# Ontologies for Knowledge Retrieval in Organizational Memories

**Minghong Liao, Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, and Michael Sintek**

German Research Center for Artificial Intelligence–DFKI GmbH
P.O.Box 2080, D-67608 Kaiserslautern, Germany
Phone:+49 631 205 3467, Fax:+49 631 205 3210
{liao,aabecker,bernardi,hinkelma,sintek}@dfki.uni-kl.de

**Abstract.** An Organizational Memory (OM) captures, stores and disseminates valuable corporate knowledge and is thus a central prerequisite for enterprise knowledge management. For structuring, accessing, and maintaining large amounts of heterogeneous information, appropriate meta-level descriptions are needed which specify the structure, content, and potential usage of the object-level knowledge. Such meta-level descriptions are provided for data in the form of data models, for formal knowledge as ontologies, and for informal documents as document descriptors. In this paper, we sketch an ontology-based approach for comprehensive meta-modeling and retrieval of heterogeneous data, formal knowledge, and documents. We identify information ontology, domain ontology, and enterprise ontology as main contributors to a vocabulary for comprehensive information meta modeling. We elaborate a bit on the underlying representation formalism, sketch a sample scenario, and present ontology-based heuristic retrieval in the organizational memory.

**Keywords:** knowledge representation and retrieval,
web-based knowledge management

## 1 Introduction

The systematic management of knowledge has been recognized as a necessity to enhance a company's survival and success in the global market place. To be effective, organizational knowledge management has to improve the capitalization on existing knowledge assets and to facilitate the creation of new knowledge. An Organizational Memory (or Corporate Memory, OM) can be characterized as a comprehensive computer system which captures a company's accumulated know-how and other knowledge assets, and makes them available to enhance the efficiency and effectiveness of knowledge-intensive work processes [Kühn and Abecker, 1997].

A couple of years before the big Knowledge Management hype in the business and management sciences, a very similar concept had already been introduced in the Software Engineering community under the name "Experience Factory" (EF) [Basili *et al.*, 1994]. Recently, [Althoff *et al.*, 1998] showed how the idea of an experience factory can be supported by the use of case-based reasoning (CBR) technology for storage and reuse of documents, designs, code, and other artifacts in the Learning Software Organisation.

A main design issue for an OM (and, consequently, also for the EF) are the respective roles of formal knowledge compared to semi-structured or non-formal documents. Many application-oriented authors agree on the fact that semi-structured and non-formal documents are playing a predominant role in a company's knowledge management (see, e.g., [Choo, 1995,Hartley *et al.*, 1997]). [Abecker *et al.*, 1998b] propose the OM as sort of a "meta information system" providing a uniform access to a diversity of knowledge and information sources of different degree of formality. Since formalization is costly, error-prone, and requires extensive maintenance efforts later on, they propose to use formalized knowledge mainly for coupling task and retrieval, and for supporting precise-content retrieval to the OM.

Quite a similar point of view can be found in Figure 1 describing Richter's view on the CBR approach (taken from [Althoff *et al.*, 1998]). There, only very stable (vocabulary), useful (similarity measure), or worthful (solution transformation) knowledge is codified into formal representations, here referred to as "compiled knowledge".

The remainder is left in the cases.

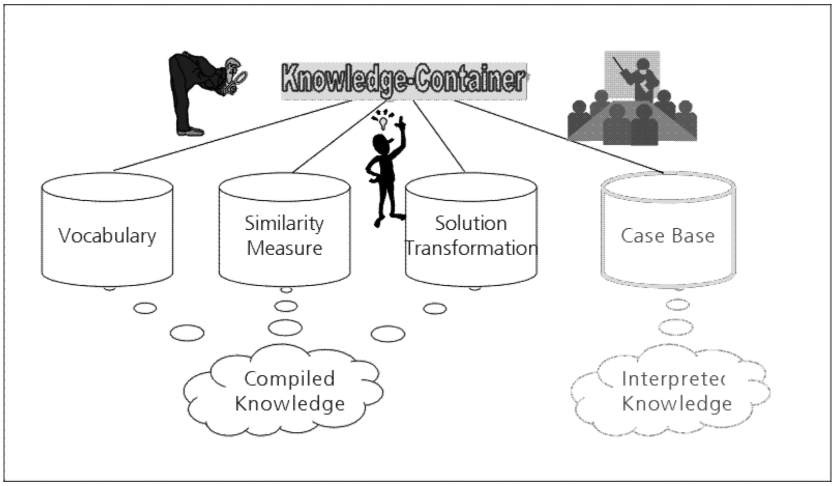


**Fig. 1.** The knowledge container view on CBR systems (taken from [Althoff *et al.*, 1998], adapted from [Richter, 1995])

Coming from the knowledge-based system point of view, [Benjamins *et al.*, 1998] distinguish two widespread approaches to building knowledge management systems. *Vertical systems* are task-specific performance support systems. They can provide high value in particular business situations because they incorporate much application specific knowledge. Naturally, their usage is restricted to a narrow application environment. *Horizontal systems*, on the other hand, are general frameworks for providing useful corporate knowledge in a wide area of application situations. In practice, this approach essentially amounts to more or less intelligent document management and information retrieval systems.

In their own approach, [Benjamins *et al.*, 1998] propose formal ontologies to allow for comfortable access and knowledge intensive usage of data and information embedded in HTML pages annotated with ontological information. Numerous other approaches propose formal ontologies to ease finding of and access to data and semi-structured information in HTML pages and databases [Luke *et al.*, 1997].

If one understands an information retrieval process as a similarity assessment between query situation and document description, both the CBR and the KBS point of view come together (except for the solution transformation, which is often neglected in practical applications). Further, if one fills the "knowledge containers" of the CBR approach with the heterogeneous information sources available in an enterprise, understands the case retrieval as a logical inference process on the basis of ontologies and meta data (as at least roughly proposed already by [Kamp, 1996]), and equips the resulting system not only with one similarity measure, but with a library of application programs (consisting of vocabulary, similarity measure, solution transformation), the result may be one step towards a system which encompasses the depth of system services provided by vertical systems and the breadth of usage scenarios applicable to horizontal systems.

Although we will not further elaborate the analogue to the CBR point of view, this is the ultimate aim of the work described here. Essentially, we ask what ontologies are required to have appropriate "containers" and to define useful "similarity measures", how to represent such ontologies, and how to use them.

## 2 Overall Organizational Memory Design

The organizational memory of the KnowMore project [Abecker *et al.*, 1998a] is an enterprise-internal application-independent information and assistant system. It stores data, information, and knowledge from different sources of an enterprise. They are represented in various forms, such as databases, documents, and formal knowledge-bases. It will be permanently extended to keep it up to date and accessible enterprise-wide through an appropriate network infrastructure. A three-layered model as sketched in Figure 2 [Abecker *et al.*, 1998b].

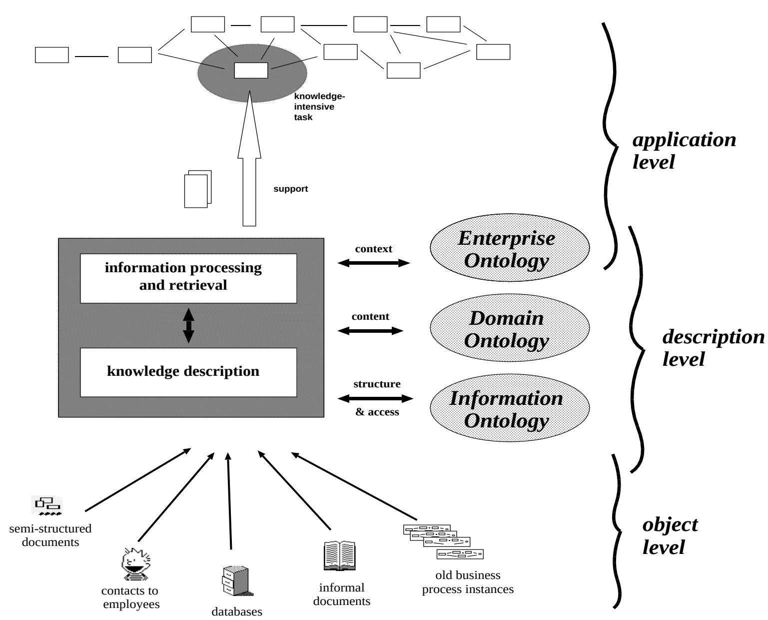


**Fig. 2.** The Organizational Memory Model

The *object level* comprises manifold information and knowledge sources, ranging from machine-readable formal representations to human-readable informal representations. Crucial parts of corporate knowledge to be processed by the computer must be formalized, whereas other parts that need only be understood by humans might be left informal. The decision whether to formalize or not rests on cost-benefit analyses, stability of knowledge, and the question whether some portion of knowledge can reasonably be formalized at all.

The *knowledge description level* enables a uniform, intelligent access to the diversity of object-level sources. Because legacy information systems must be incorporated without modification, we propose a separate, knowledge-rich information-modeling level, the details of which will be the main focus of this paper.

The OM's *application level* links the information model and the concrete application situation. When a knowledge worker recognizes an information need within the actual flow of work, a query to the OM must be derived. This query is instantiated and constrained as specifically as possible on the basis of the actual work context. In the opposite way, the OM can also store new information created within a given working situation in a contextually enriched form such that subsequent retrieval processes might compare the query situation with the creation situation for estimating context-specific relevance. As one of many possibilities for realizing the application level, we include conventional business-process models and workflow-management systems. Doing so lets us rely on a body of well-understood knowledge already formalized in enterprises and used to guide and support work processes.

## 3 Construction of the Ontologies

### 3.1 Three basic ontologies

Every information and knowledge item is described by a number of attributes representing the information metamodel, the information content, and the creation and application context. The concepts necessary for these descriptions form the fundamental ontologies for this modeling. Based on their specific roles in the context of an OM it is useful to distinguish three different ontologies (see Figure 3).

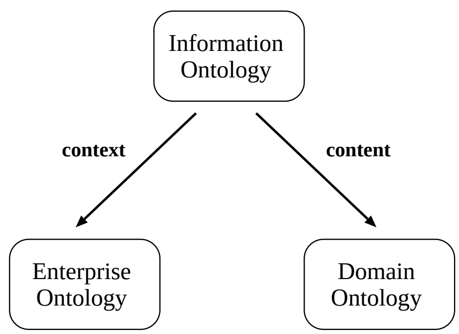


**Fig. 3.** Three dimensions of knowledge description.

The *information ontology* describes the vocabulary of the information metamodel, which characterizes the different kinds of information sources with their respective structure, access, and format proporties. The concepts in the information ontology are stable and domain independent: a book has author, title, etc. as meta data, consists of chapters, and so on. This is always the case regardless of the concrete use of this information source. Specific instances of this ontology, i.e., descriptions of particular information sources, need to refer to concepts from the enterprise and the domain ontologies, respectively.

The *enterprise ontology* is used to describe information context, which is expressed as organizational structure and process models. Both the context of the creation of some information element and the context of its intended use are important contributions when evaluating the relevance of some information element with respect to a particular task at hand. Thus this context information plays a crucial role within the knowledge handling in an organizational memory. The concepts in the enterprise ontology are expected to be rather independent of an actual company; various projects have provided standard enterprise ontologies which are applicable for most enterprises. However, they are of a different league than the information ontology concepts, thus the distinction between the two ontologies.

The *domain ontology* is used to model the content of the information sources. Typically, the concepts in this ontology are highly specific for a particular application. Thus it has to be taken into account that for the realization of a particular OM that it might be necessary to construct the domain ontology from scratch. However, there have been substantial efforts in the Knowledge Sharing community for providing reusable, task-independent real-world domain ontologies, e.g., in the areas of chemistry [van der Vet and Mars, 1993] or materials science [van der Vet *et al.*, 1995].

In order to emphasize the engineering aspect of ontology research, it would be an interesting exercise to find out how easy it is to (i) take such an "off-the-shelf" ontology which describes the domain of activity of a big company, e.g., in the chemistry sector, (ii) combine it with a preexisting enterprise and information ontology, (iii) put the pieces together, adapt them to the specific needs of the company and assess how much effort it is to configure such an information system from standard ontology modules. Our identification of the basic ontological dimensions information structure and meta data, static and dynamic enterprise context, and application domain should ease this enterprise a bit.

To repeat, a concrete information element is described via concepts taken from the information ontology, where the various attributes are filled with concepts from the enterprise and domain ontology, respectively. This interplay is illustrated in Figure 4 where the various concept types and their interrelations as used in the KnowMore OM are shown.

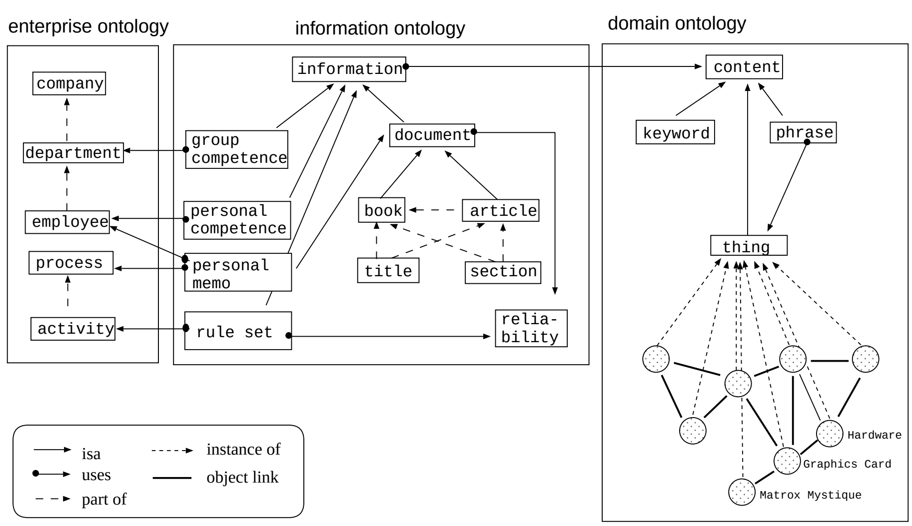


**Fig. 4.** Sketch of sample ontologies.

### 3.2 Ontology representation

Although since the very first AI related work on ontologies, there have also been representation languages (of which Ontolingua is the most prominent one

[Gruber, 1993]), most researchers concerned with ontology-based retrieval nevertheless used their own representation formats, e.g., Frame-Logic for the sake of powerful inferences [Fensel *et al.*, 1998], SHOE for the sake of efficient reasoning [Luke *et al.*, 1997], or CKML (the Conceptual Knowledge Markup Language) for the sake of adequate modeling primitives and compliance with de facto standards for the Web and document description [Kent and Neuss, 1996]. It might also be noticed that most of these languages were not originally built for knowledge description or Information Retrieval, but employed in this area. This may indicate that the language debate is far from being finished.

In the following, we will spent few words on the representation requirements we found in our experiments and on our intermediary result, an object-centered relational algebra (OCRA) for knowledge description.

As already the rough sketch of our example (Figure 4) indicates we need some *classification* and *aggregation* (documents consist of parts) mechanism in the information ontology. Further, we need an *instantiation* mechanism to model concrete instances of the respective classes in the concrete information meta models. The same representational means are sufficient for modeling the enterprise ontology. Things get complicated coming to the domain ontology. As, among others, [Lenz, 1998] points out, the weakness of simple keyword-based IR methods is its lack of exploitation of knowledge about domain structures and relationships. Basically, the more domain knowledge I have the more sophisticated retrieval I can get. In the consequence, the aim of modeling the domain of discourse in as much detail as possible leads to the aim of having a maximally powerful, general-purpose knowledge representation language for domain modeling. Consider, for instance, a technical application domain. In the ESB[1] project [Bernardi *et al.*, 1998], we built a sophisticated domain model for intelligent management and retrieval of natural-language based records of maintenance experiences with a highly-complex machine. There, it was not only necessary to have the classification and the aggregation (part-of) hierarchy of the machine in quest, but also very useful for retrieval and analysis of maintenance records to model additionally the hydraulic, electric, and functional connections and relationships in this machine. However, the aim of a very comfortable and expressive knowledge representation language is in conflict with the general requirement of having efficient reasoning mechanisms which can handle huge amounts of documents in the company's archives indexed with respect to complex, large domain ontologies. Hence, we would like to have a simple, efficient core language which could be extended if needed by additional representation primitives provided that there are efficient inferences which can be delivered together with these primitives.

Another problem using off-the-shelf ontology and KR languages for knowledge-based indexing in the OM has extensively be discussed by [Welty, 1998] who identified the semantical and representational issues when talking about subject taxonomies for content description of documents: a subject topic is usually embedded into a hierarchical structure (like a *class* in an object-oriented formalism) but is used as an attribute value (i.e., like an *instance* in an OO formalism). A related problem has been mentioned by a few authors, recently, but was also worked around, up to now (cp. [Schmiedel and Volle, 1996,van der Vet and Mars, 1996]): the higher-order aspects coming into play if one wants to give complex proposition as content descriptions (i.e., technically: assertions as attribute values).

Putting all pieces together we came to a preliminary solution as indicated in the example (Figure 4) and in the language description in Figure 5 (for more details, see [Abecker *et al.*, 1998a]): our main modeling approach is a conventional

[1] ESB = "Elektronisches Störungsbuch" is the German acronym for "Electronic Fault Recording".

object-centered formalism with only classification (subclass-superclass relationship) built-in as first-order means of representation, extended by two important features: (i) set-valued attributes, and (ii) annotated links. I.e., each attribute-value assertion can be associated with one or more annotation objects taken from annotation classes which can also be used to build up an annotation hierarchy. For document representation, content topics are represented as instances of a `concept` class which can be linked by annotated relations. Then, annotation objects can specify several relationships between concepts of the domain ontology, e.g., classification within the topic taxonomy, aggregation of topics, or other, application-specific relationships. In Figure 4, these annotated links are indicated by "object links". Since the evaluation of retrieval conditions formulated with respected to annotations can directly be compiled into efficient database operations of the underlying DBMS, this combines efficient processing and good expressiveness. Furthermore, for special annotation classes, the database procedures can be extended by efficient special-purpose inferences.

---

Ontologies are represented in an object-oriented manner. The declaration of a *class* looks like this:

$$\text{class}(attribute_1\text{: } type_1,\ldots,\ attribute_n\text{: } type_n)$$

If a class class1 *extends* a class class2, the following format is used:

$$\text{class1: class2}(attribute_{n+1}\text{: } type_{n+1},\ldots,\ attribute_{n+m}\text{: } type_{n+m})$$

*Attributes* are used to model semantic nets. In order to allow edges in these nets to be labeled with objects, a new complex type constructor is introduced, namely *annotations*. Annotated attributes are declared as follows:

attribute: class1/class2

or (for set-valued attributes)

attribute: {class1}/class2

In both cases, class2 is the annotation class.
An *object* of a class is represented as follows:

$$\text{class}(attribute_1{=}value_1/ann_1,\ldots) \text{ or}$$
$$\text{name: class}(attribute_1{=}value_1/ann_1,\ldots)$$

or (for set-valued attributes)

$$\text{class}(attribute_1{=}\{value_1/ann_1,\ldots\},\ldots)$$

---

**Fig. 5.** A short overview of the object-centered representation formalism.

### 3.3 Representing the example

Here, we show in some detail how the example introduced in Figure 4 can be implemented with the OCRA formalism.

*Information Ontology:* Any piece of information has a location (given as an URL) and a content, which is given as a set of content descriptions. Information sources may be documents, data, and rules, as well as references to personal and group competences. Any association of some piece of information with a content description may be annotated with a strength object, which for personal competences is a capability specification (Figure 6).

```
strength: ann()                 // ann is the annotation super class
                                // strength is an abstract super class
capability: strength(value: string) // specific subclass of strength

information(                    // information source
  name: string,
  url: string,
  content: {content} / strength // set-valued attribute for content
)                               // description, each content
                                // identifierannotated with a strength
                                // specification

personalCompetence: information(
  employee: employee            // defined in the enterprise ontology
)

groupCompetence: information(
  unit: organizationalUnit      // defined in the enterprise ontology
)
```

**Fig. 6.** A part of the sample information ontology.

*Enterprise Ontology:* Here, we consider the static organizational structure of a company is modeled, consisting of departments, employees, and their respective roles. In KnowMore, the enterprise ontology is used mainly for two reasons: the employees are actors in business processes, and they have competences and are therefore modeled as information sources in the information ontology (Figure 7).

```
company(
  name:string,
  address: string,
  departments: {department}
)

role: ann(name: string)

department(
  name: string,
  employees: {employee} / role
)
```

```
employee(
   person: person,
   phone: string,
   eMail: string
)

person(
   lastName: string,
   firstName: string,
   dateOfBirth: string
 )
```

**Fig. 7.** A part of the sample enterprise ontology.

*Content Descriptions:* Content descriptions as used in the information ontology may either be unstructured, like keywords or concepts of the domain ontology, or they may be complex, structured terms, like a sentence of the form subject-predicate-object (Figure 8).

*Domain Ontology:* The domain ontology is a complex semantic net, all primitive concepts are instances of class concept where the interrelationships are modeled

```
content()      // abstract super class for information content description

keyword: content(    // the most primitive content identifier: keywords
  name: string       // are just strings
)

concept: content(    // currently used for the KnowMore conceptual retrieval:
  name:string,       // formal concepts have a name and, maybe,
  ld:{ld},           // a number of textual descriptions or representations
  links: {concept}/conceptLink  // all semantic relationships between formal
)                               // concepts are represented by annotated links

conceptLink: ann()   // a conceptLink relates two formal concepts
isco: conceptLink()  // the 'is-subconcept-of' annotation class
                     // marks links which establish the is-a
                     // hierarchy within the domain ontology

ld(                  // a linguistic description is an evidence
  lang: string,      // (representation) of a formal concept in a text;
  name: string,      // several linguistic descriptions may describe occurrence
  abbr: string       // of the same formal concept in different languages
)

relation: concept(        // a complex content description could
  relationship: concept,  // contain a statement
  objects: {concept}
)
```

**Fig. 8.** Content descriptions as a part of the information ontology.

with the links attribute and some annotation objects like `sub` which is an annotation instance of the `isco` (= is-subconcept-of) annotation class which describes the classification hierarchy between concepts for information content description (Figure 9).

## 4 Heuristic Retrieval in Organizational Memories

If our knowledge sources are modeled as described above, retrieving information from the OM essentially amounts to a "select" operation on the object-oriented database, performed with appropriate search conditions formulated with respect to (i) the meta data given in the information ontology (e.g., which information sources to consider, or how old information to retrieve), (ii) specific context information (here, sophisticated similarity measures can be employed for comparison of actual query situation and context factors of knowledge sources described in the OM), and (iii) the content searched for. Since at the implementation level, all topic concepts are instances linked via annotated relationships, and the OCRA provides specialized operators for querying with respect to annotations, the retrieval can efficiently be supported by the underlying database system.

Concerning object links, search conditions can be specified (a) with respect to annotation (sub-)classes, (b) with respect to specific annotation objects (e.g., find all links marked with a "strength=strong" annotation), or (c) giving specific attributes with values (i.e., an "ordinary" database selection condition).

If we search for any information source about the concept "database", we can look for "database" in the domain ontology, and then pick out all information

```
...

sub: isco()     // an annotation object instantiates the standard
                // subconcept link annotation class

sw: concept(
  name="software",
  ld={ld(lang="English", name="software")},
  links={cs/sub}    // cs is a subconcept of computer science
)

db: concept(
  name="database",
  ld={ld(lang="English", name="database"),
      ld(lang="German", name=" Datenbank")
     },
  links={sw/sub}         // db is a subconcept of sw
)

...
```

**Fig. 9.** A part of the domain ontology used for information content descriptions.

sources which have a content characterization containing the "database" concept. The more interesting case is when there is not any information source about "database." Then we should use the links in the domain ontology. Consider, for instance, Figure 10. As we know "deductive database" and "relational database" are subconcepts of database. If no information about "database" is available, material about "deductive database" or "relational database" may also be suitable. Further, if nothing can be found in the subLink of "database", we have to search the upLink of "database", i.e., software. For undirected edges (e.g., some not further specified "has-to-do-with" relationship in a complex domain) we can select any precedent or post node in the domain ontology graph randomly for further retrieval.

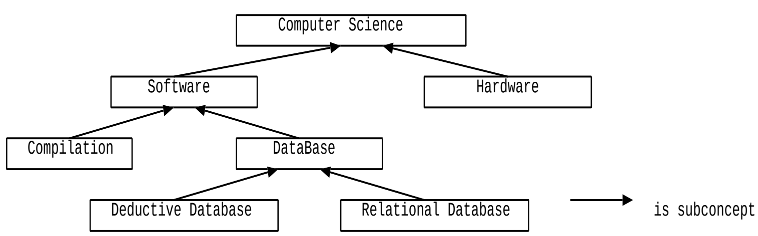


**Fig. 10.** A part of the domain ontology.

Besides such general search strategies, one can imagine that in each concrete application scenario there may be manifold sophisticated specific search heuristics. For giving to the user a means to specify such domain or application specific search heuristics, we introduced the notion of a *heuristics expression* which is a sequence of formulae of the following form:

$$f_1 \circ f_2 \circ \ldots \circ f_n$$

(denoting the functional composition of the $f_i$) with

$$f_i \equiv (\lambda)^\gamma$$

where $\lambda$ is a link or an inverse link (written as $link^{-1}$) and $\gamma$ is a "partial closure specification", i.e., one of the following path length specifications: $n$, $n..m$, $\geq n$, $*$ (as abbreviation for $\geq 0$), or $+$ (as abbreviation for $\geq 1$).

Such a formula takes as input a set of nodes of the directed graph under consideration and, for each node, follows the links specified in the formula in right-to-left order, in each step delivering an intermediary set of nodes as starting point for the next step. "Partial closure" means repeatedly following the same link type (in the case of $\gamma \equiv *$ generating the reflexive and transitive closure of the relation denoted by that link in the ontology). A heuristics formula makes sense if it delivers only information items as result. A sequence of formulae is evaluated in its sequential order with the semantics in mind that less trustworthy heuristics should be denoted last. Some sample retrieval heuristics can be expressed as follows:

1. $(content^{-1})^1$
   "First search for information sources directly linked to a search concept."
2. $(content^{-1})^1 \circ (isco^{-1})^+$
   "Then look for material about any subfield."
   For the sake of clarity, we have denoted two formulae here. An alternative formulation would have been: $(content^{-1})^1 \circ (isco^{-1})^*$
3. $(content^{-1})^1 \circ (isco)^1$
   "If result is empty after step 1 and 2, look for information concerning the direct superconcept of the topic in quest."

## 5 An Experimental System: CKBS

As a sub-module of the project KnowMore, we designed a competence query system called Competence Knowledge Base System (CKBS), which aims to provide a tool for querying the competence of the employees of an organization. In [Liao *et al.*, 1999] we elaborate in some more detail on the benefits of sophisticated domain models for the formulation of specific search heuristics.

The CKBS is designed as a client-server model, its architecture is shown in Fig. 11, the input and output interface are shown in Fig. 12.

In the input mask, the tool allows to formulate queries over competence fields, project memberships, or (directly) employee names. Complex queries can be composed using "AND", "OR", and "NOT".

The actual knowledge base with persons and their competence indices, as well as the ontological structure of competence fields, project membership, etc., are stored in a conventional relational database (RDB) which is coupled to the JAVA [JAVA, 1998] system code via JDBC [JDBC, 1998]. Details about how to efficiently store and access these object-oriented knowledge structures within the relational paradigm can be found in [Abecker *et al.*, 1998a]. The relational storage approach together with some additional schema information (denoted in the picture as DB-signature) allows to implement an object-centered relational algebra (OCRA, see [Abecker *et al.*, 1998a]) which provides an object-oriented view and access methods with special (weak) deductive capabilities for the underlying data. In detail, the OCRA directly implements the above introduced "partial closure" operator, an essential part of heuristics expressions, which allows to efficiently follow a predefined number of links between objects.

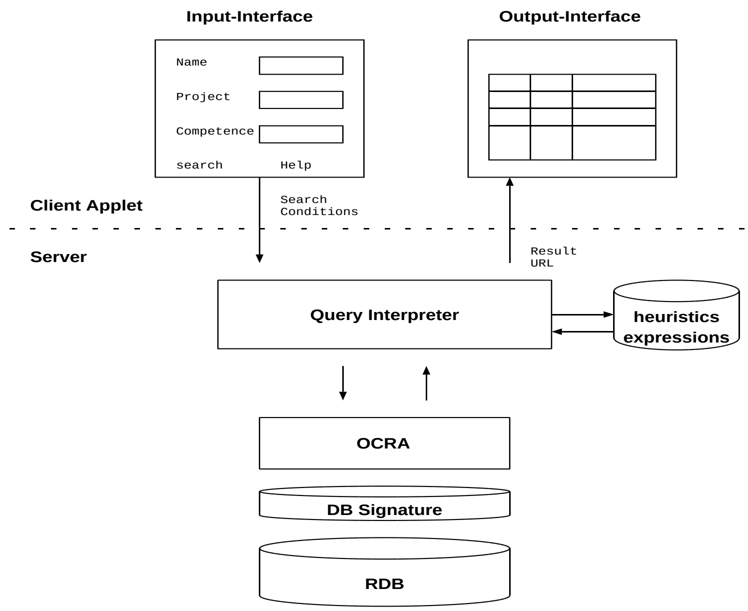


**Fig. 11.** The architecture of CKBS.

## 6 Conclusions

With this paper, we made some remarks to the discussion about ontologies for knowledge retrieval in the Organizational Memory. Although many people claim that ontologies would be a natural part of an OM, mostly they do not discuss which ontologies should serve for which purposes, how they are represented, and how they are built and maintained. We identified information source ontology, context, and application domain as the main ontological modeling dimensions to be considered when building practical "knowledge containers" in the sense of Richter. From the engineering point of view it makes sense to separately investigate all three ontological dimensions since they can be developed and reused separately. It is subject to further work to propose a reusable information ontology.

From the knowledge representation point of view, we identified some problems to be tackled, and presented aspects of our "intermediary solution", the object-centered relational algebra. Actually, the OCRA is a pragmatic trade-off between representational ease and efficiency, conceptually near to the Frame-Logic approach. But we will further work in this area. In contrast to most CBR approaches, we relied on a strongly logical basis for describing ontologies. This is not necessarily a contradiction to the CBR point of view, as Kamp showed.

However, it is still open and very interesting to further develop information retrieval inferences at the intersection of logical inference, theoretical foundations of utility and uncertainty processing (like the Dempster-Shafer-Approach, again, cp. [Richter, 1995]) and information theory [Barwise and Seligman, 1997]. In our current system, we employ logical inference only in a very rudimentary way, and merely make use of user-defined search heuristics as shown above. At first hand, this more imperative, navigation-oriented approach for heuristics specification seems to be promising for practical use.

If one wants to really put into use such systems as described here, knowledge acquisition and maintenance with respect to ontology construction as well as ontology use for document description get really crucial for success. Figure 13 shows a

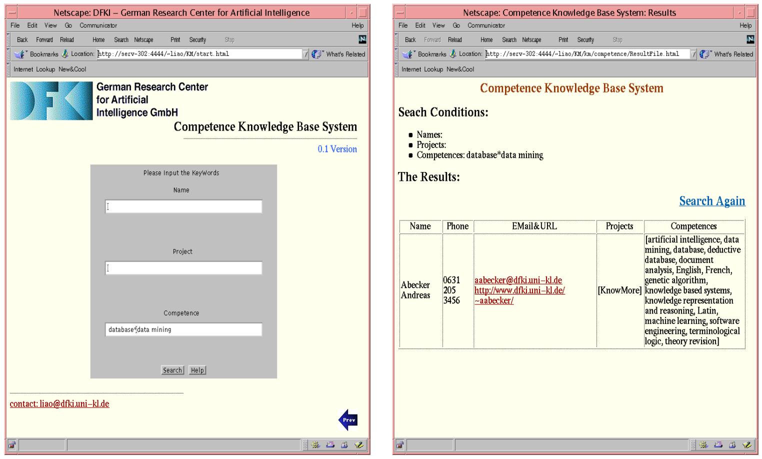


**Fig. 12.** The input and output interface of CKBS.

screenshot of our KnowMore ontology and knowledge description editor. Currently we investigate comfortable interfaces for end user and knowledge engineer, e.g., by incorporation of learning, automatic text categorization approaches.

## References


[Abecker *et al.*, 1998a] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Techniques for Organizational Memory Information Systems. DFKI Document D-98-02, DFKI GmbH, February 1998.

[Abecker *et al.*, 1998b] A. Abecker, A. Bernardi, K. Hinkelmann, O. Kühn, and M. Sintek. Toward a Technology for Organizational Memories. *IEEE Intelligent Systems and Their Applications*, 13(3), May/June 1998.

[Althoff *et al.*, 1998] K.-D. Althoff, F. Bomarius, and C. Tautz. Using case-based reasoning technology to build learning software organizations. In A. Abecker, St. Decker, N. Matta, F. Maurer, and U. Reimer, editors, *ECAI-98 Workshop on Building, Maintaining, and Using Organizational Memories (OM-98)*, August 1998. URL `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-14`.

[Barwise and Seligman, 1997] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. Tracts in Theoretical Computer Science. Cambridge University Press, 1997. To appear.

[Basili *et al.*, 1994] Victor R. Basili, Gianluigi Caldiera, and H. Dieter Rombach. *Experience Factory*. John Wiley & Sons, 1994.

[Benjamins *et al.*, 1998] V.R. Benjamins, D. Fensel, and A. Gómez Pérez. Knowledge management through ontologies. In U. Reimer, editor, *Proc. of the 2nd Int. Conference on Practical Aspects of Knowledge Management (PAKM98)*, October 1998. URL `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-13`.

[Bernardi *et al.*, 1998] Ansgar Bernardi, Knut Hinkelmann, and Michael Sintek. Model-based information systems for knowledge management. In *IT& Knows, Conference on Information Technology and Knowledge Systems, a part of the 15th IFIP World Computer Congress*, Vienna & Budapest, August 1998.

[Choo, 1995] C. W. Choo. Information Management for the Intelligent Organization: Roles and Implications for the Information professions. In *Proc. of the 1995 Digital Libraries Conference, Singapore*, March 1995.

[Fensel *et al.*, 1998] Dieter Fensel, Stefan Decker, Michael Erdmann, and Rudi Studer. Ontobroker: The very high idea. In *Proc. 11th Int. Florida AI Research Symposium (FLAIRS-98)*, May 1998.

[Gruber, 1993] T.R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2), 1993.

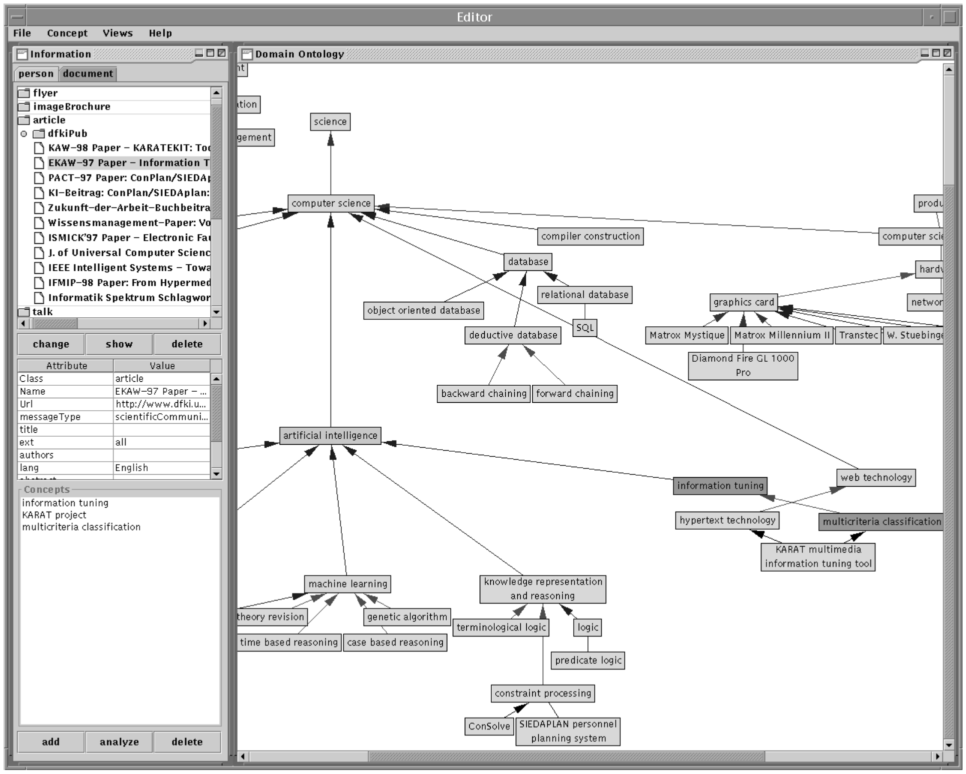


**Fig. 13.** The KnowMore ontology and document description editor


[Hartley *et al.*, 1997] R. Hartley, R. Kelsey, and R. Webster. Consolidating Multi-Source and Multi-Media Knowledge. In *AAAI Spring Symposium Artificial Intelligence in Knowledge Management*. AAAI, March 1997.

[JAVA, 1998] JAVA. JAVA Hompage at SUN. http://java.sun.com/, 1998.

[JDBC, 1998] JDBC. JDBC Homepage at SUN. http://java.sun.com/products/jdbc/, 1998.

[Kamp, 1996] G. Kamp. Using description logics for knowledge intensive case-based reasoning. Technical Report LKI-M-96/03, Labor für Künstliche Intelligenz, Universität Hamburg, August 1996.

[Kent and Neuss, 1996] R.E. Kent and Ch. Neuss. Web conceptual space. In *Proc. of WebNet'96, the World Conference of the Web Society*, October 1996.

[Kühn and Abecker, 1997] O. Kühn and A. Abecker. Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. *Journal of Universal Computer Science*, 3(8):929–954, 1997.

[Lenz, 1998] M. Lenz. Managing the knowledge contained in technical documents. In U. Reimer, editor, *Proc. of the 2nd Int. Conference on Practical Aspects of Knowledge Management (PAKM98)*, October 1998. URL `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-13`.

[Liao *et al.*, 1999] Minghong Liao, Knut Hinkelmann, Andreas Abecker, and Michael Sintek. A competence knowledge base system for the organizational memory. In Frank Puppe, editor, *XPS-99 / 5. Deutsche Tagung Wissensbasierte Systeme, Würzburg*. Springer Verlag, LNAI 1570, March 1999.

[Luke *et al.*, 1997] S. Luke, L. Spector, D. Rager, and J. Hendler. Ontology-based web agents. In *Proc. of First Int'l Conf. on Autonomous Agents, AA-97*, 1997.

[Richter, 1995] Michael M. Richter. The knowledge contained in similarity measures. Invited talk at the First Int. Conf. on CBR 1995 (ICCBR-95), 1995.

[Schmiedel and Volle, 1996] Albrecht Schmiedel and Philippe Volle. Using structured topics for managing generalized bookmarks. In *WWW5 Workshop on Artificial*

*Intelligence-based tools to help W3 users at the 5th Int'l WWW Conference, Paris, France*, 1996. URL http://www.info.unicaen.fr/~serge/3wia/workshop/.

[van der Vet and Mars, 1993] P.E. van der Vet and N.J.I. Mars. Structured system of concepts for storing, retrieving, and manipulating chemical information. *Journal of Chemical Information and Computer Sciences*, 33, 1993.

[van der Vet and Mars, 1996] Paul E. van der Vet and Nicolaas J.I. Mars. Coordination recovered. In *Informatiewetenschap 1996*. Delft, 1996.

[van der Vet *et al.*, 1995] P.E. van der Vet, P.H. Speel, and N.J.I. Mars. Ontologies for very large knowledge bases in materials science: a case study. In N.J.I. Mars, editor, *Towards very large knowledge bases: Knowledge Building and Knowledge Sharing*. IOS Press, Amsterdam and Ohmsha, Tokyo, 1995.

[Welty, 1998] Chris Welty. The ontological nature of subject taxonomies. 1998. 1998 Int'l Conference on Formal Ontology in Information Systems (FOIS'98).