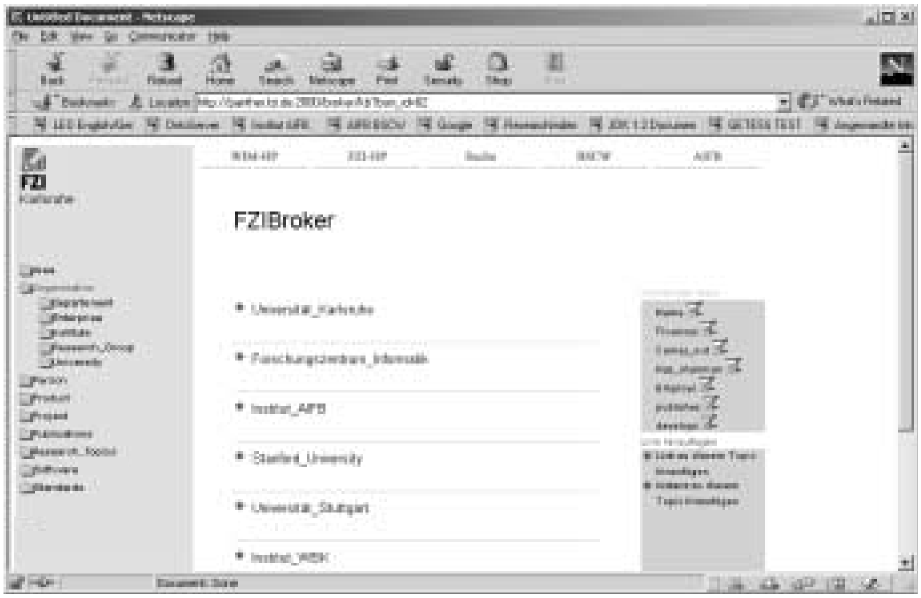
7 Open Conceptual Hypermedia

SEAL-II relies on ontologies to structure the available information and to define machine-readable metadata for the information sources at hand. By exploiting the concepts and relations being defined in the ontology, links between different knowledge elements that are presented to the user may be handled as first class objects. In that way, links are managed separately from the contents, a characteristic of Open Hypermedia Systems (Osterbye & Wiil, 1996). Furthermore, the relations of the ontology provide a conceptual underpinning of these links and thus pave the way to so-called Conceptual Hypermedia Systems (cf. (Nanard & Nanard, 1991)).
The conceptual model that is defined by the ontology may directly be transformed into functionalities of the portal user interface:

- Conceptual navigation: In contrast to conventional hypermedia systems, i.e. hypermedia systems that just provide syntactic links between hypermedia documents, links within Conceptual Hypermedia Systems come with a clearly defined semantics, as specified by the corresponding relation of the ontology. In that way, users may navigate along conceptual links that relate knowledge elements to each other. Thus, semantic navigation in the knowledge space is achieved.
- Semantic querying: Based on the concepts of the ontology and the associated relations users may define semantic queries. i.e. queries that come with a clearly defined semantics. As a consequence, the portal delivers an exact answer containing the knowledge elements the user is interested in.

A second advantage of an ontology-based approach is the exploitation of the ontology for the generation of the portal. This is a crucial aspect since the manual construction of a portal is a time- and resource-consuming task. The automatic generation of the portal has to address the following aspects (cf. Figure 8):

- Navigation structure: The concepts of the ontology and their embedding into a taxonomy provide a basic conceptual navigation structure for the portal. E.g. in Figure 8 we can see on the left hand side the concept taxonomy. The user may browse in this concept hierarchy until she has found the concept she is interested in. In our example, the user has selected the concept ORGANIZATION.
- Concept instances: In the middle part of the screen, the portal displays the instances being available in the portal for the selected concept. In our example, we see e.g. the instance of organization `Universität_Karlsruhe`.



FZIBroker

- Universitaet_Karlsruhe
- Forschungszentrum_Informatik
- Institut_AIFB
- Stanford_University
- Universitaet_Stuttgart
- Institut_WBK

Figure 8: Screenshot FZI-Broker

– Relational (Meta-)data: The conceptual navigation structure is supplemented with relations being defined in the ontology for the selected concept. I.e. the portal presents exactly those relations that are available in the current state of the user navigation. E.g. in Figure 8 we see all the relations being defined for the concept ORGANIZATION. By selecting one of the displayed relations the user is able to specify instances of this relation for the selected concept instance. In our example, the user could e.g. specify a (meta-)data value for the CARRIES_OUT-relation for the project `OntoKnowledge`. An obvious advantage of this approach is the fact that the user is only able to specify relational (meta-)data facts that are consistent with the conceptual model, i.e. the ontology.

As can be seen from the example in Figure 8, SEAL-II supports the generation of portals with limited complexity from a given ontology. Since the SEAL-II framework is domain independent, the generation of a new portal just requires the construction of a new ontology. Obviously, one might think of more sophisticated means, e.g. of tailoring the interface to specific user profiles or of defining a subset of the ontology as the conceptual navigation ontology. Such aspects are discussed in the conclusion.

8 Instantiations of SEAL-II

In its current version, the SEAL-II architecture has been instantiated in two applications: FZI-Broker[9] and HR-TopicBroker. HR-TopicBroker is a system that supports the location of human resource (HR) topics in relevant web pages. Additionally, it allows the joint definition of a knowledge base by human resource managers sharing relevant information (e.g. contact addresses).

Along the same lines FZI-Broker as an realization of SEAL-II is a system that is internally used by the FZI knowledge management research group. A screenshot of the FZI-Broker application is depicted in Figure 8. The FZI-Broker ontology is based on the Semantic Web Research Community (SWRC) ontology. The ontology contains information like ACADEMICSTAFF is a subclass of EMPLOYEE and ACADEMICSTAFF has, e.g., the relation COOPERATESWITH.

The underlying idea of FZI-Broker is that members of the research group share a common ontology with common interests. Thus, FZI-Broker supports the instantiation of these common interests. First, it supports a document-centric view on the ontology. It uses the focused document crawler and the clustering mechanism to automatically offer views on web documents. Additionally, as depicted in Figure 8 it allows the manual definition of fact knowledge in the form of concept and relation instantiations following the ontology definitions.

9 Related Work

This section positions our work, the SEAL-II approach, 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 semantic community web portals.

Related Work on Portals. One of the well-established web portals is Yahoo[10]. In contrast to our approach Yahoo only utilizes a very light-weight ontology that solely consists of categories manually arranged in a hierarchical manner. Yahoo offers keyword search in addition to hierarchical navigation. SEAL-II offers a much wider range of technology for access to documents *and* facts.

The COHSE system (Carr et al., 2001) is a system integrating notions from Open and Conceptual Hypermedia Systems. COHSE eploits ontologies to generate conceptual links between documents and to associate metadata with documents. Currently, ontologies are restricted to thesauri offering broader-term, narrower-term and related-term relations. By enriching documents with conceptual links COHSE provides extra information and linking for existing web pages, i.e. COHSE puts emphasis on the linkage and navigation aspects between web pages. The link generator module of COHSE

[9] A demo version of FZI-Broker is available at http://panther.fzi.de:2000/demofzibroker.

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

and the concept vector representation of SEAL-II use similar techniques for associating terms in documents with concepts from the ontology. In contrast to COHSE, we also put emphasis on the querying aspects of a portal. Furthermore, we rely on a “heavy-weight” ontology being exploited by our Ontobroker inference engine (Decker et al., 1999).

The Broker’s Lounge approach (Jarke et al., 2001) provides methods and tools for user-adaptive knowledge management putting emphasis on contextualization and conceptualization. Personalization is defined by interest profiles that are based on the concepts of the domain model and their categorization into different types. Whereas the Broker’s Lounge puts special emphasis on contextualization of knowledge, SEAL-II sets up an overall portal framework offering a smooth integration of ontology-based and information retrieval based techniques.

The Ontobroker system and the knowledge warehouse (Decker et al., 1999) lay the semantical foundations for the SEAL-II approach. The approach closest to Ontobroker is SHOE (Heflin & Hendler, 2000). 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 techniques of both systems differ significantly: SHOE offers only very limited inferencing capabilities, whereas Ontobroker relies on Frame-Logic and thus supports complex inferencing for query answering. A more detailed comparison to other portal approaches and underlying methods may be found in (Staab et al., 2000).

Related Work on Focused Crawling. The need for focused crawling in general has recently been realized by several researchers. The main target of all of these approaches is to focus the search of the crawler and to enable goal-directed crawling. In general a focused crawler takes a set of well-selected web pages exemplifying the user interest. (Chakrabarti et al., 1999) present a generic architecture of a focused crawler. The crawler uses a set of predefined documents associated with topics in a Yahoo like taxonomy to build a focused crawler. Two hypertext mining algorithms build the core of their approach: a classifier that evaluates the relevance of a hypertext document with respect to the focus topics, and a destiller, that identifies hypertext nodes that are access points to many relevant pages within a few links. The approach presented in (Diligenti et al., 2000) uses so-called context graphs as a means to model the paths leading to relevant web pages. Context graphs in their sense represent link hierarchies within which relevant web pages occur together in the context of such pages. (Rennie & McCallum, 1999) propose a machine learning oriented approach for focused crawling. Their crawler uses reinforcement learning to learn to choose the next link such that reward over time is maximized. A problem of their approach is that the method requires large collections of already visited web pages.

In contrast to our approach these crawlers do not include relevancy that exploits lexical & ontological information for focusing the search. Additionally, none of the crawlers above supports the combined crawling of documents and metadata.

Related Work on Clustering. All clustering approaches based on frequencies of terms/ concepts and similarities of data points suffer from the same mathematical properties of the underlying spaces (cf. (Beyer et al., 1999; Hinneburg et al., 2000)). These properties imply that even when "good" clusters with relatively small mean squared errors can be built, these clusters do not exhibit significant structural information as their data points are not really more similar to each other than to many other data points. Therefore, we derive the high-level requirement for text clustering approaches: either they should rely on much more background knowledge (and thus can come up with new measures for similarity) or they should cluster in subspaces of the original high dimensional representation.

In general, existing approaches (e.g., (Agrawal et al., 1998; Hinneburg & Keim, 1999)) on subspace clustering face the dual nature of "good quality". On the one hand, there are sound statistical measures for judging quality. State-of-the-art methods use them in order to produce "good" projections and, hence, "good" clustering results, for instance:

- Hinneburg & Keim (Hinneburg & Keim, 1999) show how projections improve the effectiveness and efficiency of the clustering process. Their work shows that projections are important for improving the performance of clustering algorithms. In contrast to our work, they do not focus on cluster quality with respect to the internal structures contained in the clustering.
- The problem of clustering high-dimensional data sets has been researched by Agrawal et al. (Agrawal et al., 1998): They present a clustering algorithm called CLIQUE that identifies dense clusters in subspaces of maximum dimensionality. Cluster descriptions are generated in the form of minimized DNF expressions.
- A straightforward preprocessing strategy may be derived from multivariate statistical data analysis known under the name principal component analysis (PCA). PCA reduces the number of features by replacing a set of features by a new feature representing their combination.
- In (Schuetze & Silverstein, 1997), Schuetze and Silverstein have researched and evaluated projection techniques for efficient document clustering. They show how different projection techniques significantly improve performance for clustering, that is not accompanied by a loss of cluster quality. They distinguish between local and global projection, where local projection maps each document onto a different subspace, and, global projection selects the relevant terms for all documents using latent semantic indexing.

Now, on the other hand, in real-world applications the statistically optimal projection, such as used in the approaches just cited, often does not coincide with the projection most suitable for humans to solve a particular task, such as finding the right piece of knowledge in a large set of documents. Users typically prefer explicit background knowledge that indicates the foundations on which a clustering result has been achieved.

Hinneburg et al. (Hinneburg et al., 1999) consider this general problem as a domain specific optimization task. Therefore, they propose to use a visual and interactive environment that involves the user to derive meaningful projections. Our approach automatically solves some part of the task they assign to the user environment automatically: we give the user some first means to explore the result space interactively in order to select the projection most relevant for her particular objectives.

Finally, we want to mention an interesting proposal for feature selection made in (Devaney & Ram, 1998). Devaney and Ram describe feature selection for an unsupervised learning task, namely conceptual clustering. They discuss a sequential feature selection strategy based on an existing COBWEB conceptual clustering system. In their evaluation they show that feature selection significantly improves the results of COBWEB. The drawback that Devaney and Ram face, however, is that COBWEB is not scalable like K-Means. Hence, for practical purposes of clustering in large document repositories, our approach seems better suited.

10 Conclusion

We have shown in this paper how to extend our methodology for semantic portals, SEAL, into SEAL-II, such that the range of possible uses is extended in several ways: First, the initial start-up process of the portal becomes easier. Second, the overall approach is made more flexible creating synergy between unstructured resources and ontology available as background knowledge. Third, we have started to investigate the architecture of an overarching framework concerning the use of ontologies.

Let us briefly elaborate on the latter. In a typical application the ontology has most often played only a single role. We, however, see ontologies as a means to structure content that may be engaged in a plethora of means even within one single application. The arrival of readily configurable software modules (like “products” in the application server ZOPE) makes it *practically* much more feasible to exploit the ontology for such broad variations as content structuring, content selection, content presentation, and content providing. We envision a situation where we will have a collection of various ontologies that are clearly aligned to each other and that are used to support these functionalities.

The technology that we have discussed here is very general. Obvious uses exist for knowledge management. In fact, its very initial conception was triggered by a KM application for DaimlerChrysler Human Resource Management, i.e. the HR-TopicBroker. However, many more and very different applications are probably just appearing on the horizon right now.

Acknowledgements.

Research reported in this paper has been partially financed by BMBF in the project GETESS (01IN901C0), by US AirForce in the DARPA DAML OntoAgents project, by EU in the IST project On-To-Knowledge (IST-1999-10132) and by DaimlerChrysler in the project “HR-TopicBroker”. We thank our student Lars Kuehn and our colleagues, in particular Kalvis Apsitis, Nenad Stojanovic, Gerd Stumme, York Sure and Raphael Volz for discussions and work that influenced parts of this paper, e.g. our notion of ontologies, the RDF crawler (written by Kalvis), and the overall architecture. We thank Klaus Goetz from DaimlerChrysler for discussions that influenced our thinking on information brokering.

References

Agrawal, R., Gehrke, J., Gunopulos, D., & Raghavan, P. (1998). Automatic subspace clustering of high dimensional data for data mining applications. In *Proceedings of the ACM SIGMOD Int’l Conference on Management of Data, Seattle, Washington*, pages 94–105. ACM Press.

Angele, J., Schnurr, H.-P., Staab, S., & Studer, R. (2000). The times they are a-changin’ — the corporate history analyzer. In Mahling, D. & Reimer, U. (Eds.), *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland*, pages 1–1 – 1–11. http://research.swisslife.ch/pakm2000/.

Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is ‘nearest neighbor’ meaningful. In *Proceedings of ICDT-1999, Jerusalem, Israel*, pages 217–235. ACM Press.

Carr, L., Bechhofer, S., Goble, C., & Hall, W. (2001). Conceptual linking: Ontology-based open hypermedia. In *WWW10 — Proceedings of the 10th International World Wide Web Conference, Hong Kong, China*, pages 334–342. ACM Press.

Chakrabarti, S., van den Berg, M., & Dom, B. (1999). Focused crawling: a new approach to topic-specific web resource discovery. In *WWW8 - Proceedings of the 8th World Wide Web Conference, Toronto, Canada*, pages 545–562. Elsevier.

Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1999). Ontobroker: Ontology based access to distributed and semi-structured information. In Meersman, R. et al. (Eds.), *Database Semantics: Semantic Issues in Multimedia Systems, Boston, USA*, pages 351–369. Kluwer Academic Publisher.

Devaney, M. & Ram, A. (1998). Efficient feature selection in conceptual clustering. In *Proceedings 14th International Conference on Machine Learning, Nashville, TN*, pages 92–97. Morgan Kaufmann.

Diligenti, M., Coetzee, F. M., Lawrence, S., Giles, C. L., & Gori, M. (2000). Focused crawling using context graphs. In *Proceedings of the 26th International Conference on Very Large Databases (VLDB-00), Cairo, Egypt*, pages 527–534. Morgan Kaufmann.

Eriksson, H., Puerta, A., & Musen, M. A. (1994). Generation of knowledge-acquisition tools from domain ontologies. *International Journal of Human Computer Studies(IJHCS)*, 41:425–453.

Fellbaum, C. (1998). *WordNet – An electronic lexical database*. MIT Press, Cambridge, Massachusetts and London, England.

Fensel, D., van Harmelen, F., Horrocks, I., McGuinness, D. L., & Patel-Schneider, P. F. (2001). OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–44.

Grosso, W. E., Eriksson, H., Fergerson, R. W., Gennari, J. H., Tu, S. W., & Musen, M. A. (1999). Knowledge modeling at the millennium (the design and evolution of Protégé-2000). In *Proceedings of the 12th Workshop for Knowledge Acquisition, Modeling and Management (KAW'99), Banff, Canada*.

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

Heflin, J. & Hendler, J. (2000). Searching the web with SHOE. In *Artificial Intelligence for Web Search. Papers from the AAAI Workshop. WS-00-01, Menlo Park, CA*, pages 35–40. AAAI Press.

Hinneburg, A., Aggarwal, C., & Keim, D. A. (2000). What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Databases (VLDB-00), Cairo, Egypt*, pages 506–515. Morgan Kaufmann.

Hinneburg, A. & Keim, D. A. (1999). Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In *Proceedings of the 25th International Conference on Very Large Databases (VLDB-99), Edinburgh, Scotland*, pages 506–517. Morgan Kaufmann.

Hinneburg, A., Wawryniuk, M., & Keim, D. A. (1999). Hd-eye: visual mining of high-dimensional data. *IEEE Computer Graphics and Applications*, 19(5):22–31.

Hotho, A., Maedche, A., & Staab, S. (2001). Ontology-based text clustering. In *Proceedings of the IJCAI-2001 Workshop "Text Learning: Beyond Supervision", August, Seattle, USA*.

Jarke, M., Klemke, R., & Nick, A. (2001). Broker's lounge — an environment for multi-dimensional user-adaptive knowledge management. In *HICSS-34: 34th Hawaii International Conference on System Siences, Maui, Hawaii*, pages 83–83. http://fit.gmd.de/ klemke/.

Kifer, M., Lausen, G., & Wu, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42:741–843.

Maedche, A., Staab, S., Stojanovic, N., & Studer, R. (2001). SEAL - A Framework for Developing SEmantic portALs. In *Proceedings of the 18th British National Conference on Databases, July, Oxford, UK*, LNCS. Springer.

Nanard, J. & Nanard, M. (1991). Using structured types to incorporate knowledge in hypertext. In *Proceedings of the ACM Hypertext Conference, San Antonio, Texas, USA*, pages 329–343. ACM Press.

Osterbye, K. & Wiil, U. (1996). The flag taxonomy of open hypermedia systems. In *Proceedings of the ACM Conference on Hypertext, Washington, DC*, pages 129–139. ACM Press.

Pinkerton, B. (1994). Finding what people want: Experiences with the webcrawler. In *WWW2 — Proceedings of the 2nd International World Wide Web Conference, Chicago, USA*.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Rennie, J. & McCallum, A. (1999). Using reinforcement learning to spider the web efficiently. In *Proceedings of the 16th International Conference on Machine Learning (ICML-99), Bled, Slovenia*, pages 335–343.

Schuetze, H. & Silverstein, C. (1997). Projections for efficient document clustering. In *Proceedings of SIGIR-1997, Philadelphia, PA*, pages 74–81. Morgan Kaufmann.

Staab, S., Angele, J., Decker, S., Erdmann, M., Hotho, A., Maedche, A., Schnurr, H.-P., Studer, R., & Sure, Y. (2000). Semantic community web portals. In *WWW9 — Proceedings of the 9th International World Wide Web Conference, Amsterdam, The Netherlands*, pages 473–491. Elsevier.

Staab, S., Braun, C., Düsterhöft, A., Heuer, A., Klettke, M., Melzig, S., Neumann, G., Prager, B., Pretzel, J., Schnurr, H.-P., Studer, R., Uszkoreit, H., & Wrenger, B. (1999). GETESS — searching the web exploiting german texts. In *Proceedings of the 3rd Workshop on Cooperative Information Agents, Uppsala, Sweden*, LNCS, pages 113–124. Springer.

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

Staab, S. & Maedche, A. (2001). Knowledge portals — ontologies at work. *AI Magazine*, 21(2).

Sure, Y., Maedche, A., & Staab, S. (2000). Leveraging corporate skill knowledge - From ProPer to OntoProper. In Mahling, D. & Reimer, U. (Eds.), *Proceedings of the Third International Conference on Practical Aspects of Knowledge Management, Basel, Switzerland*, pages 22–1 – 22–9. http://research.swisslife.ch/pakm2000/.