Journal of Universal Computer Science, vol. 9, no. 7 (2003), 608-626
submitted: 3/2/03, accepted: 21/7/03, appeared: 28/7/03 © *J.UCS*

Structural Case-Based Reasoning and Ontology-Based Knowledge Management: A Perfect Match?

Ralph Bergmann
(University of Hildesheim, Germany
bergmann@dwm.uni-hildesheim.de)

Martin Schaaf
(University of Hildesheim, Germany
schaaf@dwm.uni-hildesheim.de)

Abstract: This article 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.

Keywords: Knowledge Management, Case-based Reasoning, Ontology, XML, Semantic Web
Categories: H.3.1, H.3.3, I.2.1, I.2.4

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 [Althoff 00], [Bergmann 02], [Staab 02], [Abecker 02]. 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 [Bergmann 01] 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 [Kifer 95]) 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 conceptualisation 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 analysed systematically and explicitly stated before. With this article 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.

2 Structural CBR for KM

The basic idea of CBR is to solve new problems by comparing them to problems already solved [Aamodt, Plaza 94], [Leake 96], [Bergmann 99]. The key assumption is that if two problems are similar, then their solutions are probably also similar. This approach can be successfully applied for building OMIS that retrieve knowledge items based on a particular notion of similarity. In CBR there are three main approaches that differ in the sources, materials, and knowledge they use [Bergmann 99].

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.

2.1 Knowledge Containers

In the SCBR approach, knowledge is distributed among the four knowledge containers (see Figure 1): the *vocabulary* used, the *similarity measure*, the *solution transformation*, and the *case-base* [Richter 95]. 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. In our opinion, a main attractiveness of CBR comes from the flexibility to decide pragmatically which container includes which knowledge.

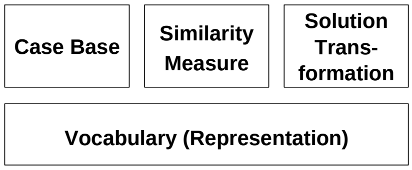

Figure 1: CBR Knowledge Containers

When applying SCBR to knowledge management, the characterizations of the knowledge items are stored as cases in the case base (see Figure 2). Each characterization contains a link to the knowledge item itself. Ideally, the vocabulary used to represent the cases is developed a-priori for the domain at hand and is considered as stable. The vocabulary shall contain the relevant concepts of the domain that occur in the knowledge items.

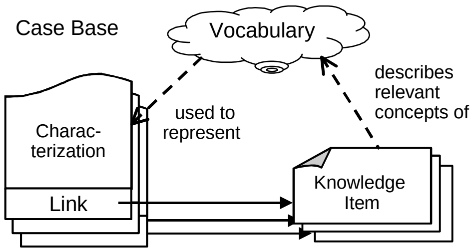

Figure 2: CBR Knowledge Containers

2.2 Vocabulary Representation in CBR

State-of-the-art CBR systems make use of an object-oriented vocabulary representation [Manago 94], [Arcos, Plaza 95]. 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 an 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. Several representation languages for the vocabulary have been developed such as CASUEL [Manago 94] and the XML-based Orenge Model Markup language [Schumacher, Traphöner 00] used in the commercial CBR tool orenge from empolis.

For KM applications based on structural CBR, the development of the vocabulary is a crucial issue and 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 [Bergmann 02].

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 to the OMIS. That is, all people involved should characterize a knowledge item the same way and should characterize their queries the same way. In many KM projects 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 [Bergmann 99].

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.

Figure 3 gives an example of a fragment of a vocabulary, modelled in an object-oriented manner, which could be used in a OMIS for managing a company's experience about resolving problems with certain computer hardware. This figure shows a class hierarchy with 13 classes, some of which have simple and/or relational attributes. The PC class, for example, has three relational attributes (printed in *italics*) that hold objects to represent the *main board*, the *hard disk* and the *optional storage* as well as a simple attribute that describes the *case* (e.g. ATX case type for a PC). The example shows some more detail in the modelling of different kinds of storage devices. Please note that in this model simple as well as relational attributes are inherited to all subclasses. Not shown, but also part of the vocabulary, are the definitions of different Symbol types, each of which enumerates the range of possible values for each symbolic attribute.

2.3 Characterizations

When applying CBR for KM, the cases to be stored in the case 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.

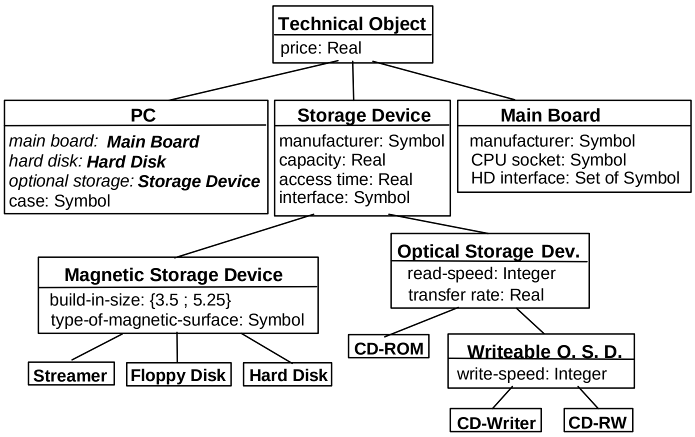

Figure 3: Example Object-Oriented Vocabulary

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 techniques. 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.

The example given in Figure 4 shows four characterizations that could have been derived from four exemplary knowledge items, each of which describes a certain faulty behaviour of some hardware component. Please note that due to the limitations of this example we omit the vocabulary fragment for modelling the failure type itself. The first three knowledge items C1-C3 shown, describe a failure with a certain hardware component (a Hard Disk, a CD-ROM a CD-RW). Knowledge item C4, however, describes a general failure that could occur with all optical IDE storage devices with a read speed of 56x, such as problems caused by the high rotation speed of the CD. Suppose that each of these characterizations include a link to a particular document (e.g. a Web document in a company's intranet) that describes the failure and possible remedies in detail.

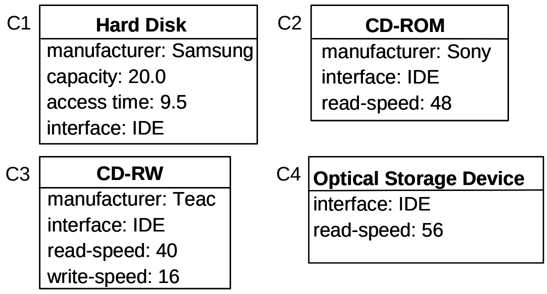

Figure 4: Example Characterizations

2.4 Similarity and Utility

The similarity measures used in CBR are of critical importance during the retrieval of knowledge items for a given query. Today it is common to measure the similarity by a real value within the interval [0..1]. In contrast to early CBR approaches, similarity is no longer considered as an arbitrary distance measure, but a function that approximately measures utility. More precisely, 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 [Bergmann 01]. 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 [Bergmann 02]. Similarity measures are often defined by the following general scheme (see Figure 5). The goal is to determine the similarity between two objects, i.e., one object representing the characterization (or a part of it) and one object representing the query. We call this *object similarity*. It is determined recursively in a bottom up fashion, i.e., for each simple attribute, a local similarity measure determines the similarity between the two attribute values, and for each relational attribute an object similarity measure recursively compares the two related sub-objects. Then, the similarity values from the local similarity measures and the object similarity measures, respectively, are aggregated by an aggregation function to the object similarity between the objects being compared.

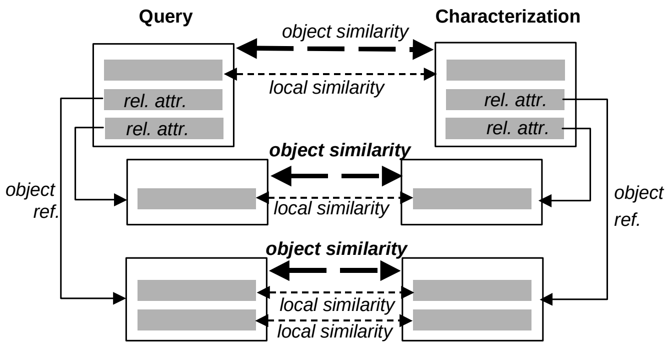

Figure 5: Sketch of the similarity computation

When comparing objects of different classes, the comparison of attributes must be restricted to those attributes that occur in the most specific common subclass both classes belong to. Additionally, one must consider that there is a general difference due to the fact that the objects do belong to different classes. To take this fact into consideration, one introduces a particular similarity measure for comparing classes, called *inter-class similarity measure*. Such an inter-class similarity measure might state that an object of class CD-RW is closer to any object of class CDROM than to any object of class Hard Disk.

In summary, the knowledge encoded in similarity measures for object-oriented representation is structured into:

- one specific local similarity measure for each attribute in each class
- one aggregation function of each class
- one inter-class similarity measure for the class hierarchy.

For the example introduced in Figure 3 and Figure 4, we need to model an individual local similarity measure for each simple attribute, one for manufacturers, one for comparing the read speed, etc. For the read-speed one would usually have a measure that indicates a higher similarity if the difference of the speed values is small and vice versa. For the manufacturer attribute we could have a similarity table with entries for each pair of manufacturers that represent to what degree the components of different manufactures are designed in a similar way (and hence show similar failures). For each individual class in the class hierarchy we need to model an aggregation function (for example a weighted sum) that takes care of the influence of different attributes on the overall similarity of the object. For example, for the purpose of failure diagnosis, the price attribute could be of less importance than the manufacturer attribute. Finally, we need one inter-class similarity measure that might state that an object of class CD-RW is closer to any object of class CDROM than to any object of class Hard Disk.

2.5 Similarity-based Retrieval

When searching for knowledge items, the knowledge need of a user of the OMIS is expressed in a query formalized as a set of related objects. The similarity measure allows the retrieval of knowledge items that do not exactly match the query, but which can differ in many ways. For this purpose the similarity between the query and all characterizations must be assessed. Research in CBR came up with plenty of different retrieval algorithms that help to improve the retrieval efficiency [Bergmann 02], for example by using index structures.

The similarity value determined for each characterization imposes a partial ordering on the knowledge items according to their relevance for the current query. This ranking is an important feedback to the user of an OMIS. Imagine, for example, the user states the following query shown in Figure 6, which describes a problem with a certain CD-ROM.

Query:	**CD-ROM**
	manufacturer: Sony interface: IDE read-speed: 24

Figure 6: A sample query

Given this query, the similarity measure might induce the following ordering on the four knowledge items from Figure 4: C2 > C4 > C3 > C1. Although the order depends on the particular weighting of the attributes in the aggregation functions, the inter-class similarity for the class hierarchy should at least give a rough direction. C2 should be the most similar knowledge item, because it belongs to the same class as the query. From the class membership point of view C4 is also a perfect match, because C4 describes a failure situation that holds for all kinds of optical storage devices, i.e. for all subclasses. Whether the ordering is C2 > C4 or C4 > C2 certainly depends on how the similarity measure weights the particular differences in read speed attribute. C3 will be rated with a lower similarity than C2 and C4 because the failure refers to a different type of component. However, both are optical storage devices. C1 should be rated worse by the inter-class similarity measure since it is most distant in terms of the class hierarchy. This example demonstrates the kind of reasoning by similarity that is performed in a CBR approach.

2.6 Integration of Rules into CBR

Beside the use of similarity measures, CBR research also came up with approaches for integrating rule-based background knowledge [Aamodt 91], [Bergmann 96], [Bergmann 02]. 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 as part of the case in the case base. During retrieval the deductive closure 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.

3 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 meta-physics as a separate discipline. According to [Burkhardt, Smith 01], 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 article we will emphasize only the technical aspects of OBKM and from this perspective we consider ontologies as formal descriptions 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, ontologies 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.

3.1 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 [Lenat, Guha 90], which first addressed the issue of reusability and modularity of large knowledge bases, and the development of the knowledge representation language KL-ONE [Brachmann, Schmolze 85], which was the first logical formalization of a frame-based semantic network. 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 subsumption 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.

Other approaches for developing knowledge-based systems include contexts respective microtheories, compositional modelling, or knowledge composition and merging [Guha 91], [Falkenheimer, Forbus 91], [Clark, Porter 97], [Noy, Musen 00]. Nowadays, research in OBKM focuses on methodologies for introducing and maintaining OBKM systems and addresses important issues like the integration of knowledge processes and knowledge meta-processes into the organizational process [Sure, Studer 03].

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) [Karp 99], the Web Ontology Language (OWL) [Dean, Schreiber 03], the Resource Description Framework (RDF) [Lassila, Swick 99] and the corresponding RDF Schema Specification [Brickley, Guha 03], or XML Topic Maps (XMT) [Pepper, Moore 01].

In the following we will briefly characterize two approaches, RDF(S) and OWL, that already have reached a certain level of maturity.

3.2 RDF(S)

The Resource Description Framework (RDF) [Lassila, Swick 99] 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).

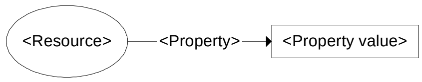

Figure 7: RDF Simple Node and Arc Diagram

A property denotes a named relationship between resources and other objects as property values. It can be visualized as in Figure 7. RDF defines a set of atomic types for property values like strings or integers. Furthermore, an object 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 [Brickley, Guha 03] 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.

3.3 OWL

Although current efforts of the W3C aim to supply a model-theoretic semantic for RDF and RDF Schema [Hayes 03] in order to enable a unique interpretation for automatic reasoning, RDF(S) still lacks the necessary expressive power for many applications. The language OWL [Dean, Schreiber 03] has been developed as a vocabulary extension of RDF and realizes description logics encoded in RDF. OWL is derived from the DARPA Agent Markup Language (DAML) [DARPA 02] and the Ontology Inference Layer (OIL) [Fensel 02], which had been merged into DAML/OIL [Harmelen 01] because of their similarity. As a successor of DAML/OIL, OWL 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, OWL makes use of XML Schema providing a rich set of data types, which are still missing in RDF(S).

3.4 On the Usage of Ontologies in OBKM

Gruber [Gruber 93] 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, is the systematic creation and storage of knowledge assets based on the characterization of knowledge items [Fensel 98]. Here, ontology and characterization are the key for content-based access (filter, retrieve, render, etc.) to knowledge items [Guarino 99]. 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 revisit the example from section 2 and start with the formal model of the object-oriented vocabulary depicted in Figure 3. Because XML-based ontology representation languages are cumbersome to read without appropriate graphical editors, we have chosen an F-Logic like syntax [Kifer 95] for this example. In contrast to KL-ONE, F-Logic does not distinguish between A-Box and T-Box knowledge. Instead, it introduces data-F-atoms for expressing information about objects and signature-F-atoms for expressing information about classes. Both can be combined into F-molecules. Hence, the example ontology shown in Table 1 consists of F-molecules containing only signature-F-atoms as schema for the corresponding objects.

Please note that the property *HD Interface* of class *Mainboard* is a multi-valued method indicated by "=>>" instead of "=>" for single valued methods. Modern main boards often have interfaces of different types (e.g. IDE, SCSI) for hard drives. However, if the main board is assembled in a PC with a connected hard drive, we can infer that the list of possible interface types on the main board at least consists of the interface type of the hard drive. This can be formalized in F-Logic be the following rule:

$$\forall_{M,I,P,H}\ M{:}Mainboard[hd\ interface \text{->} I] \leftarrow P{:}PC[main\ board \text{->} M;\ hard\ disk \text{->} H] \wedge H{:}HardDisk[interface \text{->} I]$$

Defined within the ontology, the rule acts as an axiom that defines a criterion for consistency required by objects of the knowledge base corresponding to the particular classes. Defined outside the ontology, e.g. as part of the problem knowledge of an agent, the rule provides a strategy for resolving missing information for a *Mainboard* object.

Technical Object[price => Real;]	*PC::Technical Object[main board => MainBoard; hard disk => Hard Disk; optional storage => Storage Device; case => Symbol;]*
Storage Device::Technical Object[manufacturer => Symbol; capacity => Real; access time => Real; interface => Symbol;]	*Mainboard::Technical Object[manufacturer => Symbol; CPU Socket => Symbol; HD Interface =>> Symbol;]*
Magnetic Storage Device::Storage Device[built-in size => Disc Size; type of magnetic surface => Symbol;]	*Optical Storage Device::Storage Device[read speed => Integer; transfer rate => Real;]*
Writeable OSD::Optical Storage Device[write speed => Integer;]	*Streamer::Magnetic Storage Device Floppy Disk::Magnetic Storage Device Hard Disk::Magnetic Storage Device CD-ROM::Optical Storage Device CD-Writer::Writeable OSD CD-RW::Writeable OSD 3,5 inch::Disk Size 5,25 inch::Disk Size*

Table 1: Example in F-Logic

3.5 Metadata Characterizations

The link to knowledge items, e.g. PDF-files containing experiences about resolving problems with computer hardware, 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 8 illustrates the usage of RDF for characterizing the examples of Figure 4. The properties of the RDF description are labelled as the slots of the corresponding F-Logic molecules and assign resources as values.

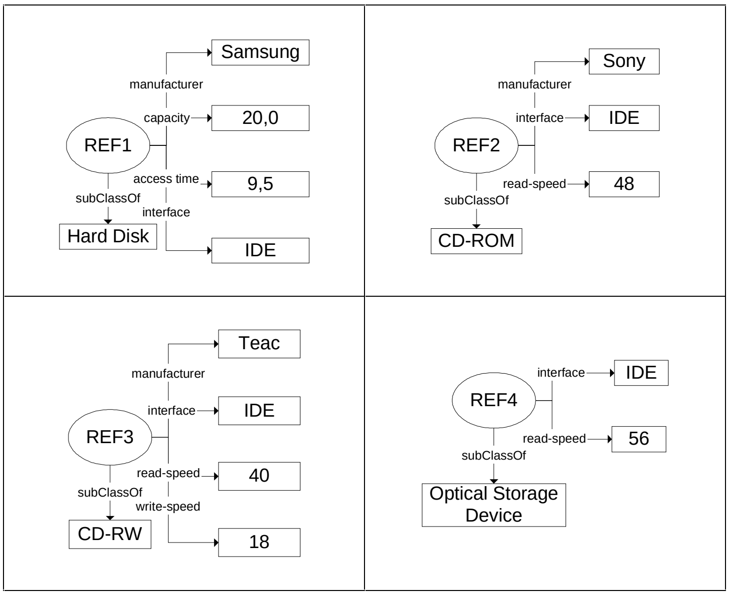

Figure 8: RDF Example Characterizations

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 OWL 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.

3.6 Retrieval based on Deductive Inference

The search for knowledge items in OBKM is usually based on deductive inference. The OBKM application answers queries by proving if it is a consequence from the ontology and the set of characterizations (considered as facts) and thereby finding characterizations that represent the requested documents. A query corresponding to the sample query shown in Figure 6 could be formulated like:

$$\forall_{C} \leftarrow \textit{C:CD-ROM[manufacturer->Sony; interface->IDE; read speed->24]}$$

Unfortunately, none of the characterizations in Figure 8 matches the query exactly and, consequently, the set of possible substitutions for the variable *C* is empty. The strictness of deductive reasoning approaches has been recognized as one of the major problems in weakly structured environments, e.g. the semantic web, and can be tackled by query relaxation techniques as described in [Stuckenschmidt 03]. For instance, a relaxation of the query above would be:

$$\underset{C,R}{\forall} \leftarrow \textit{C:Optical Storage Device[interface->IDE; read speed->R]} \wedge R >= 24$$

In this query, *C* is no longer required to be an instance of the class *CD-ROM* but of the more general class *Optical Storage Device*. Furthermore, no specific manufacturer has been specified and the *read speed* attribute has been weakened to be at least 24. The evaluation of the reformulated query would return all characterizations of Figure 8 with the exception of the *Samsung Hard Drive*. However, the result set has no specific order as shown by the SCBR example. 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.

4 Relations between SCBR 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 9. 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 [Bergmann 99], [Tautz 01], and they are at least partially structured according to the CBR knowledge containers and do address the vocabulary development as well. For OBKM, [Staab 02] and [Sure, Studer 03] follow a KADS oriented methodology and present a meta-process for systematic ontology development.

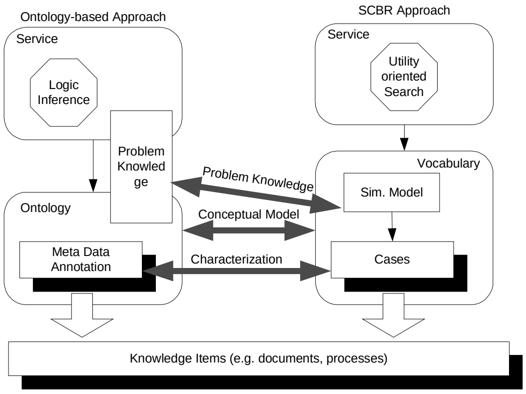

Figure 9: Ontology vs. SCBR Knowledge Containers

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, although research of vocabulary representation languages led to expressive languages [Manago 94], [Arcos, Plaza 95], 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 facilitate the exchange of knowledge. However, current research for distributed CBR [Leake, Sooriamurthi 02] 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 article. As we have also demonstrated with the example, 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 deductive 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 [Bergmann 99]. For realizing the utility-oriented search, SCBR systems introduce an additional 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 linked strongly with the vocabulary representation language. This emphasizes the more problem-oriented approach of SCBR.

The major differences just discussed are finally summarized in Table 2.

SCBR	**OBKM**
mostly isolated: not developed with respect to cooperation with other systems standardization of representation languages not a big issue systems mostly rely on proprietary representations, although XML-based, no standardized semantic vocabularies don‘t conceptualize the domain of discourse per se, but on a task-specific manner	open: cooperation among agents within an ontology-based OMIS is very important W3C standardizations for the semantic web ontologies claim to provide a standardized conceptualization of the domain of discourse ontologies should be „problem free“ (nearly impossible)
Utility-based inference - suitable for many real world applications- not exactly matching solutions can be found	*Logic-based inference* - correct and provable results - required by computer agents for further processing

Table 2: Summary of Differences between SCBR and OBKM

5 Conclusions

Within this article we analysed knowledge management facilitated by SCBR and ontologies. We 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 [Bergmann 99]. 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.

Acknowledgements

We would like to thank Gerd Stumme, Steffen Staab, and Brigitte Bartsch-Spörl for providing valuable input to this article during the *Professional Knowledge Management: Experiences and Visions* conference in Luzern, Switzerland.

References

[Aamodt 91] Aamodt, A.: A knowledge intensive integrated approach to problem solving and sustained learning. PhD Thesis, University of Trondheim, 1991.

[Aamodt, Plaza 94] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, in AICOM, Vol. 7, No. 1, 1994, pp. 39–59.

[Abecker 02] Abecker, A., Hinkelmann, K., Maus, H., Müller, H.J. (Eds.): Geschäftsprozess-orientiertes Wissensmanagement. Springer Verlag, 2002.

[Althoff 00] Althoff, K.D., Bomarius, F., Tautz, C.: Using a case-based reasoning strategy to build learning software organizations. IEEE Journal on Intelligent Systems, 2000.

[Arcos, Plaza 95] Arcos J., Plaza E.: 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, 1995.

[Bergmann 96] Bergmann, R., Wilke, W., Vollrath, I. & Wess, S.: 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, 1996.

[Bergmann 99] Bergmann, R., Breen, S., Göker, M., Manago, M., Wess, S.: Developing industrial case-based reasoning applications. LNAI 1612, Springer, 1999.

[Bergmann 01] Bergmann, R., Richter, M.M., Schmitt, S., Stahl, A., Vollrath, I.: 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, 2001.

[Bergmann 02] Bergmann, R.: Experience Management: Foundations, Development Methodology, and Internet-Based Applications. LNAI 2432, Springer, 2002.

[Brachmann, Schmolze 85] Brachmann R., Schmolze J. G.: An Overview of the KL-ONE Knowledge Representation System. Cognitive Science 9(2): 171-216. 1985.

[Brickley, Guha 03] Brickley D., Guha R. V.: RDF Vocabulary Description Language 1.0: RDF Schema: http://www.w3.org/TR/rdf-schema/.

[Burkhardt, Smith 01] Burkhardt H., Smith B.: Handbook of Metaphysics and Ontology. Munich: Philosophia, 1991.

[Clark, Porter 97] Clark P., Porter B.: Building Concept Representations from Reusable Components. In Proceddings of AAAI '97. Menlo Park, CA: AAAI Press, pp. 369-376, 1997.

[DARPA 02] The DARPA Agent Markup Language Homepage (2002): http://www.daml.org/.

[Dean, Schreiber 03] Dean M., Schreiber G. (Eds.): OWL Web Ontology Language Reference, W3C Working Draft 29 July 2002. http://www.w3.org/TR/owl-ref/.

[Falkenheimer, Forbus 91] Falkenheiner B., Forbus K.: Compositional Modeling: Finding the Right Model for the Job. Artificial Intelligence 51:95-143, 1991.

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

[Fensel 02] Fensel D. et al.: 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, 2002.

[Gruber 93] Gruber T. R.: A translation approach to portable ontologies. Knowledge Acquisition, 5(2): 199-220, 1993.

[Guarino 99] Guarino N., Masolo C., Vetere G.: 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, 1999.

[Guha 91] Guha, R. V.: Contexts: A Formalization and Some Applications. Ph.D. Thesis, Stanford University, 1991.

[Harmelen 01] Harmelen van, F., Peter F. Patel-Schneider, P., Horrocks, I.: Reference description of the DAML+OIL ontology markup language. http://www.daml.org/2001/03/reference.html.

[Hayes 03] Hayes P.: RDF Model Theory, W3C Working Draft 23 January 2003. http://www.w3.org/TR/rdf-mt/.

[Karp 99] Karp P. D., Vinay C. K., Thomere J.: XOL: An XML-Based Ontology Exchange Language. Pangaea Systems and SRI, International, http://www.ai.sri.com/pkarp/xol/, 1999.

[Kifer 95] Kifer M., Lausen G., Wu J.: Logical Foundations of Object Oriented and Frame Based Languages. Journal of ACM 1995, vol. 42, p. 741-843.

[Lassila, Swick 99] Lassila O, Swick R.: Resource Description Framework (RDF) Model and Syntax Specification. http://www.w3.org/TR/REC-rdf-syntax/, 1999.

[Leake 96] Leake, D.: CBR in Context: The Present and Future, in Case-Based Reasoning, Experiences, Lessons, and Future Directions, D. Leake, ed., AAAI/MIT Press, Menlo Park, Calif., 1996, pp. 3–30.

[Leake, Sooriamurthi 02] Leake D. B., Sooriamurthi R.: Automatically Selecting Strategies for Multi-Case-Base Reasoning. In Proceedings ECCBR 2002: 204-233, 2002.

[Lenat, Guha 90] Lenat, D., Guha R.: Building Large Knowledge-Based Systems. Reading, MA: Addison-Wesley, 1990.

[Manago 94] Manago, M., Bergmann, R., Wess, S., Traphöner, R.: CASUEL: A common case representation language. ESPRIT Project INRECA. Deliverable D1, University of Kaiserslautern, 1994.

[Noy, Musen 00] Noy N. F., Musen M. A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. Technical Report SMI-2000-0831. Stanford Medical Informatics, 2000.

[Pepper, Moore 01] Pepper S., Moore G. (Eds.): XML Topic Maps (XTM) 1.0-TopicMaps.Org Specification. http://www.topicmaps.org/xtm/index.html, 2001.

[Richter 95] Richter, M. M.: The Knowledge Contained in Similarity Measures. Invited talk at the First International Conference on CBR (ICCBR-95), 1995.

[Staab 02] Staab, S.: Wissensmanagement mit Ontologien und Metadaten. Informatik Spektrum, 2002.

[Schumacher, Traphöner 00] Schumacher J., Traphöner R.: Knowledge Modelling. Technical Report, WEBSELL Project, Deliverable, 2000.

[Stuckenschmidt 03] Stuckenschmidt H.: Ontology-Based Information Sharing in Weakly-Structure Environments Ph.D. Thesis,Faculty of Sciences, Vrije Universiteit Amsterdam, January 2003.

[Sure, Studer 03] Sure, Y., Studer R.: A Methodology for Ontology-based Knowledge Management. In: Towards the Semantic Web: Ontology-driven Knowledge Management. Davies J., Fensel D., Harmelen van, F. (Eds.), ISBN: 0-470-84867-7, Wiley, 2003.

[Tautz 01] Tautz, C.: Customizing Software Engineering Experience Management Systems to Organizational Needs. PhD Thesis, Department of Computer Science, University of Kaiserslautern, Fraunhofer IRB, Stuttgart, Germany, 2001.