

On the Relations between Structural Case-Based Reasoning and Ontology-based Knowledge Management

Ralph Bergmann & Martin Schaaf
University of Hildesheim
Data- and Knowledge Management Group
PO Box 101363
31113 Hildesheim, Germany

{bergmann | schAAF}@dwm.uni-hildesheim.de
www.dwm.uni-hildesheim.de

Abstract: This paper addresses the relations between ontology-based knowledge management implemented by logic-oriented knowledge representation/retrieval approaches and knowledge management using case-based reasoning. We argue that knowledge management with CBR does not only very much resemble but indeed is a kind of ontology-based knowledge management since it is based on closely related ideas and a similar development methodology, although the reasoning paradigms are different. Therefore, we conclude by proposing to merge logic-oriented and case-based retrieval and also to extend the current view of the semantic web architecture respectively.

1 Motivation

Structural Case-based Reasoning (SCBR) and ontology-based knowledge management (OBKM) are widely discussed as technologies for building organizational memory information systems (OMIS) to support knowledge management [A100; Be02; St02; Ab02]. When applying SCBR, the knowledge items (e.g., documents) are described by a characterization constructed from a previously developed domain vocabulary. The collection of all characterizations of the knowledge items constitutes the case base. In the traditional CBR view, the characterization can be considered as the problem description with the knowledge item itself (or a reference to it) as the solution. Queries to the OMIS are formulated in terms of the domain vocabulary and the similarity measure is used during retrieval to assess the utility [Be01] of knowledge items.

When applying OBKM, a domain ontology is constructed as a conceptual model for knowledge items described by metadata annotations. The domain ontology is represented using some logic formalism (e.g. F-Logic [Ki95]) that facilitates the specification of relevant domain relations axiomatically. The metadata annotations of the documents are considered as facts and build, together with the ontology, a knowledge base that is the foundation of the OMIS. A dedicated inference mechanism is used to answer queries conforming to the logic formalism and the terms defined in the ontology.

By comparing these two approaches, it becomes obvious that both are based on the same principle: knowledge items are abstracted to a characterization by metadata descriptions,

which are used for further processing. This characterization is based on some vocabulary/ontology that is a shared conceptualization of the domain among the computer agents and users of the OMIS. Despite of these similarities, there is currently not much cross-citation in papers addressing the one or the other approach. Although some of the relations between both approaches might be implicitly clear, they have never been analyzed systematically and explicitly stated before. With this paper we want to unveil those relationships and break the borders between both approaches by claiming that KM by SCBR is a kind of OBKM. The difference lies mainly in the inference mechanism used: logic vs. utility-based reasoning. With this paper, we also want to contribute to a better understanding within the (recently) merged German SIGs on KM and CBR. This might be a starting point for further exploring new synergies.

2 Ontology-based Knowledge Management

The notion of Ontology-based Knowledge Management (OBKM) refers to activities concerning the creation, accumulation, sharing, reuse and further development of knowledge in an organization within the context of explicitly defined conceptual models. The term ontology stands for the representation of a conceptual model and is the core of OBKM. Its philosophical origin goes back to Aristotle who is supposed to be the founder of metaphysics as a separate discipline. According to [BS91], *metaphysic* and *ontology* coincide partially and can be regarded from two different points of view: a) with respect to its object, e.g. thing, Ding, being etc., b) in relation to other philosophical and non-philosophical disciplines. Within this paper we will emphasize only the technical aspects of OBKM and from this perspective we consider an ontology as formal description of the entities, relationships, and constraints that make the conceptual model. Depending on the expressiveness and the degree of formality of the underlying representation language, an ontology can range from a simple taxonomic hierarchy of classes to a logic program utilizing first-order predicate logic, modal logic, or even higher order logics with probabilities. In contrast to classical expert systems, ontology-based systems typically distinguish between multiple levels of knowledge from common sense knowledge to highly specific domain knowledge.

2.1 Ontological Engineering

As a relatively new sub-discipline of knowledge engineering, ontological engineering focuses on the systematic development of ontologies in a reusable and modular fashion and their maintenance. Ontological engineering has probably its origins in the CYC project [LG90], which first addressed the issue of reusability and modularity of large knowledge bases, and the development of the knowledge representation language KL-ONE [BS85], which was the first logical formalization of a frame-based semantic network. Other approaches for developing knowledge-based systems include contexts respective microtheories, compositional modelling, or knowledge composition and merging ([Gu91], [FF91], [CP97], [NM00]).

KL-ONE inspired an entire new discipline in logical frame-based languages called terminological logics or description logics. It distinguishes between a T-Box, which is a sub-

sumption hierarchy called the axioms or ontology of the knowledge base, and the A-Box that comprises the instance level knowledge (facts etc.). The T-Box is somewhat similar to a schema in relational database theory, while the A-Box particularly corresponds to tuples of a database.

Although engineering principles for ontologies emphasize modularity and reusability, this is still very difficult to achieve for systems beyond research prototypes. It requires formal and declarative representation languages that have a standardized syntax, a well founded semantic, and the sufficient expressiveness for real world applications. Consequently, the most important advances in ontological engineering currently come from the research and standardization efforts for representation languages and models for the semantic web, which are developed on top of XML. A variety of languages compete to be the language of choice like the XML Ontology Exchange Language (XOL) [Ka99], the Ontology Inference Layer (OIL) [Fe00], the DARPA Agent Markup Language (DAML) [DA02], the Resource Description Framework (RDF) [LS99] and the corresponding RDF Schema Specification [Ha02], or XML Topic Maps (XMT) [PM01]. In the following we will briefly show two approaches, RDF(S) and DAML/OIL, that already have reached a certain level of maturity.

2.2 RDF(S)

The Resource Description Framework (RDF) [LS99] is a W3C recommendation for encoding, exchange, and reuse of structured metadata and uses XML as underlying language. The RDF Data Model is based on resources and properties. A resource is everything that can be uniquely identified by a Uniform Resource Identifier (URI). A property can associate resources with values and can be labeled by a name illustrating the relationship. RDF defines a set of atomic types for property values like strings or integers. Furthermore, a property value may be another property enabling the specification of directly labeled graphs, which can be interpreted as a semantic network, or a collection of values. RDF is an easy to use formalism that resembles very much an entity relationship diagram. Meanwhile, it has become the foundation of higher-level standardizations and many ontology-based systems allow using RDF for metadata (A-Box) but keep a proprietary formalism for the ontology itself. An approach to close this gap led to the development of RDF Schema [BG02] that denotes some special associations, for instance a “subClassOf” relation, and thereby provides mechanisms to define classes of resources, to restrict possible combinations of classes and relationships, and detect violations of those restrictions.

2.3 DAML+OIL

Although current efforts of the W3C aim to supply a model-theoretic semantic for RDF and RDF Schema [Ha02b] in order to enable a unique interpretation for automatic reasoning, RDF(S) still lacks the necessary expressive power for many applications. Hence, two extensions to RDF(S), namely the DARPA Agent Markup Language (DAML) [DA02] and the Ontology Inference Layer (OIL) [Fe00], have been proposed and finally merged into DAML+OIL [Ha02a] because of their similarity. DAML+OIL is based on description logics encoded in RDF. In addition to RDF(S), DAML provides the ability to express the equivalence or disjointness of classes, additional restrictions like cardinality, or to build

new classes as intersections or complements of other classes. Furthermore, DAML has been integrated with XML Schema providing a rich set of data types, which are still missing in RDF(S). The further evolution of DAML+OIL is the Ontology Web Language (OWL) [De02] for what a first language reference as a W3C working draft is available, now.

2.4 On the Usage of Ontologies in OBKM

Gruber [Gr93] defines an ontology as “an explicit specification of a conceptualization” committed by a set of agents “so that they can communicate about a domain of discourse”. This definition proposes ontologies as a formal representation of background knowledge in a multi-agent environment enabling, for instance, distributed reasoning across multiple knowledge bases. By assuming any problem or task specific knowledge being implemented by the agents, it implies also an important design principle for ontology-based systems with respect to modularity and reusability.

A more focused use for ontologies, especially for OBKM purposes, is the systematic creation and storage of knowledge assets based on the characterization of knowledge items [Fe98]. Here, ontology and characterization are the key for content-based access (filter, retrieve, render, etc.) to knowledge items [Gu99]. Furthermore, the ontology itself can serve as a communication base about the products and processes e.g. for generating explanations to users. In the following we will illustrate these very basic ideas by a simple example. Table 1 shows an excerpt of an F-Logic¹ [Ki95] specification how it might be used in the FLORID [FL02] system as ontology for a document management approach maintaining publications submitted to a particular event (e.g. a workshop).

<pre>publication[title => string; authors =>> author; location => URI; submittedTo => event; topics =>> topic;].</pre>	<pre>workshop : event. event[title => string; cfp => URI; appDay@(string) => date; topics =>> topic;].</pre>
<pre>knowledgeManagement :: topic. experienceManagement :: knowledgeManagement. knowledgeRepresentation :: knowledgeManagement.</pre>	
$\forall_{X,T,E} X : publication[topics- \gg T]$ $\leftarrow X[submittedTo- \rightarrow E] \wedge E[topics- \gg T] \quad (*)$	

Table 1: Excerpt of an Ontology in F-Logic

The specification contains only F-Logic signatures that correspond to the T-Box in KL-ONE. They indicate, for example, that each publication has authors, a location given by a Uniform Resource Identifier (URI), and the link to the particular event. An event, e.g. a workshop, has some dates associated (e.g. submission deadline) as well as a set of topics.

¹ F-Logic is a descendant of KL-ONE, which does not distinguish between A-Box and T-Box knowledge.

Furthermore, rule (*) formalizes the simple relationship between the associated topics of the document and the topics of the event.

The necessary link to a knowledge item, e.g. a PDF-file containing a publication or a web site announcing a workshop, is now established by a meta-data characterization that somehow conforms to the ontology in Table 1. Such a characterization does not require an expressive underlying formalism but must allow identifying the concepts and associated entities of the ontology in a unique and unambiguous way. For instance Figure 1 illustrates the usage of RDF for characterization of a publication. The properties of the RDF description are labeled as the slots of the corresponding F-Logic frame and assign resources as values.

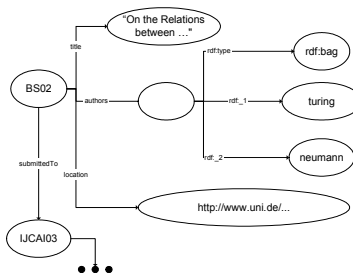


Figure 1: RDF Example Characterization

From a theoretical point of view, it does not matter if such a characterization is part of the knowledge itself or provided separately. Furthermore, depending on the ontology-based system, conformance requirements are more or less strict. Typically, even a simple and weakly typed entity-relationship representation like RDF is sufficient. Some ontology representation languages like DAML/OIL extend RDF by own constructs that allow interpreting the ontology itself as a kind of schema for the characterization. In addition to traditional database schemes, ontologies provide an axiomatic base of the stored knowledge items making it possible to implement retrieval, filtering etc. by derivation. For instance, retrieval of knowledge items is, beside communication and cooperation, the most frequent application of ontology-based systems in OBKM and can be facilitated by queries like

$$\forall x \leftarrow X : publication[topics \rightarrow \{experienceManagement\}, authors \rightarrow \{turing\}]$$

that selects all documents with (co-) author *turing* related to the topic *experienceManagement*. An appropriate inference system can answer the query by proving if it is a consequence from the ontology and the set of characterizations (considered as facts) and thereby finding characterizations as substitutions for the variable *X* that represent the requested documents. Finally, we would like to mention, that using the ontology as an axiomatic base for a logic calculus, derivation is, of course, restricted to the deductive closure of the axioms.

3 Structural CBR for KM

In CBR there are three main approaches that differ in the sources, materials, and knowledge they use [Be99]. The textual CBR approach is similar to traditional information retrieval in that it works directly on the text documents. There is no a-priori domain model, but similarity measures can be introduced between the words occurring in the documents. Therefore, retrieval is very similar to keyword matching, but considers the similarity for document scoring.

Conversational CBR captures the knowledge contained in customer/agent conversations. A case is represented through a list of questions that varies from one case to the other. There is no domain model and no standardized structure for all the cases. This approach is very useful for domains where a high volume of simple problems must be solved again and again.

The *structural CBR* approach is the third approach and relies on cases that are described with attributes and values that are pre-defined. In different SCBR systems, attributes may be organized as flat tables, or as sets of tables with relations, or they may be structured in an object-oriented manner. The SCBR approach is useful in domains where additional knowledge, beside cases, must be used in order to produce good results. In the following we focus on the SCBR approach.

3.1 Knowledge Containers

In the SCBR approach, knowledge is represented in one of the four knowledge containers: the vocabulary used, the similarity measure, the solution transformation, and the case-base [Ri95]. In principle, each container is able to carry all the available knowledge, but this does not mean that this is advisable. The first three containers include compiled knowledge (with “compile time“ we mean the development time before actual problem solving, and “compilation“ is taken in a very general sense including human coding activities), while the case-base consists of case-specific knowledge that is interpreted at run time, i.e. during actual problem solving. For compiled knowledge the maintenance task is as difficult as for knowledge-based systems in general. However, for interpreted knowledge, the maintenance task is easier because it results in updating the case-base only. In our opinion, a main attractiveness of CBR comes from the flexibility to decide pragmatically which container includes which knowledge.

When applying SCBR to knowledge management, the characterization of the knowledge items is stored in the case base. Ideally, the vocabulary used to represent the cases is developed a-priori for the domain at hand and is considered as stable.

3.2 Vocabulary Representation in CBR

State-of-the-art CBR systems make use of an object-oriented vocabulary representation [Ma94; AP95]. Object-oriented case representations can be seen as an extension of the attribute-value representation. They make use of the data modeling approach of the object-oriented paradigm including is-a and other arbitrary binary relations as well as the inheritance principle. Such representations are particularly suitable for complex domains in which cases with different structures occur.

The structure of an object is described by an object class that defines the set of attributes together with a type (set of possible values or sub-objects) for each attribute. Object classes are arranged in a class hierarchy, that is usually a n-ary tree in which sub-classes inherit attributes as well as their definition from the parent class. Moreover, we distinguish between *simple attributes*, which have a simple type like Integer or Symbol, and so-called *relational attributes*. Relational attributes hold complete objects of some (arbitrary) class from the class hierarchy. They represent a directed binary relation, e.g., a part-of relation, between the object that defines the relational attribute and the object to which it refers. Relational attributes are used to represent complex case structures. The ability to relate an object to another object of an arbitrary class (or an arbitrary sub-class from a specified parent class) enables the representation of cases with different structures in an appropriate way. Different representation languages for the vocabulary have been developed such as CASUEL [Ma94] and the XML-based Orange Model Markup language [ST02] used in the commercial CBR tool orange from empolis.

When applying CBR for KM, the development of the vocabulary is a crucial issue. The vocabulary is used for characterizing the knowledge items to be searched. When developing a vocabulary, the following must be considered:

Utility Distinguishability: The vocabulary must be complete in the following sense: it must be possible to decide based on the selected classes and attribute values whether it is possible to make use of the knowledge item in a new situation. If it is not possible to distinguish two knowledge items that must be distinguished based on the attributes in the characterization, new attributes or classes must be added to enable the differentiation between the two. This criterion has been formalized in [Be02].

Common Understanding: There must be a common understanding of the use of the vocabulary items (and the entire representation language) among the persons or agents in charge of characterizing knowledge items and the users formulating a query with these items. That is, all people involved should characterize a knowledge item the same way and should characterize their queries the same way. In many KM project that involve CBR technology, it has been recognized that the development of such a shared vocabulary is a very difficult task explicitly addressed in development methodologies for CBR applications, such as the INRECA methodology [Be99].

Besides these criteria, one usually aims at achieving a vocabulary in which the attributes are independent from each other (i.e., there is no functional dependency) and the set of attributes is minimal (i.e., there is no redundant attribute). Although these criteria help in the engineering of appropriate similarity measures, they are not mandatory and are often ignored if there is not one single clearly defined task to be supported with the OMIS.

3.3 Characterizations

For applying CBR for KM, the cases to be stored in the cases base usually consist of a characterization part and a lesson part. The characterization part is represented using the vocabulary and consists of a collection of objects instances from the classes of the vocabulary. The lesson part just consists of a link to the knowledge item that is characterized.

For a given set of knowledge items, these characterizations must be constructed either manually, i.e. the documents must be annotated with their characterization, or by applying

text-mining approaches. In the latter case, syntactic text analysis rules can be applied to map certain text patterns to attribute values or object instances of the characterization.

3.4 Similarity, Utility, and Rules

The similarity measures used in CBR are of critical importance during the retrieval of knowledge items for a new query. Unlike in early CBR approaches, the recent view is that similarity is usually not just an arbitrary distance measure, but a function that approximately measures utility. The similarity measure assesses the utility of a knowledge item only based on the characterization. The knowledge container view made clear that the similarity measure itself contains (compiled) knowledge. This is knowledge about the utility of a knowledge item re-applied in a new context [Be01]. Connected with this observation was the need to model similarity knowledge explicitly for an application domain, as it is done with other kinds of knowledge too. Current similarity modeling approaches are tightly integrated with object-oriented vocabulary representations [Be02]. The similarity measure allows the retrieval of knowledge items that do not exactly match the query, but which can differ in many ways. The similarity value imposes a partial ordering on the knowledge items according to their relevance for the current query, which is an important feedback to the user of the OMIS.

Beside the use of similarity measures, CBR research also came up with approaches for integrating rule-based background knowledge [Aa91, Be96, Be02]. For example with completion rules, it is possible to derive deductive, logical conclusions from the characterization of knowledge items. These conclusions are stored as part of the characterization for each knowledge item, i.e., for each knowledge item, the deductive closure (which must of course be guaranteed to be finite) is determined and stored in as part of the case in the case base. During retrieval the deductive close is also computed for the query and the similarity is determined between the extended representations. The most common use of this approach is for determining derived (or also called virtual) attributes that are computed from the given representation for the means of similarity assessment.

4 Relations between Structural CBR and OBKM

From the previous analysis of knowledge management by SCBR and OBKM it should have become clear that both rely on metadata annotations that serve the purpose of characterizing instead of formalizing knowledge items. In CBR these characterizations are called cases and, basically, it does not matter where the representation of the characterization is physically located. It may be stored together with the knowledge item itself (e.g. by using a structured XML-based format) or, as with CBR, in a case base. A more important relationship is given by the SCBR vocabulary that very much resembles the ontology in OBKM. Both are formal models for restricting the possible interpretations of metadata annotations thereby providing the necessary background knowledge for semantic-based access to knowledge items. It is obvious that the fundamental types of knowledge of SCBR and OBKM are strongly related as shown in Figure 2. Hence, from these relationships follows that design principles for SCBR and OBKM are closely related, too. Several CBR development and maintenance approaches have been researched, for instance the

INRECA methodology [Be99, Ta01], and they are at least partially structured according to the CBR knowledge containers and do address the vocabulary development as well. For OBKM, [St02] follows a KADS oriented methodology and presents a meta-process for systematic ontology development that utilizes SE-tools.

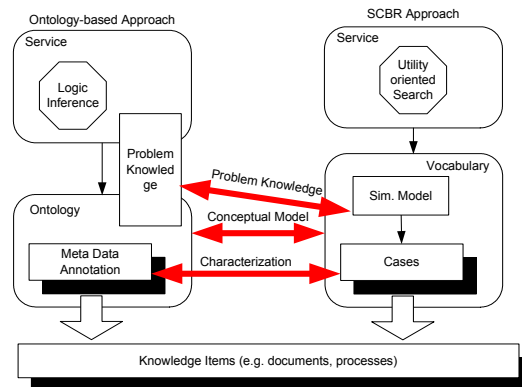


Figure 2: Ontology vs. SCBR Knowledge Containers

An important difference between both approaches results from the fact that SCBR-systems are often isolated and closed in the sense that they are not developed with respect to cooperation with other systems. For that reason, though research of vocabulary representation languages led to expressive languages [Ma94; AP95], standardization was not a big issue in past CBR research. Most SCBR-based systems rely on proprietary, sometimes even XML compliant, languages for the vocabulary and the cases but do not allow an easy knowledge exchange. However, current research for distributed CBR [LS02] shows how CBR can benefit from systems that are able the search across multiple-case bases. Of course, this is only possible if a standardized, shared knowledge representation language enables unambiguous interpretation of cases stored in the different case bases.

The coincidence of an SCBR vocabulary and an ontology becomes even more prevalent if we compare vocabulary representation approaches to ontology representation languages mentioned earlier in this paper. They provide nearly the same expressiveness by utilizing object-oriented technology allowing the specification of concept hierarchies, arbitrary binary relations, types, and rules e.g. like definite clauses in horn logic. Neglecting the fact that an ontology typically serves many purposes one can say that a SCBR vocabulary is an ontology of the domain of discourse underlying the SCBR application.

The major difference between the SCBR and OBKM approach results from different reasoning strategies. As mentioned before, most ontology-based systems utilize logic-based inference, while SCBR systems provide a search functionality that makes use of similarity measures for ranking results according to their utility with respect to a given query. In our opinion, both reasoning strategies complement each other very well. On the one hand, logic deduction produces only correct and provable results, which are consequences of the ontology and metadata. Computer agents normally require this for further processing. On the other hand, SCBR retrieval suggests results even in the case that no exactly matching answers can be found. This has been proven as highly efficient in many real-world applications [Be99]. For realizing the utility-oriented search, SCBR systems introduce an addi-

tional kind of knowledge that is the similarity model. Although the similarity model is part of the problem knowledge, it is a first-class citizen of each CBR system in the sense that constructs required for specification are usually part of the vocabulary representation language. This emphasizes the more problem-oriented approach of SCBR.

5 Conclusions

Within this paper we analyzed knowledge management by SCBR and ontology-based knowledge management and showed a strong relationship between both approaches with respect to technological but also to methodological issues. However, we identified several differences, too, being a potential source for synergies. For example, OBKM comes up with a variety of standardized knowledge representation languages. Their incorporation into SCBR-based systems would enable to apply CBR technology to a broader range of applications. As a consequence, this makes it possible to develop unified modeling tools for greater flexibility. The decision between the different reasoning strategies supported by SCBR and OBKM may be postponed to a later phase of the development. Conversely, ontology engineering could take advantage of experiences with real-world SCBR applications that are discussed, for example, in [Be99]. Finally, by having a closer look at the current state of the semantic web, it becomes obvious that, even under the assumption of standardized knowledge representation languages, ontologies are often highly specific to their domain of discourse. Hence, interoperability can only be achieved by some kind of semantic unification. For that purpose, a strict, logic-oriented approach does not seem to be the ultimate solution, especially when only an approximation of unification is possible. SCBR, beside arbitrary probabilistic approaches, seems to be a good starting point for further research because of its strong relationship to OBKM. It introduces the similarity model as another type of knowledge that recommends itself to become part of future extensions to knowledge representation standards.

References

- [Aa91] Aamodt, A. (1991). A knowledge intensive integrated approach to problem solving and sustained learning. PhD Thesis, University of Trondheim.
- [Ab02] Abecker, A., Hinkelmann, K., Maus, H., Müller, H.J., (Eds.) (2002): *Geschäftsprozess-orientiertes Wissensmanagement*. Springer Verlag
- [Al00] Althoff, K.D., Bomarius, F., Tautz, C. (2000). *Using a case-based reasoning strategy to build learning software organizations*. IEEE Journal on Intelligent Systems.
- [AP95] Arcos J., Plaza E. (1995). *Reflection in NOOS: An object-oriented representation language for knowledge modelling*. In IJCAI-95 Workshop on reflection and meta-level architecture and their applications in AI.
- [Be01] Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I. (2001). *Utility-Oriented Matching: A New Research Direction for Case-Based Reasoning*. In: Vollrath, Schmitt, & Reimer: 9th German Workshop on Case-Based Reasoning, GWCBR'01. In Schnurr, Staab, Studer, Stumme, Sure (Eds.): *Professionelles Wissensmanagement*, Shaker.
- [Be02] Bergmann, R. (2002). Experience Management: Foundations, Development Methodology, and Internet-Based Applications. LNAI 2432, Springer.
- [Be96] Bergmann, R., Wilke, W., Vollrath, I. & Wess, S. (1996) Integrating general knowledge with object-oriented case representation and reasoning. In: H.-D. Burkhard & M. Lenz (Hrsg.) *4th German Workshop: Case-Based Reasoning - System Development and Evaluation*, Informatik-Berichte Nr. 55, Humboldt-Universität Berlin, 120-127.
- [Be99] Bergmann, R., Breen, S., Göker, M., Manago, M., Wess, S. (1999). *Developing industrial case-based reasoning applications*. LNAI 1612, Springer.
- [BG02] Brickley D., Guha R. V. (2002): RDF Vocabulary Description Language 1.0: RDF Schema: <http://www.w3.org/TR/rdf-schema/>,
- [BS85] Brachmann R., Schmolze J. G. (1985): An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171-216
- [BS91] Burkhardt H., Smith B. (1991): *Handbook of Metaphysics and Ontology*. Munich: Philosophia.
- [CP97] Clark P., Porter B. (1997): *Building Concept Representations from Reusable Components*. In Proceedings of AAAI '97. Menlo Park, CA: AAAI Press, pp. 369-376.
- [DA02] The DARPA Agent Markup Language Homepage (2002): <http://www.daml.org/>, last visited: 10/01/2002
- [De02] Dean M., Connolly D., Harmelen F. van, Hendler J., Horrocks I., McGuinness D., Patel-Schneider P. F., Stein L. A. (2002): OWL Web Ontology Language 1.0 Reference, W3C Working Draft 29 July 2002. <http://www.w3.org/TR/owl-ref/>, last visited: 10/01/2002
- [Fe00] Fensel D. et al. (2002): OIL in a nutshell In: Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), Lecture Notes in Artificial Intelligence, LNAI, Springer-Verlag.
- [Fe98] Fensel D., Decker S., Erdmann M.I, Studer R. (1998): Ontobroker: The Very High Idea. In: Proceedings of the 11th International Flairs Conference (FLAIRS-98), Sanibal Island, Florida, May 1998.
- [FF91] Falkenheiner B., Forbus K. (1991): *Compositional Modeling: Finding the Right Model for the Job*. *Artificial Intelligence* 51:95-143
- [FL02] The FLORID/FloXML Project (2002): <http://www.informatik.uni-freiburg.de/~dbis/florid/>, last visited: 10/01/2002

- [Gr93] Gruber T. R. (1993): *A translation approach to portable ontologies*. Knowledge Acquisition, 5(2):199-220.
- [Gu91] Guha, R. V. (1991): *Contexts: A Formalization and Some Applications*. Ph.D. Thesis, Stanford University.
- [Gu99] Guarino N., Masolo C., Vetere G. (1999): *OntoSeek: Content-Based Access to the Web*. IEEE Intelligent Systems 14(3): 70 – 80.
<http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/OntoSeek.pdf>
- [Ha02a] Frank van Harmelen, Peter F. Patel-Schneider, and Ian Horrocks. Reference description of the DAML+OIL ontology markup language.
<http://www.daml.org/2001/03/reference.html>, 2002.
- [Ha02b] Hayes P. (2002): *RDF Model Theory, W3C Working Draft 29 April 2002*.
<http://www.w3.org/TR/rdf-mt/>,
- [Ka99] Karp P. D., Vinay C. K., Thomere J. (1999): *XOL: An XML-Based Ontology Exchange Language*. Pangaea Systems and SRI, International.
<http://www.ai.sri.com/pkarp/xol/>, last visited: 10/03/2002.
- [Ki95] Kifer M., Lausen G., Wu J. (1995): *Logical Foundations of Object Oriented and Frame Based Languages*. Journal of ACM 1995, vol. 42, p. 741-843.
- [LG90] Lenat, D., Guha R. (1990): *Building Large Knowledge-Based Systems*. Reading, MA: Addison-Wesley.
- [LS02] Leake D. B., Sooriamurthi R. (2002): *Automatically Selecting Strategies for Multi-Case-Base Reasoning*. In Proceedings ECCBR 2002: 204-233
- [LS99] Lassila O, Swick R. (1999): *Resource Description Framework (RDF) Model and Syntax Specification*. <http://www.w3.org/TR/REC-rdf-syntax/>, last visited: 10/03/2002.
- [Ma94] Manago, M., Bergmann, R., Wess, S., Traphöner, R. (1994). *CASUEL: A common case representation language*. ESPRIT Project INRECA. Deliverable D1, University of Kaiserslautern.
- [NM00] Noy N. F., Musen M. A. (2000): *PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment*. Technical Report SMI-2000-0831. Stanford Medical Informatics.
- [PM01] Pepper S., Moore G. (Eds.) (2001): *XML Topic Maps (XTM) 1.0 - TopicMaps.Org Specification*. <http://www.topicmaps.org/xtm/index.html>, last visited: 10/03/2002.
- [Ri95] Richter, M. M. (1995). *The Knowledge Contained in Similarity Measures*. Invited talk at the First International Conference on CBR (ICCB-95).
- [St02] Staab, S. (2002). *Wissensmanagement mit Ontologien und Metadaten*. Informatik Spektrum.
- [Ta01] Tautz, C. (2001): *Customizing Software Engineering Experience Management Systems to Organizational Needs*. PhD Thesis, Department of Computer Science, University of Kaiserslautern, Fraunhofer IRB, Stuttgart, Germany, 2001.
- [ST02] Schumacher J., Traphöner R. (2002). Knowledge Modelling. Technical Report, WESELL Project, Deliverable, 2000.