



(KA)² : building ontologies for the Internet: a mid-term report

V. RICHARD BENJAMINS

Social Science Informatics (SWI), University of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands. email: richard@swi.psy.uva.nl

DIETER FENSEL AND STEFAN DECKER

University of Karlsruhe, Institute for Applied Computer Science and Formal Description Methods (AIFB), 76128 Karlsruhe, Germany. email: dfe@aifb.uni-karlsruhe.de, sde@aifb.uni-karlsruhe.de

ASUNCIÓN GÓMEZ PÉREZ

Technical University of Madrid, Department of Artificial Intelligence, 28660 Boadilla del Monte, Madrid, Spain. email: asun@delicias.dia.fi.upm.es

Ontologies are becoming increasingly more important in many different areas, including the knowledge management area. In knowledge management, ontologies can be used as an instrument to make knowledge assets intelligently accessible to people in organizations through an Intranet or the Internet. Most enterprises agree that knowledge is an essential asset for success and survival on an increasingly competitive and global market. In this paper, we present an ontology-based approach through a large-scale initiative involving knowledge management for the knowledge-acquisition research community.

© 1999 Academic Press

1. Introduction

According to Information Week (Angus, Patel & Harty, 1998) “the business problem that knowledge management is designed to solve is that knowledge acquired through experience doesn’t get reused because it isn’t shared in a formal way”. Because this can be knowledge of any kind—tacit, documented, procedural, etc.—the term knowledge management may refer to such various things (Wiig, 1994; O’Leary, 1998) as corporate memories and instincts, expert systems, document managing systems, learning organizations (van Heijst, van de Spek & Kruizinga, 1996), etc.

Knowledge management is neither a product in itself, nor a solution that organizations can buy off-the-shelf or assemble from various components. It is a process implemented over a period of time, which has as much to do with human relationships as it does with business practice and information technology (IT). The process of managing knowledge involves the following actions.

- Knowledge gathering: acquisition and collection of the knowledge to be managed.

- Knowledge organization and structuring: imposing a structure on the knowledge acquired in order to manage it effectively.
- Knowledge refinement: correcting, updating, adding, deleting knowledge, in short: maintaining knowledge.
- Knowledge distribution: bringing the knowledge to the professionals who need it.

In this paper, we present an approach to knowledge management that is based on research on knowledge engineering. Knowledge engineering is a field that—during the past 15 years—has been concerned with capturing, analysing, organizing, structuring, representing, manipulating and maintaining knowledge in order to obtain intelligent solutions for hard problems (Studer, Benjamins & Fensel, 1998; O’Leary, 1997). It is therefore no surprise that knowledge engineering methodologies and techniques can be of high value for knowledge management, which is exactly concerned with the issues mentioned above in a business environment (Schreiber *et al.*, 1999). This paper is about the use of ontologies for knowledge management in Intranets/Internet. There are different uses of ontologies for this purpose including using an ontology as a central corporate memory or for interoperability of databases through a common ontology or for automatic document generation for a particular product. This paper is about one particular use, namely using the ontology for building, maintaining and querying a distributed enterprise knowledge map.

In order for our approach to work in a particular organization, we assume that it has an Intranet/Extranet or access to the Internet and that each member of the organization has a browser. Many companies already have an Intranet, which is an easy-to-use infrastructure that gives companies access to a large variety of Internet techniques. Therefore, for users already familiar with browsers, our approach has a short learning curve. In addition, the approach requires that the knowledge of interest is available in HTML pages on the net or in a format from which such pages can be generated, such as databases or text documents—by wrappers (Kushmerick, Weld & Doorenbos, 1997) or mediators (Wiederhold & Genesereth, 1997).

The main part of the paper discusses a large initiative of knowledge management for the knowledge-acquisition research community. The relevant issues involved are presented and discussed in the context of this research community, involving many researchers. One of the reasons to choose a research community has been that many researchers already have homepages and often put information related to their research areas on the web. Moreover, researchers are often enthusiastic to participate in practical experiments. In Section 2, we review the use of ontologies in knowledge management, and outline the technology underlying our approach. In Section 3, we present an application of our approach for a virtual organization: the knowledge-acquisition research community. We indicate how this case study relates to a business context. In Section 4, we identify a number of possible dangers to successful implementation of knowledge management systems, and we evaluate the case study. Section 5 discusses related work, and Section 6 concludes the paper.

2. An ontology-based approach to knowledge management

2.1. ORGANIZATIONAL MEMORIES

Information technology (IT) support for knowledge management can be characterized according to different schemes: (1) horizontal vs. vertical knowledge management

systems and (2) a process-centred view vs. a product-centred view (Abecker & Decker, 1998).

Horizontal vs. vertical systems: Vertical systems are developed for one particular kind of business situation. Such systems are effective and have proven their value. Often, vertical systems are developed inside a company and are highly situation specific. Therefore, such systems are of little value for other business situations. Horizontal knowledge management systems are general systems that can be applied to a variety of business situations. They are frameworks that can be instantiated to particular situations [see Angus *et al.* (1998) for a discussion of five of such systems: Wincite, Intraspect, ChannelManager, BackWeb and Knowledge X].

Process-centred vs. product-centred: The process-centred view mainly understands KM as a social communication process, which can be improved by various aspects of groupware support. It is based on the observation that the most important knowledge source in an enterprise are employees, and that solving wicked problems is more a process of achieving social commitment than of problem solving. The product-centred view focuses on knowledge documents, their creation, storage and reuse in computer-based corporate memories (OMIS, organizational memory information system). It is based on the notion to have this as a tangible resource or on the idea of supporting the user's individual knowledge development and usage by presenting the right information sources at the right time. Ontologies are often proposed as a means for knowledge dissemination and are used in early OMISs only in an implicit manner. See Ackerman and Malone (1990) and Ackerman and McDonald (1996) for an OMIS called AnswerGarden, where the organization structure can be regarded as an ontology. Abecker, Bernardi, Hinkelmann, Kuhn and Sintek (1998) describe three kinds of ontologies, which are useful for OMISs.

- An information ontology describes the information meta-model, e.g. the structure and format of the information sources. This is the lowest-level ontology.
- A domain ontology is used to describe the content of the information source. This is exactly the kind of ontology we are aiming at.
- An enterprise ontology is used for modelling business processes. Its purpose is to model the knowledge needs in a business process in order to describe a process context, which enables active knowledge delivering.

Concerning the first scheme, this paper is about a horizontal knowledge management system. Concerning the second scheme, this paper is about both aspects (process and product-centred). The building of an ontology is a social process, where different stakeholders have to agree on a shared terminology. Such a process has to be defined, and the right abstractions have to be found, which enable the building of supporting tools for such kind of processes.

The usage of the ontology in itself for knowledge dissemination results in an OMIS. With respect to the above mentioned three types of ontologies, we deal only with the domain ontology. An information ontology is not needed, because we have only two fixed kinds of sources: annotated HTML pages and collections of similar-structured HTML pages. These are web pages, all of which use the same structure (i.e. HTML layout) to present their information, such that it pays off to construct a wrapper that makes the information of the web pages available to an OMIS. A typical example for

such a collection of web pages is the CIA-World Factbook (<http://www.odci.gov/cia/publications/factbook/index.html>). Furthermore, because we use a horizontal approach, the enterprise ontology is not needed. However, if the system described later is integrated into a specific workflow, the use of this kind of ontology would enable the active delivering of knowledge relevant to the context in any specific business process.

2.2. APPROACH

Our approach comprises three main sub-tasks: (1) ontological engineering to build an ontology of the subject matter, (2) characterizing the knowledge in terms of the ontology and (3) providing intelligent access to the knowledge. In a sense, this is reminiscent of relational database technology, where the ontology would correspond to the data model, the characterization would correspond to the instances (data) contained in the database, and access would take place through SQL. We will show, however, that our approach is significantly different from centralized databases, especially with respect to distributiveness and intelligence. Our approach captures distributive, rather than centralized knowledge. The knowledge is directly accessed at its original location (in HTML pages) rather than be separately input to a database. The approach allows to infer knowledge that is not explicitly known, but that can be deduced based on general knowledge (captured in the ontology). For example, in the context of human-resource management, if in some company only senior managers can lead projects, and Mr Paton is project leader, then we can deduce that Mr Paton is a senior manager, even though this is nowhere stated explicitly.

Figure 1 gives a general overview of the approach. An ontology of the subject matter has to be built, which is used to characterize the subject matter (i.e. to fill the ontology with instances). An intelligent web crawler receives a query in terms of the ontology, consults the subject matter (the instances), interprets them using the ontology and generates an answer. The instances (the actual knowledge to be managed) are distributed over different HTML pages (of an Intranet or the Internet).

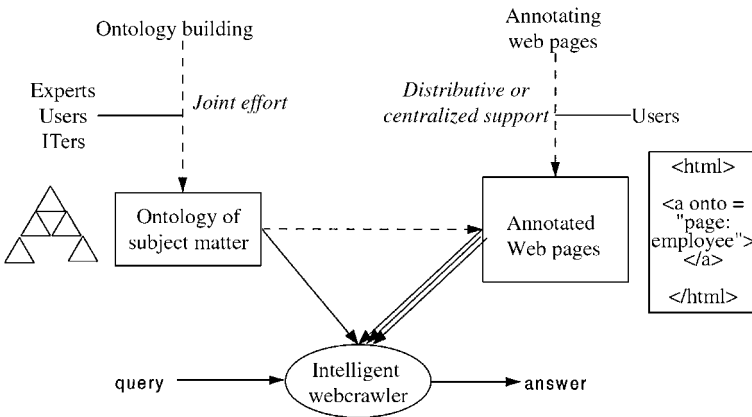


FIGURE 1. The approach.

2.2.1. *Ontological Engineering*

An ontology is a shared and common understanding of some domain that can be communicated across people and computers (Gruber, 1993; Guarino, 1995; Uschold & Gruninger, 1996; van Heijst, Schreiber & Wielinga, 1997). Ontologies can therefore be shared and reused among different applications (Farquhar, Fikes & Rice, 1997), which is one of the main reasons why ontologies are popular nowadays. An ontology can be defined as a formal, explicit specification of a shared conceptualization (Gruber, 1993; Borst, 1997). “Conceptualization” refers to an abstract model of some phenomenon in the world by identifying the relevant concepts of that phenomenon. “Explicit” means that the type of concepts used and the constraints on their use are explicitly defined. “Formal” refers to the fact that the ontology should be machine readable. “Shared” reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group. An ontology describes the subject matter using the notions of concepts, instances, relations, functions and axioms. Concepts in the ontology are organized in taxonomies through which inheritance mechanisms can be applied.

In order to come up with a consensual ontology of some domain, it is important that the people who have to use ontology have a positive attitude towards it. Dictating the use of a particular ontology to people to which they have not contributed, is not likely to succeed. Preferably, an ontology is constructed in a collaborative effort of domain experts, representatives of end-users and IT specialists. Such a joint effort requires (1) the use of a methodology that guides the ontology development process and (2) tools to inspect browse, codify, modify and down-load the ontology. Examples of such methodologies include METHONTOLOGY (Fernandez, Gomez-Perez, Pazos & Pazos, 1999), Uschold’s and Grüninger’s methodology (Uschold & Gruninger, 1996) and that of Grüninger and Fox (Grüninger & Fox, 1995). These methodologies have in common the features that they start from the identification of the purpose of the ontology and the need for domain knowledge acquisition. The Ontology Server (Farquhar *et al.*, 1997) is the best-known environment for building ontologies. It is an interactive environment especially useful for updating, maintaining and browser ontologies. Ontolingua ontologies can be translated to different languages, including Prolog, CORBA’s IDL (Orfali, Harkey & Edwards, 1996), CLIPS, LOOM (MacGregor, 1991), KIF, Epikit (Genesereth, 1992). Ontologies built in Ontolingua use the Frame Ontology (Gruber, 1993), which is written in Knowledge Interchange Format (KIF) (Genesereth & Fikes, 1992). The Frame Ontology is, as its name suggests, a frame-based language which includes primitives such as classes, sub-classes, attributes, values, relations and axioms. Related ontologies can be connected to each other by inclusion. Other languages used to formalize ontologies include CycL (Lenat & Guha, 1990) and Loom (MacGregor, 1991). CycL is the language used in the Cyc project, whose goal is to construct a foundation of basic common-sense knowledge. Loom is the language used by the server called Ontosaurus (Swartout, Patil, Knight & Russ, 1997).

As an example, consider the context of the automobile industry. Here, the ontology would include, among others, terms related to mechanical and hydraulic devices. In the mechanical device ontology, examples of classes are “cylinder”, “crankshaft” and “engine”. An example of a binary relation is “part-of”, which could be used to say that the cylinder is part-of the engine. The hydraulic device ontology could include the class

```

Concept: Component

Relation : Part-of
Number of arguments: 2
Type of argument #1: component
Type of argument #2: component

```

FIGURE 2. Part of a physical device ontology.

```

Concept: Cylinder
Subclass-of: Component
Part-of: Engine

Concept: Crankshaft
Subclass-of: Component
Part-of: Engine

Concept: Engine
Subclass-of: Component

```

FIGURE 3. Part of a mechanical device ontology.

“pipe” and the ternary relation “connection” to express that two mechanical devices are connected by a given kind of pipe. Note that the terms “cylinder”, “crankshaft” and “engine” will be part of an ontology in the domain of “mechanical devices”, while the concept “component” and the relation “part-of” will belong to a meta-ontology, applicable to any kind of physical device. Figures 2 and 3 illustrate, respectively, part of a physical device ontology and part of a mechanical device ontology.

In a human-resource management context, classes could be “employee”, “manager”, “project leader”, “skill”, “area of expertise”. Applied to a concrete company, an ontology can fulfil the role of an “enterprise knowledge map”. Notice that terms from the human-resource management ontology are reusable for any kind of company. However, terms from the automobile ontology are only reusable for automobile companies.

2.2.2. Characterizing the knowledge

As already mentioned briefly, in our approach, the knowledge to be managed is distributively organized in HTML pages [e.g. in a company’s Intranet or on the world wide web (WWW)]. The relevant knowledge can thus be maintained distributively by different persons (the responsible persons for the respective HTML pages). The subject matter knowledge within the HTML pages is annotated using the ontology as a scheme for expressing meta-data. For example, in the human-resource management domain, the homepage of Mr Paton would state that he is a project leader. We thus add meta-data to make this explicit. In our approach, we do this by extending HTML with a new attribute of the “anchor” tag: the `onto` attribute. Figure 4 gives a simple illustration.

```

<html>
<head><TITLE> Mr. Paton </TITLE>
<a ONTO = ``page: ProjectLeader`` >/a>
</head>
<body>
.....
<a ONTO = ``page [lastName = body]``>Paton</a>
.....
</body>
</html>

```

FIGURE 4. A simple extension to HTML. The ONTO attribute allows to express ontological information in HTML pages.

The HTML code in Figure 4 states that the URL of the page containing the information represents a *ProjectLeader* (a term defined in the ontology). Page in `` refers to the URL of the web page. Body refers to what follows and what is within the scope of the anchor, i.e. until the closing ``. The `onto` attribute does not affect the visualization of HTML documents in standard web browsers such as Netscape or Explorer. The only thing that it does, is that it makes visible the subject matter knowledge for the intelligent web crawler. This small extension of HTML has been chosen to keep annotation as simple as possible. Also, it enables the direct usage (actually, reuse) of textual knowledge already in the body of the anchor. This prevents the knowledge annotator from representing the same piece of information again (the text *Paton* appearing as the value of meta-data `onto` above, is the same text as is visualized in the browser). This simple solution suffices for our approach because the HTML pages contain only factual knowledge (Fensel, Decker, Erdmann & Studer, 1998).

2.2.3. *Ontology browsing*

Ontologies play an important role in our approach, and participants need to use the ontology actively in the annotation and querying process. Therefore, in this section, we devote attention to easy browsing of ontologies. An ontology can be represented as a hierarchy of concepts. With respect to the handling of this hierarchy, a user has at least two requirements: first, he wants to scan the vicinity of a certain class looking for classes better suitable to formulate a certain query. Second, a user needs an overview over the whole hierarchy to allow a quick and easy navigation from one class in the hierarchy to another class. These requirements are met by a presentation scheme based on Hyperbolic Geometry (Lamping & Rao, 1996): classes in the centre are depicted by a large circle, whereas classes at the border of the surrounding circle are only marked with a small circle (see Figure 5). This visualization technique allows a quick navigation to classes far away from the centre as well as a closer examination of classes and their vicinity. Classes can be dragged around while the size of the visualization nodes changes corresponding to their location, that is, the more centric the bigger they appear (see Figure 6).

2.2.4. *Intelligent knowledge retrieval*

Having discussed the ontology and the annotated HTML pages, we will now turn to using this knowledge for intelligent retrieval. We use the ontology-based brokering

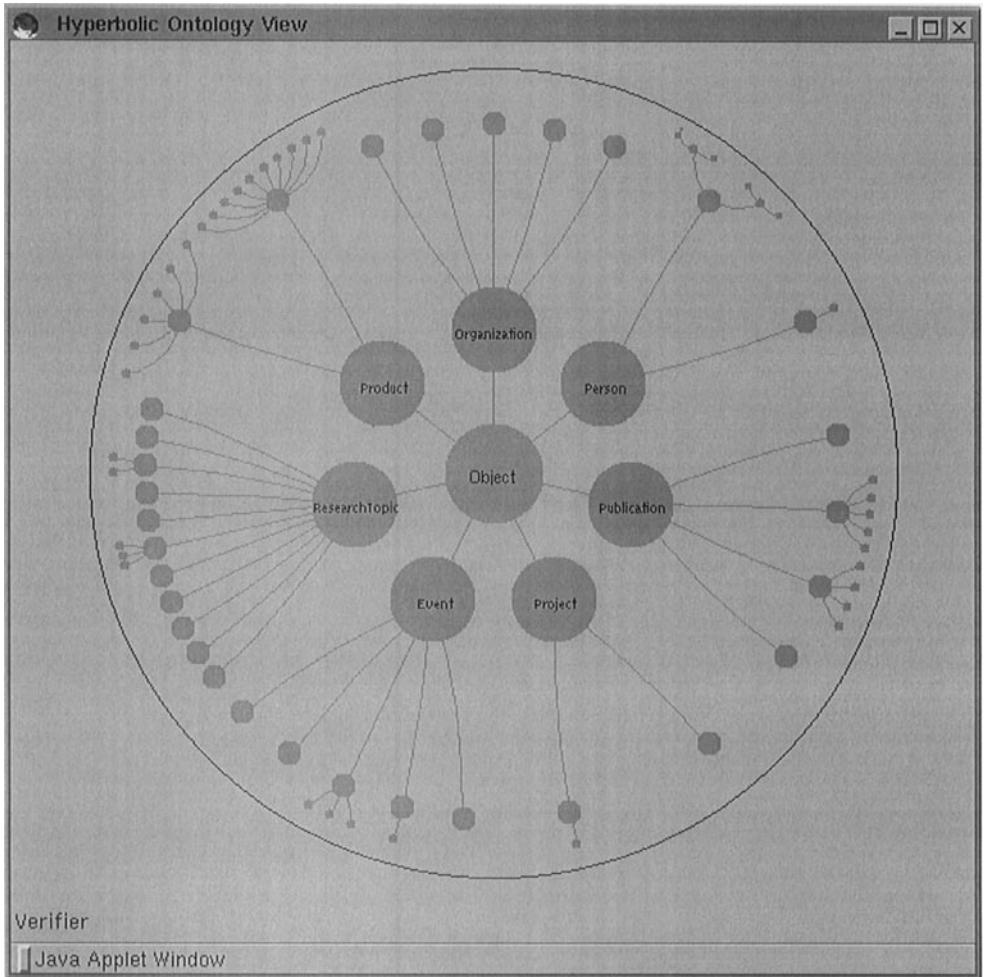


FIGURE 5. The hyperbolic view of an ontology.

service Ontobroker,[†] which consists of three main elements: a web crawler (called Ontocrawler), an inference engine and a query interface (Fensel *et al.*, 1998).

First, Ontocrawler searches through the annotated pages (e.g. on an Intranet) and collects the annotated knowledge fragments. Second, it translates the annotated knowledge fragments into facts formulated in the representation language used by Ontobroker. Neither the inference engine nor the querying user have to be aware of the syntactical way in which the facts are represented on the Internet. Only the annotaters have to use the annotation language.

The inference engine receives the query of a user and exploits two information sources for deriving an answer: the ontology of the subject matter and the facts that were found

[†]The URL of Ontobroker is <http://www.aifb.uni-karlsruhe.de/WBS/broker/>

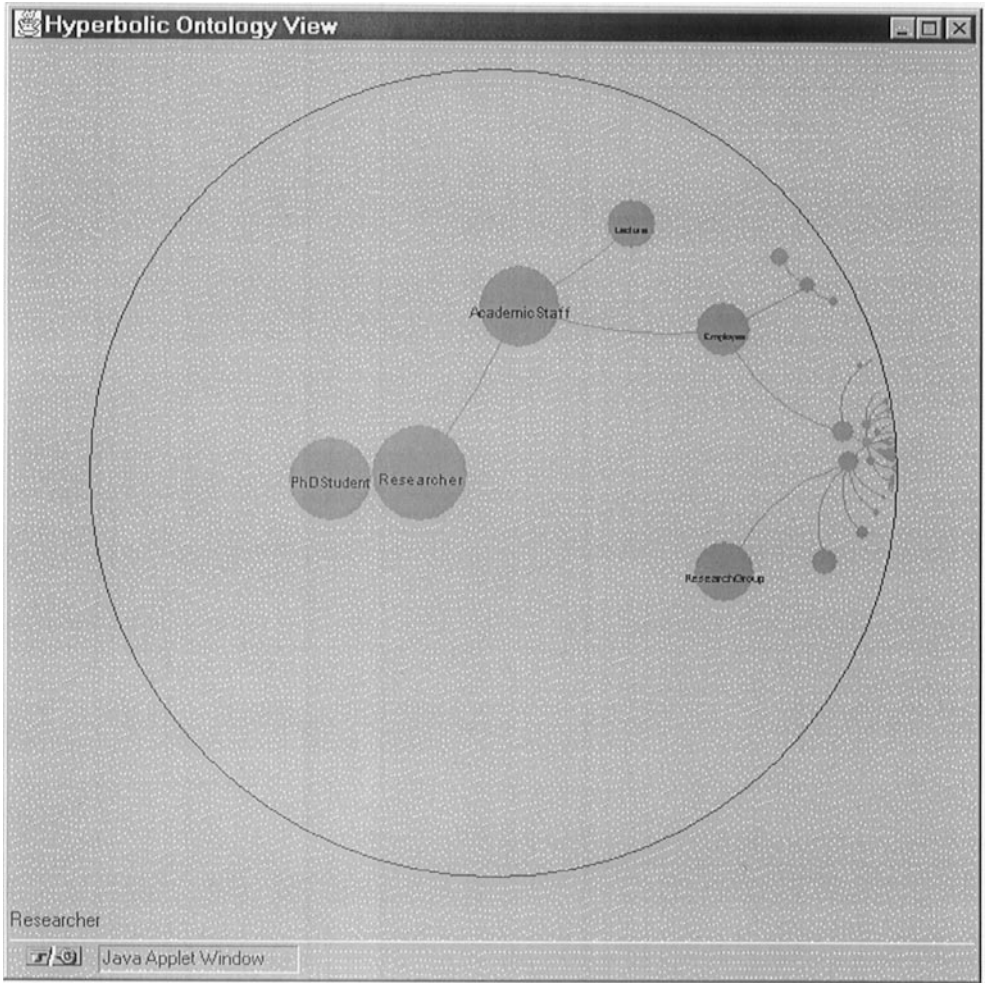


FIGURE 6. Using the hyperbolic view to zoom in on researcher.

by Ontocrawler. The basic inference mechanism of the inference engine is the derivation of a minimal model of a set of Horn clauses (see Fensel *et al.*, 1998 for details). This resembles intelligent reasoning as known in Knowledge-Based Systems, with the difference that the instances of the knowledge base are now distributed over the different HTML pages. The query interface of Ontobroker lets the user formulate specific queries to topics related to knowledge acquisition (KA). It consists of a textual and graphical interaction mode (see Section 3.3).

3. The (KA)² initiative

In order to investigate the feasibility of our approach, we are performing a large-scale initiative on the web, where the subject matter is the scientific knowledge acquisition

community: the Knowledge Annotation Initiative of the Knowledge-Acquisition Community†: (KA)². We describe thus a virtual organization consisting of researchers, universities, projects, publications, etc. The information resides at the WWW in the homepages of the KA researchers where they publish information about their affiliation, projects, publications, research interests, etc. (Benjamins & Fensel, 1998).

From a concrete knowledge management point of view, the (KA)² initiative is not an esoteric, academic toy example. Imagine a large multinational company with thousands of employees worldwide. For such a large organization, effective human-resource management (HRM) is of vital importance. However, finding “who knows what” in large organizations has always been a time-intensive process. A knowledge management system that allows to find adequate people based on their skills, experience and area of expertise would certainly be of high value. For large companies that have an organization-wide Intranet, our approach is a real possibility to enhance the HRM task. It allows improvement of the precision and presentation of the results of searches on an Intranet or the WWW.

Notice, however, the fact that (KA)² is naturally related to the HRM task, does not imply that it is limited to this knowledge management task. In principle, the subject matter of our approach can concern any kind of company-vital knowledge that needs to be managed more effectively.

3.1. ONTOLOGICAL ENGINEERING IN (KA)²

In (KA)², we build an ontology of the KA community (cf. an “enterprise knowledge map”). Since an ontology should capture consensual knowledge, in (KA)², several researchers cooperate together — at different locations — to construct the ontology. In this way, we ensure that the ontology will be accepted by a majority of KA researchers. The design criteria used to build the (KA)² ontology were: modularity, to allow more flexibility and a variety of uses, specialization of general concept into more specific concepts, classification of concepts by similar features to guarantee inheritance of such features and standardized name conventions. The current ontology for the KA community consists of seven related ontologies: an organization ontology, a project ontology, a person ontology, a research-topic ontology, a publication ontology, an event ontology and a research-product ontology. A first release of this ontology was built in F-logic (Kifer, Lausen & Wu, 1995), which is the language that Ontobroker uses. To make this ontology accessible to the entire community, it was decided to translate this (F-logic) ontology to Ontolingua and to make it accessible through the Ontology Server. Translation between the F-logic and Ontolingua formats of the ontology is performed automatically by translators, which form part of ontological design environment (ODE) (Blázquez, Fernandez, Garcia-Pinar & Gomez-Perez, 1998). These translators ensure that the F-logic and Ontolingua versions of the ontology are always in sync.

The current version of the ontology can be viewed at the European mirror site in Madrid of the Ontology Server of Stanford University‡. For illustration purpose, we

†The homepage of (KA)² is <http://www.aifb.uni-karlsruhe.de/WBS/broker/KA2.html>

‡URL is <http://www-ksl-svc-lia.dia.fi.upm.es:5915/Login> as “ontologias-ka2” with password “adieu007”, and then load one of the seven sub-ontologies of the KA community.

include here examples of two sub-ontologies of the KA ontology: the person ontology and the publication ontology.

The Person ontology defines the types of persons working in academic environments, along with their characteristics. This ontology defines 10 classes and 23 relations. The overview does not show which classes the relations connect (but it can be browsed at Ontology Server). Indentation denotes the sub-class of relation.

Class hierarchy (10 classes defined):

```

Person
  Employee
    Academic-Staff
      Lecturer
      Researcher
    Administrative-Staff
      Secretary
      Technical-Staff
  Student
    Phd-Student

```

23 relations defined:

```

Address, Affiliation, Cooperates-With, Editor-of,
Email, First-Name, Has-Publication, Head-Of-Group,
Head-Of-Project, Last-Name, Member-Of-Organization,
Member-Of-Program-Committee, Member-Of-Research-Group,
Middle-Initial, Organizer-Of-Chair-Of, Person-Name,
Photo, Research-Interest, Secretary-Of, Studies-At,
Supervises, Supervisor, Works-At-Project

```

The Publication-ontology defines—in 13 classes and 28 relations—the usual bibliographic entities and attributes.

Class hierarchy (13 classes defined):

```

On-Line-Publication
Publication
  Article
    Article-In-Book
    Conference-Paper
    Journal-Article
    Technical-Report
    Workshop-Paper
  Book
  Journal
    IEEE-Expert

```

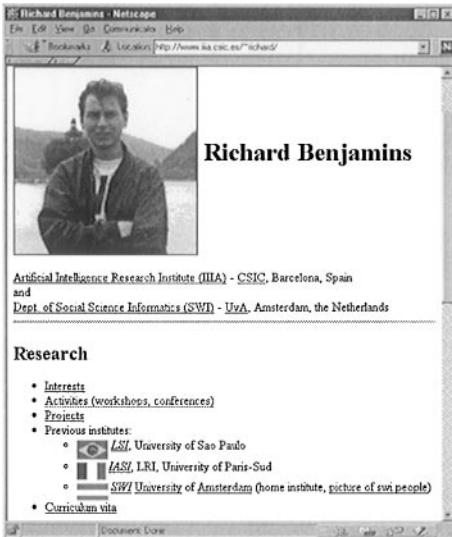
IJHCS
Special-Issue

28 relations defined:

Abstract, Book-Editor,
Conference-Proceedings-Title,
Contains-Article-In-Book,
Contains-Article-In-Journal, Describes-Project,
First-Page, Has-Author, Has-Publisher, In-Book,
In-Conference, In-Journal, In-Organization,
In-Workshop, Journal-Editor, Journal-Number,
Journal-Publisher, Journal-Year, Last-Page,
On-Line-Version, On-Line-Version-Of,
Publication-Title, Publication-Year,
Technical-Report-Number, Technical-Report-Series,
Type, Volume, Workshop-Proceedings-Title

3.2. ANNOTATING PAGES IN (KA)²

Annotating HTML pages in (KA)² means that each participating researcher in the KA community has to annotate the relevant knowledge in his or her homepage environment. Figure 7 illustrates fragments of an annotated homepage of a researcher using the onto attribute. Page in $\langle a \text{ ONTO} = \text{'page [address = body]'} \rangle$ refers to the URL of the web page. Body refers to what follows and what is within the scope of the anchor, i.e. until the



```
<HTML><HEAD>
<a onto = 'page: Researcher'></a>
<TITLE>Richard Benjamins</TITLE>
</HEAD>
<H1><A HREF = 'pictures/id-rich.gif'>
<IMG SRC = 'pictures/richard.gif'></A>
<a onto = 'page [photo = href]'
  HREF = 'http://www.iiia.csic.es/richard/
    pictures/richard.gif'></a>

<a onto = 'page [firstName = body]'>Richard</a>
<a onto = 'page [lastName = body]'>Benjamins</a>
</h1><p>

<A onto = 'page[affiliation = body]'>
Artificial Intelligence Research Institute
(IIIA)</A>-
<a href = 'http://www.csic.es/'>CSIC</a>,
Barcelona, Spain<br>
HREF = 'http://www.uva.nl/uva/english/'>UvA</a>
```

FIGURE 7. Example web page annotated with the ONTO attribute. Page in $\langle a \text{ ONTO} = \text{'page [address = body]'} \rangle$ refers to the URL of the page. Body refers to what follows and what is within the scope of the anchor, i.e. until the closing $\langle /a \rangle$. Address is a term of the KA ontology.

```

        <a name = ``Benjamins:97a`` ONTO = ``name:JournalArticle``></a>
        <a name = ``Benjamins:97a`` ONTO = ``name [author = href]``
          href = ``http://www.iiaa.csic.es/~richard/index.html``>
V. R. Benjamins
and
        <a name = ``Benjamins:97a`` ONTO = ``name [author = body]``>
M. Aben
        <a name = ``Benjamins:97a`` ONTO = ``name [title = body]``>
Structure-Preserving KBS Development through
Reusable Libraries: a Case-Study in Diagnosis.
        <a name = ``Benjamins:97a`` ONTO = ``name [journal = body]``>
IJHCS
International Journal of Human-Computer Studies, Vol 47, pages
        <a name = ``Benjamins:97a`` ONTO = ``name [firstPage = body]``>
259
-
        <a name = ``Benjamins:97a`` ONTO = ``name [lastPage = body]``>
288
'
        <a name = ``Benjamins:97a`` ONTO = ``name [year = body]``>
1997
        <a name = ``Benjamins:97a`` ONTO = ``name [onlineVersion = href]``
HREF = ``http://www.iiaa.csic.es/~richard/postscripts/ijhcs.ps``>
(draft version)

```

FIGURE 8. Example of an annotated publication. All values of the ONTO attribute belong to the ontology of the knowledge-acquisition community. The actual knowledge (the instances) representing the publication appears on the left-hand side, the right part contains the annotation code.

closing ``. Address is a term of the KA ontology. Figure 8 illustrates the annotation of a publication. Directly annotating the HTML source of a document is a tedious task. A knowledge provider has to examine in parallel the ontology and the HTML code of the page to annotate. If there are any coincidences, he has to write the annotation directly in the HTML source, using terms from the ontology.

Direct annotations of the source code is inconvenient for several reasons. First, the knowledge provider does not have an easy readable view on the document (he sees the HTML source instead of a nicely formatted page). Second, browsing the ontology without tool support is uncomfortable and time consuming as it means browsing a large text file and looking for the ontology terms of interest. Finally, manually writing annotations into the HTML source easily leads to typing errors, which would make the annotation useless. In spite of the amount of work involved,[†] this is the way in which the majority of the current pages are annotated in (KA)². An important factor that probably helped here is self-publicity. By annotating pages, researchers make themselves more visible to others, which enhances the likelihood that others will use and refer to their work, which — in the academic world — is a good thing.

To overcome the problems associated with directly annotating HTML code, a prototype annotation tool — Ontopad — has been developed (as an extension of an existing

[†]Our experience is that it takes roughly one hour to annotate five pages. At the Ontobroker site, an annotation checker is available to detect syntax errors.



FIGURE 9. Ontopad in action. Adding an annotation to a selected portion of text.

HTML editor), which allows normal browsing of both the HTML page to be annotated and the ontology. Parts of the HTML page and terms of the ontology can be selected with the mouse. The knowledge provider can select a portion of the text from a web page

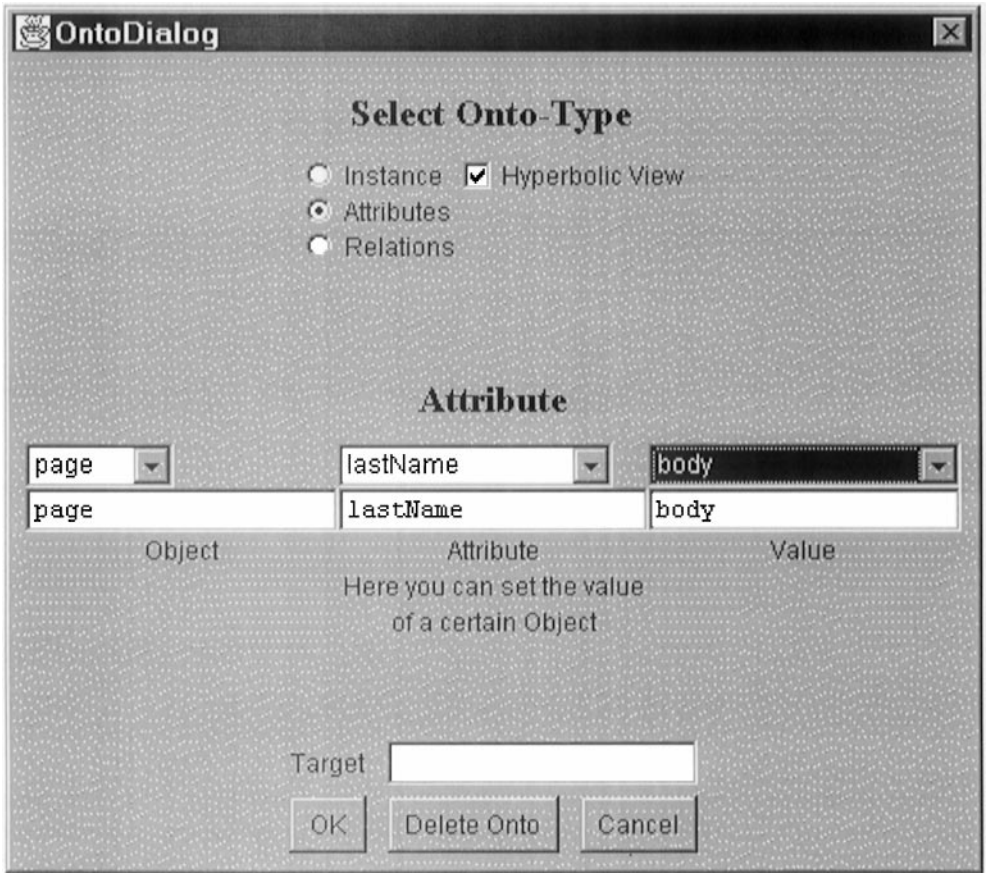


FIGURE 10. Ontotag annotation Dialog. Choosing what annotation to add and starting up the hyperbolic view.

and choose to add an annotation. In the example shown in Figure 9, the knowledge provider selects the text “Benjamins” and chooses to add an annotation (Edit/Insert Ontotag in Figure 9). Then a dialog window pops up (see Figure 10), which gives him the possibility to open the hyperbolic view on the ontology (Figure 5). In Figure 6, the knowledge provider selects the class “Researcher”. Now, all attributes of “Researcher” become available for selection in the dialog window. In Figure 10, the knowledge provider chooses the attribute “lastName”, thereby saying that “Benjamins” is the last name of the page owner. When he clicks “OK”, the annotation is inserted into the HTML text.

3.3. QUERYING THE KA COMMUNITY

In (KA)², in order for Ontocrawler to collect the knowledge from HTML pages, researchers have to register their pages. That is, they have to tell Ontocrawler which

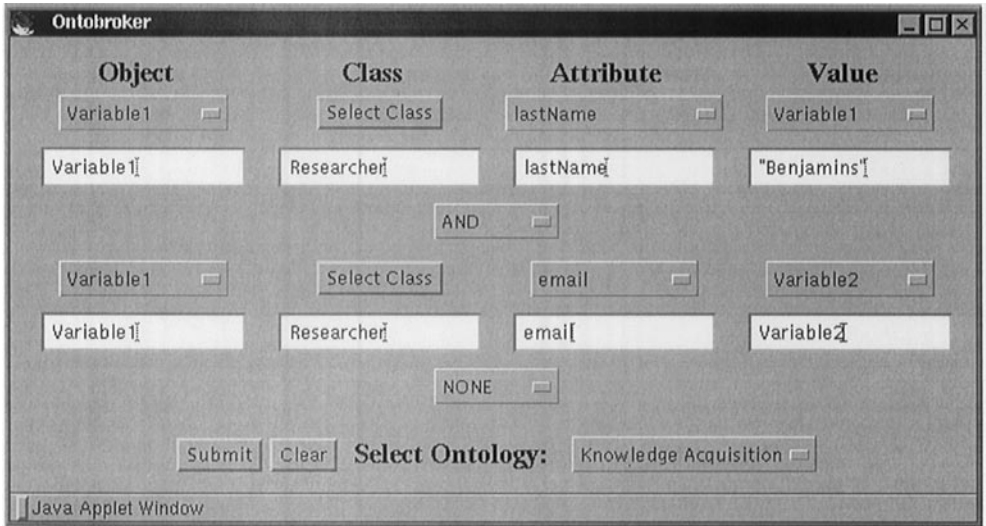


FIGURE 11. The table query interface.

URLs it needs to visit. Once that is done, intelligent knowledge retrieval is possible. There are two query interfaces: a text-based interface for expert users and a graphical interface for naive users. The text-based interface allows the direct formulation of queries in F-logic of which the general structure is: Object: Class [Attribute->> value], e.g. Object: Researcher [lastName->> Benjamins]. However, the direct formulation of such queries has two drawbacks: (1) the user has to know the syntax of the query language (F-logic) and (2) the user also has to know the ontology when formulating a query. To overcome these drawbacks, a graphical interface has been developed, which consists of two parts: (1) an ontology browser where terms for the query can be selected by clicking (the same as used in Ontopad, i.e. the hyperbolic view) and (2) a table interface to compose queries from the selected ontology terms. The ontology browser frees users from knowing the complete ontology by heart, whereas the table interface frees users from knowing the syntax of F-logic.

Selecting ontology terms for composing the query, means — in the implementation — to select classes in F-logic. The hyperbolic view allows such selection (see Section 2.2.3). When a user selects a class from the hyperbolic ontology view, the class name appears in the class field (Figure 11) and the user can select one of the attributes from the attribute choice menu because the pre-selected class determines the possible attributes (such as “lastName” and “email”). The interface also allows the construction of composite queries using conjunctives such as *and*, *or*, and *not*, or *not*.

The interface is programmed in Java as an applet, thus it is executable on all major platforms where a web-browser with Java support exists. Based on these interfaces, Ontobroker automatically derives the query in textual form and presents the result of the query.

We can for instance ask for all researchers in the KA community. The answer would not only include researchers who have their homepage annotated, but also additional

researchers who cooperate with these researchers. The ontology defines cooperation between researchers, which enables the following deduction: if X cooperates with Y then X and Y must be researchers. Ontobroker uses this type of information, not for consistency checking (which would not be a very good idea in an open web environment), but for abductively deriving new facts (i.e. Y is also a researcher). This example illustrates that it is possible to access knowledge that is not explicitly represented, which is an important advantage of our approach compared to keyword-based search. We could also ask for all researchers who have worked together in some project or for abstracts of all papers on a particular topic. More examples of queries to the knowledge-acquisition community can be obtained through Ontobroker's homepage.

3.4. SOME FACTS

The current version (July 1998) of the ontology contains 80 classes, 27 axioms and 100 attributes, which are used to annotate 1000 facts of around 20 researchers.

4. Feasibility of knowledge management systems

In order to say something about the feasibility of a horizontal knowledge management approach such as we have described and applied to the knowledge-acquisition community, we have to consider the risks involved. Risks come from various resources, and we will discuss them resourcewise; technological risks and social and organizational risks.

4.1. TECHNOLOGICAL RISKS

From a technology point of view, there are several factors that endanger the success of our knowledge management approach.

Tool support. First of all, such an initiative is likely to fail without dedicated tools to support the tasks involved. In particular, tools are needed for (1) constructing and maintaining the ontology, (2) annotating information sources and (3) querying them (see Figure 12). We have discussed (prototype) tools for annotation and querying. Concerning construction and maintenance of the ontology, we used the ontological design environment, (ODE; Blázquez *et al.*, 1998) which allows one to specify ontologies at the conceptual level by completing tables, rather than at the implementation level. From these tables, ODE is able to generate the Ontolingua code of the ontology. We need, however, to complement this with more support. For instance, WebOnto (Domingue, 1998) enables collaborative construction of ontologies over the WWW.

Maintenance. An approach is needed to maintain knowledge, both at the instance level, where researchers annotate their personal pages, as well at the ontology level. Changes to the ontology might have dramatic consequences for updating the annotations in HTML pages, especially in pages that are annotated with an ontology term that has become obsolete. Moreover, often there will not be a simple one-to-one correspondence between the old and the new ontology terms. In order to solve the issues involved with a changing ontology, one has to consider at least two aspects of the problem.

- The first, more technical, problem is how to deal with knowledge expressed in old ontology terms (in HTML pages), while Ontobroker uses an updated, newer ontology.

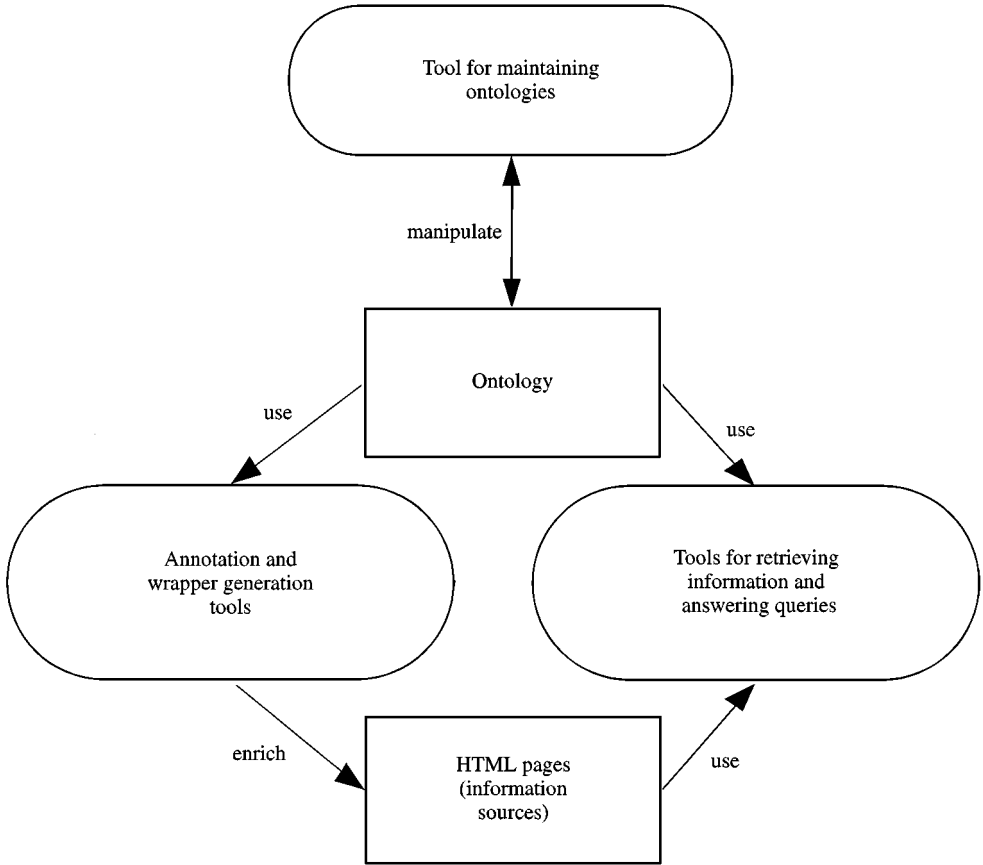


FIGURE 12. Tools to support knowledge management.

- The second is a social problem: how to organize the process to encourage people to update their web pages using the new terminology? A first observation is that such a process should avoid radical changes and enforcements, which generally discourages people from following it.

For these issues, we need a process model. Currently, we think the following process model is adequate, but we still need to demonstrate it in practice.

- (1) If a term from the ontology is replaced by one or more other terms, it is marked with a “deprecated” flag. That means that it does not show up any more in the query interface (see Figure 5) and annotation tool. This way, only up-to-date terms can be used for querying and annotation.
- (2) However, there may exist many HTML pages that use the old ontology terms. In order to have the knowledge defined in those pages still available, mapping rules have to be defined. These mapping rules map the old terminology (which is used in

the antecedent of the rules) to the new terminology (which is used in the consequent of the rule). There may not always exist an exact mapping, but a partial translation is often useful and the loss of information can be determined (Mena, Kashyap, Illarramendi & Sheth, 1998; Jannink, Pichai, Verheijen & Wiederhold, 1998).

- (3) After a while, knowledge providers, who still use the old ontology terms can be warned, because the source of each fact is known.
- (4) As a last step, the mapping rules and the old ontology have to be deleted. Old knowledge of providers who have not updated their pages, will not be available any more. This is a loss, but apparently, such providers do not maintain their pages, and the knowledge would probably be outdated anyway.

As mentioned before, we have to evaluate this process model and especially, we have to find the time constraints for it—e.g. how long should the old ontology terms be supported?

Scaling up. What happens when the knowledge is spread over ten thousands of HTML pages? Apart from the updating problem (see above) also the intelligent reasoning part might become a problem. This is a familiar problem in KBS research, when algorithms developed and tested on toy domains have to scale up to real-world applications. There is, however, a continuum in ontology building that can be exploited to enhance performance. This continuum relates to the number and type of rules defined in the ontology. If performance becomes a problem, the number and/or generality of the rules in the ontology can be reduced. We then trade-off “easy and elegance of modelling” against efficiency. In the extreme case, we drop all rules (which would reduce modelling comfort considerably!), and the approach boils down to a database lookup problem, which is known to be very efficiently solvable.

4.2. SOCIAL AND ORGANIZATIONAL RISKS

Minimum amount of participants. Without participating researchers, the (KA)² initiative would certainly fail. However, the nature of the initiative is such that participation is rewarding. It is a self-promoting activity. That is, researchers are better off if they participate because other researchers and outsiders can more easily find their work.

Competitive mentality. In many companies, the mentality is competitive rather than collaborative. In other words: “If my colleague wins, then I lose”. And: “If I make my knowledge available to others, then others will profit from that, and there will be a risk that they outperform me.” This mentality is a real threat to success of knowledge management initiatives. Increasingly more companies have become aware that a collaborative mentality leads to better results than competitive thinking (Covey, 1989). Organizations can stimulate collaborative thinking by changing the incentive system (such as making it financially rewarding to share knowledge).

Incentive system. Given the high workload of today’s employees, it may easily be felt that contributing to a knowledge management effort is a waste of time, or at least does not have priority. This is killing for any knowledge management initiative. Organizations should therefore reward knowledge management contributions equally as results that lead to direct profits. In addition, an effort should be made to reuse existing documents such that knowledge workers do not have the impression that they have to

duplicate knowledge. Tools to generate HTML pages from a variety of other formats (MSWord, email, etc.) are already in existence.

4.3. (KA)²: A SUCCESS STORY OR A FAILURE?

A final answer to the question whether (KA)² should be regarded as a success or as a failure is still too early. However, some intermediate evaluations can be provided. We will briefly sketch the social side of the initiative (i.e. the process), and the resulting ontology and web-based information sources (i.e. the products). Finally, we will discuss the role of the available tools for it.

First, (KA)² is still an initiative without funding. Therefore, it depends heavily on the enthusiasm of its participants and on the indirect benefits like publications and publicity. Although there are currently around 30 active participants, it must be stated that most of the effort has been carried out by a relatively small core group of participants. At several meetings, conferences and workshops, (KA)² has attracted a broad, but floating and often arbitrarily composed group of people. We decided therefore to have a next plenary meeting in 1999 focusing only on improving (KA)² in the near future.

Second, (KA)² has produced an ontology, which is currently used to describe about 1000 facts on the web. The ontology consists of two main parts: a general ontology useful for describing organizations, persons, publications, etc., and a specialized ontology useful for describing research topics of the knowledge-acquisition community. The former part is straightforward and could be developed quickly, mainly by reusing existing data models and ontologies. The latter, more interesting, part will be finalized in 1999 and will cover most of the research topics of knowledge acquisition and, in addition, some related scientific areas. The overall organization of the topics and the structure of describing each topic has been established in several plenary workshop meetings, and is shown in Figure 13. This ontology has to be instantiated for different research topics covered by the knowledge-acquisition community. The topics that were identified in a number of meetings are: rescue, problem-solving methods, ontologies, validation and verification, specification languages, knowledge-acquisition methodologies, agent-oriented approaches, knowledge acquisition from natural language, knowledge management, knowledge acquisition through machine learning, knowledge acquisition through conceptual graphs, foundations of knowledge acquisition, evaluation of knowledge-acquisition technique and methodologies and knowledge elicitation. Each of these topics has been given to a small group of experts that completed the scheme in Figure 13. This distributed approach worked well for some of the topics and did not produce any results for others. It required a significant additional effort of the core committee to fill in the gaps and to achieve some level of homogeneity in describing the different topics. The current version of the research topics of ontology can be viewed at the (KA)² homepage.

Why only 1000 facts? The factual knowledge out there in the web is still rather small. Only 20 persons annotated some of their web pages. However, as long as the ontology is under development this is not a bad situation. The annotations provide prototypical query access to web documents without requiring too much maintenance effort. Only when consensus on the full-fledged ontology is achieved (yielding a much richer ontology than we currently have), a more serious annotation effort makes sense.

```

Class:research-topic
Attributes:
  Name: <string>
  Description: <text>
  Approaches: <set-of keyword>
  Research-groups: <set-of research-group>
  Researchers: <set-of researcher>
  Related-topics: <set-of research-topic>
  Subtopics: <set-of research-topic>
  Events: <set-of events>
  Journals: <set-of journal>
  Projects: <set-of project>
  Application-areas: <text>
  Products: <set-of product>
  Bibliographies: <set-of HTML-link>
  Mailing-lists: <set-of mailing-list>
  Webpages: <set-of HTML-link>
  International-funding-agencies: <funding-agency>
  National-funding-agencies: <funding-agency>
  Author-of-ontology: <set-of researcher>
  Date-of-last-modification: <date>

```

FIGURE 13. Ontology to characterize research topics.

Finally, the available tool environment played a minor role in (KA)². So far, mail tools and ASCII editors have been the main means for achieving the goals. However, this will change when the initiative grows from its embryo status to a large-scale enterprise. Until now, a highly motivated group has bypassed all shortcomings of the existing tool environments. However, attracting a less “idealistic” and broader audience will require more sophisticated tools. Especially, when the size of the annotated knowledge body grows from a toy example to a real-world application, maintenance of annotations will become a nightmare. When the ontology is viewed as a living entity with significant and periodical changes, one has to think about efficient maintenance support. However, this requirement already holds for web documents in general. Already now, the WWW contains much outdated information, and many documents contain inconsistent information. At the moment, the exponential growth of the web hides these problems because most of the retrieved documents still have a short living period. When the WWW reaches a period with normalized growth, the problem of outdated information and maintenance will become one of the most challenging problems of the WWW.

5. Related work

5.1. WEB STANDARDS.

How does our simple extension of HTML relate to new technologies for the web, that might make HTML obsolete? The W3C (the international World Wide Web Consortium for developing and promoting standards for usage on the WWW) promotes the following related standards.

- XML [extensible markup language (XML, 1998)] is a modification of SGML (standard generalized markup language, ISO Standard 8879) designed under the auspices of the W3C. XML is not a markup language like HTML, rather it is a meta-language, which can be used to define customized markup languages. XML is designed to describe the *structure* of a document, not the *content*. We aim to capture the content of a document, so XML is not a replacement for our annotation language. However, the annotation language can be easily combined with XML.
- RDF [resource description framework (RDF, 1998)] is a framework for describing general-purpose meta-data. Instead of just providing a set of keywords for tagging, it allows a more flexible way to represent meta-data. As far as we can see, RDF will be a widely recognized language and a representation formalism, that can serve as a worldwide interlingua for information interchange. However, an RDF description of a document is separated from the document and cannot be easily generated from it, so maintenance of meta-data expressed in RDF might result in high efforts: if the data is changed, also the meta-data has to be changed to keep both in sync.
- RDFS [RDF schema (RDFS, 1998)] is a format to define the terminology, that can be used to describe RDF descriptions. It basically allows to define classes, attributes (property types), value ranges for attributes. So RDFS allows to define basic ontologies for RDF specifications. However, it is not possible to specify more complex relationships. Especially, it is not possible to define rules such as those contained in our ontology. So, as it stands now, RDFS is not a replacement for our ontology language.

To have the advantages of our annotation language and ontology formalism (easy maintenance and deductive power) together with the advantages of RDF (worldwide-recognized representation formalism), we developed RDF-Maker, that generates automatically RDF-meta-data out of our annotation formalism (Decker, Erdmann, Fensel & Studer, 1999). From an example homepage, RDF-Maker infers from the original set of seven explicit facts (that have been annotated on that homepage) several additional facts. All these facts represent information (explicitly and implicitly) contained on the presented homepage which are now available for RDF-enabled agents. Furthermore, our inference engine is also able to reason with RDF-based meta-data (Brickley, Saarela & Angele, 1998).

5.2. ONTOLOGY-BASED VS. KEYWORD-BASED RETRIEVAL

One could argue that, if all the knowledge is available in HTML documents, then why use an ontology to annotate the information in the pages? After all, the annotation effort is considerable. Why not use general search engines for keyword-based searching through the HTML pages? As everybody might have experienced, keyword-based search easily leads to an overwhelming amount of answers (references to web documents). In other words, there is an information overload (O'Leary, 1997), which makes it hard to find exactly what one is looking for and to get rid of nonsense (with respect to the query). Although search engines get increasingly smarter, we expect that there will be a limit to such keyword-based information retrieval. Moreover, current keyword-based search approaches do not allow to present information collected from distributive locations in a coherent way to users, since there is no knowledge of how the retrieved information

relates to each other. Ontology-based retrieval does allow for this, through the ontology. Finally, the ontology-based approach allows to access implicit knowledge, which is definitely beyond the capacity of keyword-based approaches.

To reduce the annotation effort, machine learning techniques can be used that exploit ontologies to automatically classify textual information (Craven *et al.*, 1998). Moreover, wrappers can be built that extract the semantics of web documents based on regularities in their structure, format and content. Again, machine learning techniques can be used to semi-automatically build such wrappers (Ashish & Knoblock, 1997; Kushmerick *et al.*, 1997). Clearly, this is an important research line to embark on.

5.3. ONTOLOGIES ON THE WEB

While approaches like XML (XML, 1998) and RDF (RDF, 1998) provide standardized representation formalisms for meta-information, approaches like Dublin Core (DC, 1998) and the Warwick Framework (Lagoze, Lynch & Jr, 1996) provide standardized vocabularies that meta-data can be formulated with. The SHOE project (Luke, Spector, Rager & Hendler, 1997) uses meta-data to improve information retrieval from the web. However, in SHOE, providers of information can introduce arbitrary extensions to a given ontology. Furthermore, no central provider index is defined. As a consequence, when specifying a query the client may not know all the ontological terms he can use to annotate HTML pages, and the web crawler may miss knowledge fragments because it cannot parse the entire WWW. In contrast, Ontobroker relies on the notion of a group of Web users that agree on an ontology for a given subject. Therefore, both the information providers and the clients have complete knowledge of the available ontological terms.

6. Conclusion

In this article, we presented a knowledge engineering approach to knowledge management, which is based on many years of experience in dealing with knowledge. If we relate our work to the four knowledge management actions mentioned in the introduction, we get the following.

- Knowledge gathering is performed from existing HTML pages (knowledge annotation).
- Knowledge organization and structuring is done through an ontology (ontological engineering).
- Knowledge refinement is performed distributively by each worker (update annotations).
- Knowledge distribution is done by a web crawler that gives intelligent access to the knowledge that is “managed”. This is a pull approach where users take the initiative when they need knowledge. However, the work presented here could as well be used for a push approach.

We noted that knowledge management also involves people, and therefore any knowledge management effort is doomed to fail if human factors are not taken seriously. Knowledge management only works if people cooperate and are willing to share their

knowledge. One way to stimulate sharing of knowledge is to change the incentive system accordingly. Another important social aspect is that, in order to be successful with a large-scale initiative, lightweight tools are a must.

We presented and discussed our approach to knowledge management through a large initiative for the knowledge-acquisition community concerned with ontology building on the Internet, knowledge annotation of web pages and query answering. The results so far highlight both the technical and social issues when dealing with knowledge management. We also discussed solution directions for some technical problems (e.g. tool support). Social issues concern above all people's willingness to participate in a knowledge management initiative, which, in a scientific community, seems more feasible than in a commercial organization.

One of the aims of (KA)² is to provide better and easy access to relevant information about knowledge acquisition. A simple alternative for this aim would be to provide a knowledge-acquisition meta-page with a collection of all kinds of relevant links for knowledge acquisition. Examples of such web sites in knowledge acquisition include the KAW mailing list (<http://www.swi.psy.uva.nl/mailling-lists/kaw/home.html>) and the KAW archives page at: <http://ksi.cpsc.ucalgary.ca/KAW/>. Indeed, for building the research topics ontology of knowledge acquisition, such web sites have been heavily used as useful sources of information. In the near future, we plan to extend the ontology to include also case-based reasoning.

We would like to thank all people who contributed to building the research topic ontology: F. Mitchell, Derek Sleeman, M. Winter, Robin Boswell, Susan Crow, Enrico Motta, Arthur Stutt, Annette ten Teije, Frank van Harmelen, Rose Dieng, Jan Treur. We also thank Ulf Steinberg. This research is partly funded by the program "Acciones Integradas Hispano-Alemanas" reference HA 1998-002, by the "Ministerio de Educación y Cultura. Dirección General de Enseñanza Superior e Investigación Científica". Richard Benjamins was supported by the Netherlands Computer Science Research Foundation with financial support from the Netherlands Organization for Scientific Research (NWO).

References

- ABECKER, A., BERNARDI, A., HINKELMANN, K., KÜHN, O. & SINTEK, M. (1998). Towards a technology for organizational memories. *IEEE Intelligent Systems*, **13**, 40–48.
- ABECKER, A. & DECKER, S. (1998). Topics in organisational memory. In P. MARTI & S. BAGNARA, Eds. *Designing Collective Memories, 7th Le Travail Humain Workshop*, Paris.
- ACKERMAN, M. S. & MALONE, T. W. (1990). A tool for growing organizational memory. *Proceedings of the ACM Conference on Office Information Systems*, pp. 31–39. Center for Coordination Science, Massachusetts Institute of Technology.
- ACKERMAN, M. S. & MCDONALD, D. W. (1996). Answer garden 2: Merging organizational memory with collaborative help. *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW'96)*, pp. 97–105, Department of Information and Computer Science, University of California, Irvine, CA 92717.
- ANGUS, J., PATEL, J., & HARTY, J. (1998). Knowledge management: great concept ... but what is it. *Information Week*.
- ASHISH, N. & KNOBLOCK, C. A. (1997). Semi-automatic wrapper generation for internet information sources. *Proceedings of the IFCIS Conference on Cooperative Information Systems (CoopIS), Charlston, SC*.
- BENJAMINS, V. R. & FENSEL, D. (1998). The ontological engineering initiative (KA)². In N. Guarino, Ed. *Formal Ontology in Information Systems (FOIS'98)*, pp. 287–301. IOS Press.

- BLÁZQUEZ, M., FERNÁNDEZ, M., GARCÍA-PINAR, J. M. & GÓMEZ-PÉREZ, A. (1998). Building ontologies at the knowledge level using the ontology design environment. *Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling and Management, KAW'98*, Banff, Canada.
- BORST, W. N. (1997). *Construction of engineering ontologies*. Ph.D. Thesis, University of Twente, Enschede.
- BRICKLEY, S. D. D., SAARELA, J. & ANGELE, J. (1998). A query and inference service for rdf. *Proceedings of the W3C Query Language Workshop (QL-98)*, Boston.
- COVEY, R. (1989). *The Seven Habits of Highly Effective People*. New York: Simon & Schuster, Inc.
- CRAVEN, M., DIPASQUO, D., FREITAG, D., MCCALLUM, A., MITCHELL, T., NIGAM, K. & SLATTERY, S. (1998). Learning to extract symbolic knowledge from the world wide web. *Proceedings of the 15th National Conference on AI (AAAI-98)*, Madison, WI.
- DC (1998). Dublin core. <http://purl.org/metadata/dublin-core>.
- DECKER, S., ERDMANN, M., FENSEL, D. & STUDER, R. (1999). Ontobroker: ontology based access to distributed and semi-structured information. In R. MEERSMAN et al., Eds. *Semantic Issues in Multimedia Systems*, pp. 351–369. Boston: Kluwer Academic Publisher.
- DOMINGUE, J. (1998). Tazebao and webonto: discussing, browsing, and editing ontologies on the web. *Proceedings of the 11th Workshop on Knowledge Acquisition, Modeling and Management, KAW'98*, Banff, Canada.
- FARQUHAR, A., FIKES, R. & RICE, J. (1997). The ontolingua server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, **46**, 707–728.
- FENSEL, D., DECKER, S., ERDMANN, M. & STUDER, R. (1998). Ontobroker: the very high idea. *Proceedings of the 11th International Flairs Conference (FLAIRS-98)*, Sanibel Island, FL.
- FERNANDEZ, M., GOMEZ-PEREZ, A., PAZOS, J. & PAZOS, A. (1999). Ontology of tasks and methods. *IEEE Intelligent Systems and their Applications*, **14**, 37–46.
- GENESERETH, M. R. Ed. (1992). *The Epikit Manual*. Palo Alto, CA: Epistmemics, Inc.
- GENESERETH, M. R. & FIKES, R. E. (1992). *Knowledge interchange format, version 3.0, reference manual*. Technical Report, Logic-92-1, Computer Science Dept., Stanford University. <http://www.cs.umbc.edu/kse/>.
- GRUBER, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, **5**, 199–220.
- GRÜNINGER, M. & FOX, M. (1995). Methodology for the design and evaluation of ontologies. *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing held in Conjunction with IJCAI-95*, Montreal, Canada.
- GUARINO, N. (1995). Formal ontology, conceptual analysis and knowledge representation. *International Journal of Human-Computer Studies*, **43**, 625–640 (Special issue on The Role of Formal Ontology in the Information Technology).
- JANNINK, J., PICHAI, S., VERHEIJEN, D. & WIEDERHOLD, G. (1998). Encapsulation and composition of ontologies. *Proceedings of the AAI-98 WS on AI and Information Integration*. Madison, WI: AAI Press.
- KIFER, M., LAUSEN, G. & WU, J. (1995). Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*.
- KUSHMERICK, N., WELD, D. & DOORENBOS, R. (1997). Wrapper induction for information extraction. *Proceedings of the 15th International Joint Conference on AI (IJCAI-97)*, pp. 729–735, Nagoya, Japan.
- LAGOZE, C., LYNCH, C. A. & JR., R. D. (1996). *The warwick framework: a container architecture for aggregating sets of metadata*. Technical Report TR96-1593, Computer Science Technical Report Library.
- LAMPING, L. & RAO, R. (1996). A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, **7**, 33–55.
- LENAT, D. B. & GUHA, R. V. (1990). *Building Large Knowledge-based Systems. Representation and Inference in the Cyc Project*. Reading, MA: Addison-Wesley.
- LUKE, S., SPECTOR, L., RAGER, D. & HENDLER, J. (1997). Ontology-based web agents. *Proceedings of 1st International Conference on Autonomous Agents*. <http://www.cs.umd.edu/projects/plus/SHOE/>.

- MACGREGOR, R. (1991). Inside the LOOM classifier. *SIGART Bulletin*, **2**, 70–76.
- MENA, E., KASHYAP, V., ILLARRAMENDI, A. & SHETH, A. (1998). Domain specific ontologies for semantic information brokering on the global information infrastructure. In N. GUARINO, Ed. *Formal Ontology in Information Systems (FOIS'98)*, pp. 269–283. IOS Press.
- O'LEARY, D. (1997). The internet, intranets, and the AI renaissance. *IEEE Computer*, **30**, 71–78.
- O'LEARY, D. (1998). Knowledge management: taming the information beasts. *IEEE Intelligent Systems*, **13**, 30–48 (Special Issue with three contributions).
- ORFALL, R., HARKEY, D. & EDWARDS, J., Ed. (1996). *The Essential Distributed Objects Survival Guide*. New York: Wiley.
- RDF (1998). Resource description framework. <http://www.w3.org/TR/WD-rdf-syntax>.
- RDFS (1998). Resource description framework schema. <http://www.w3.org/TR/1998/WD-rdf-schema-19980814>.
- SCHREIBER, A. T., AKKERMANS, J. M., ANJEWIERDEN, A. A., DE HOOG, R., SHADBOLT, N. R., DE VELDE, W. V. & WIELINGA, B. J. (1999). *Engineering and Managing Knowledge, The Common-KADS Methodology*. Cambridge, MA: MIT Press.
- STUDER, R., BENJAMINS, V. R. & FENSEL, D. (1998). Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, **25**, 161–197.
- SWARTOUT, W., PATIL, R., KNIGHT, K. & RUSS, T. (1997). Toward distributed use of large-scale ontologies. In *Spring Symposium Series on Ontological Engineering*, pp. 33–40. Stanford: AAAI Press.
- USCHOLD, M. & GRUNINGER, M. (1996). Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, **11**, 93–155.
- VAN HEIJST, G., SCHREIBER, A. T. & WIELINGA, B. J. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies*, **46**, 183–292.
- VAN HEIJST, G., VAN DE SPEK, R. & KRUIZINGA, E. (1996). Organizing corporate memories. In B. R. GAINES & M. A. MUSEN, Ed. *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, pp. 42.1–42.18, Alberta, Canada. SRDG Publications, University of Calgary. <http://ksi.cpsc.ucalgary.ca:80/KAW/KAW96/KAW96Proc.html>.
- WIEDERHOLD, G. & GENESERETH, M. (1997). The conceptual basis for mediation services. *IEEE Intelligent Systems*, **12**, 38–47.
- WIIG, K. M. (1994). *Knowledge Management, The Central Management Focus for Intelligent-acting Organizations*. Arlington, TX: Schema Press Ltd.
- XML (1998). Extensible markup language. <http://www.w3.org/TR/PR-xml-971208>.