Journal of Universal Computer Science, vol. 10, no. 3 (2004), 262-283
submitted: 30/1/04, accepted: 27/2/04, appeared: 28/3/04 © J.UCS

OntoShare – An Ontology-based Knowledge Sharing System for Virtual Communities of Practice[1]

John Davies
(BTexact, Orion 5–12, Adastral Park,
Ipswich IP5 3RE, UK
john.nj.davies@bt.com)
Alistair Duke
(BTexact, Orion 5–12, Adastral Park,
Ipswich IP5 3RE, UK
Alistair.duke@bt.com)
York Sure
(Institute AIFB, University of Karlsruhe
D-76128 Karlsruhe, Germany
sure@aifb.uni-karlsruhe.de)

Abstract: An ontology-based knowledge sharing system OntoShare and its evaluation as part of a case study is described. RDF(S) is are used to specify and populate an ontology, based on information shared between users in virtual communities. We begin by discussing the advantages that use of Semantic Web technology afford in the area of knowledge management tools. The way in which OntoShare supports WWW-based communities of practice is described. Usage of OntoShare semi-automatically builds an RDF-annotated information resource for the community (and potentially for others also). Observing that in practice the meanings of and relationships between concepts evolve over time, OntoShare supports a degree of ontology evolution based on usage of the system – that is, based on the kinds of information users are sharing and the concepts (ontological classes) to which they assign this information. A case study involving OntoShare was carried out. The evaluation exercise for this case study and its results are described. We conclude by describing avenues of ongoing and future research.

Key Words: Knowledge Sharing Community, Semantic Web

Category: H.4.3 [Information Systems Application]: Communications Applications – *Information browsers*

1 Introduction

1.1 Motivation

There are now several billion documents on the WWW, which are used by more than 300 million users globally, and millions more pages on corporate intranets. The continued rapid growth in information volume makes it increasingly difficult

[1] A short version of this article was presented at the I-KNOW '03 (Graz, Austria, July 2–4, 2003).

to find, organize, access and maintain the information required by users. A semantic web that provides enhanced information access based on the exploitation of machine-processable metadata has been proposed [BLHL01]. We are particularly interested in the new possibilities afforded by semantic web technology in the area of knowledge management and we discuss this below before moving on in the rest of the paper to describe OntoShare, a system for supporting Semantic Web-based communities of practice.

Central to the vision of the Semantic Web are ontologies. Ontologies are seen as facilitating knowledge sharing and re-use between agents, be they human or artificial [Fen01]. They offer this capability by providing a consensual and formal conceptualization of a given domain. As such, the use of ontologies and supporting tools offer an opportunity to significantly improve knowledge management capabilities in large organizations and it is their use in this particular area that is the subject of this paper. In OntoShare, an ontology specifies a hierarchy of concepts (ontological classes) to which users can assign information. In this process, important metadata is extracted and associated with the community information resource using RDF annotations.

1.2 Knowledge Management, Communities of Practice & the Semantic Web

The notion of communities of practice [SBD91] has attracted much attention in the field of knowledge management. Communities of practice are groups within (or sometimes across) organizations who share a common set of information needs or problems. They are typically not a formal organizational unit but an informal network, each sharing in part a common agenda and shared interests or issues. In one example it was found that a lot of knowledge sharing among copier engineers took place through informal exchanges, often around a water cooler. As well as local, geographically based communities, trends towards flexible working and globalization has led to interest in supporting dispersed communities using Internet technology [Dav00]. The challenge for organizations is to support such communities and make them effective. Provided with an ontology meeting the needs of a particular community of practice, knowledge management tools can arrange knowledge assets into the predefined conceptual classes of the ontology, allowing more natural and intuitive access to knowledge.

Knowledge management tools must give users the ability to organize information into a controllable asset. Building an intranet-based store of information is not sufficient for knowledge management; the relationships within the stored information are vital. These relationships cover such diverse issues as relative importance, context, sequence, significance, causality and association. The potential for knowledge management tools is vast; not only can they make better use of the raw information already available, but they can sift, abstract and help to share new information, and present it to users in new and compelling ways.

In this paper, we describe the OntoShare system which facilitates and encourages the sharing of information between communities of practice within (or perhaps across) organizations and which encourages people – who may not previously have known of each other's existence in a large organization – to make contact where there are mutual concerns or interests. As users contribute information to the community, a knowledge resource annotated with metadata is created. Ontologies are defined using RDF Schema (RDFS) and populated using the Resource Description Framework (RDF). (RDF [LS99] is a W3C recommendation for the formulation of metadata for WWW resources. RDFS [LS99] extends this standard with the means to specify domain vocabulary and object structures – that is, concepts and the relationships that hold between them). In the next section, we describe in detail the way in which OntoShare can be used to share and retrieve knowledge and how that knowledge is represented in an RDF-based ontology. We then proceed to discuss in Section 3 how the ontologies in OntoShare evolve over time based on user interaction with the system and motivate our approach to user-based creation of RDF-annotated information resources. We describe in Section 4 the deployment and evaluation of OntoShare in a particular community as part of a case study in the project On-To-Knowledge (see `http://www.ontoknowledge.org`). In Section 5 we first sketch the underlying methodology, then we illustrate the application and evaluation of OntoShare according to the methodology. We provide results of a user-focused evaluation of OntoShare in Section 6. Before concluding, we present further and related work.

2 Sharing and Retrieving in OntoShare

OntoShare is an ontology-based WWW knowledge sharing environment for a community of practice that models the interests of each user in the form of a user profile. In OntoShare, user profiles are a set of topics or ontological concepts (classes declared in RDFS) in which the user has expressed an interest. OntoShare has the capability to summarize and extract key words from WWW pages and other sources of information shared by a user and it then shares this information with other users in the community of practice whose profiles predict interest in the information. OntoShare is used to store, retrieve, summarize and inform other users about information considered in some sense valuable by an OntoShare user. This information may be from a number of sources: it can be a note typed by the user him/herself; it can be an intra/Internet page; or it can be copied from another application on the user's computer. As we will see below, OntoShare also modifies a user's profile based on their usage of the system, seeking to refine the profile to better model the user's interests.

2.1 Sharing Knowledge in OntoShare

When a user finds information of sufficient interest to be shared with their community of practice, a 'share' request is sent to OntoShare via the Java client that forms the interface to the system. OntoShare then invites the user to supply an annotation to be stored with the information. Typically, this might be the reason the information was shared or a comment on the information and can be very useful for other users in deciding which information retrieved from the OntoShare store to access. At this point, the system will also match the content being shared against the concepts (ontological classes) in the community's ontology. Each ontological class is characterized by a set of terms (keywords and phrases) and the shared information is matched against each concept using the vector cosine ranking algorithm [Har92]. The system then suggests to the sharer a set of concepts to which the information could be assigned. The user is then able to accept the system recommendation or to modify it by suggesting alternative or additional concepts to which the document should be assigned.

When information is shared in this way, OntoShare performs four tasks:

1. An abridgement of the information is created, to be held on the user's local OntoShare server. OntoShare's summarizer extracts key theme sentences from the document. It is based on the frequency of words and phrases within a document, using a technique based on lexical cohesion analysis [AB95]. Access to this locally held summary enables a user to quickly assess the content of a page from a local store before deciding whether to retrieve the (larger amount of) remote information.

2. The content of the page is analyzed and matched against every user's profile in the community of practice. As when recommending concepts to the user, the vector cosine ranking model is used: here, however, the shared information is matched against the set of terms (words and phrases) created from the union of all terms associated with the concepts to which has user has subscribed (i.e. the concepts which make up the user profile). If the profile and document match strongly enough, OntoShare emails the user, informing him or her of the page that has been shared, by whom and any annotation added by the sharer.

3. The information is also matched against the sharer's own profile in the same way. If the profile does not match the information being shared, the system will suggest one or more concepts which strongly match the shared information that the user can then add to their profile. Thus OntoShare has the capability to adaptively learn users' interests by observing user behavior.

4. For each document shared, an instance of the class *Document* is created, with properties holding metadata including keywords, an abridgement of

the document, document title, user annotation, universal resource locator (URL), the sharer's name and date of storage. (The ontological structure of the OntoShare store is shown in detail in the next section.) In this way, a shared and enhanced information resource is built up in OntoShare based on user contributions.

Given that users must make a conscious decision to store information, the quality of the information in the OntoShare store is high – it is effectively pre-filtered by OntoShare users. Thus each user leverages the assessment of the information made by all the other users.

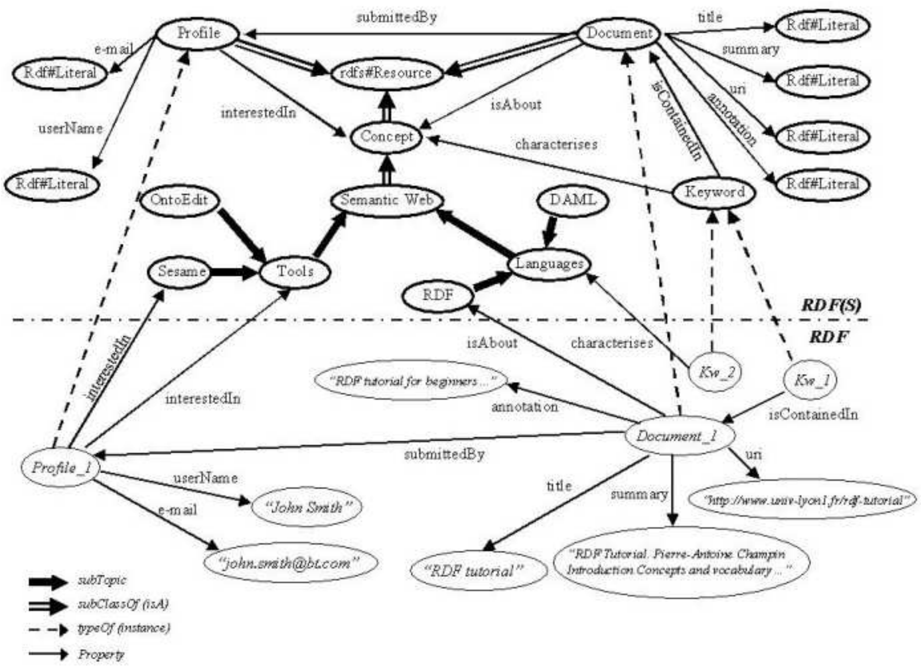

Figure 1: Ontological Structure in OntoShare

2.2 Ontological Representation

We said above that each piece of shared information leads to the creation of a new entry in the OntoShare store and that this store is effectively an ontology represented in RDFS and RDF. We now set this out in more detail. RDFS is used to specify the classes in the ontology and their properties. RDF is then used to populate this ontology with instances as information is shared. Figure 1 shows a slightly simplified version of the ontology for a community sharing information about the Semantic Web, along with an example of a single shared document ("Document_1").

It is not our intention to describe each class and property and their function here but we will mention a few key aspects. Firstly, notice *Concept* and its subclasses: this is the set of concepts which the community of practice at hand is interested in. Note that in the current version of OntoShare, the concept structure is limited to a strict hierarchy. Another key class is *Document*, which is the class used to represent shared information: each document shared generates an instance of *Document* with the set of properties shown. *Document_1*, for example, was stored by John Smith into the concept *RDF* with the annotation "RDF tutorial for beginners ..." with the summary and URI as shown in Figure 1. It also has a set of keywords associated with it. (For simplicity, note that here we show only one keyword *Kw_1*, which is an instance of the class *Keyword*, as is *Kw_2* and furthermore it is not shown that *Keyword* is a subclass of *rdfs#Resource*). The third central class is *Profile*, instances of which represent user information, including the concepts in which they are interested, their names and email addresses. *Profile_1*, for example, is the profile of a user with name "John Smith". Finally, note that keyword *Kw_2* is one of (possibly many) terms (words and phrases) which characterize the concept *Language*. Below we include excerpts from the RDFS and RDF (in XML notation) used to represent the ontology depicted above. We see the declarations of the classes *Document*, *Profile* and *Keyword* in RDFS, followed by the descriptions of *Document_1* and the user profile of *John Smith* in RDF.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ontoshare="http://www.bt.com/ontoshare#">

<!--*************RDFS SCHEMA ************* -->

<rdfs:Class rdf:ID="Document" />
<rdfs:Class rdf:ID="Profile" />
<rdfs:Class rdf:ID="Keyword" />

<!-- Document properties -->
<rdf:Property rdf:ID="submitted_by">
   <rdfs:domain rdf:resource="#Document" />
```

```
  <rdfs:range rdf:resource="#Profile" />
</rdf:Property>
     ... ... ...

<!--  ************ RDF DATA ************  -->

<!--  DOCUMENTS -->
<Document rdf:ID="Document_1">
  <title>RDF Tutorial</title>
  <uri>http://www710.univ-lyon1.fr/champin/
  rdf-tutorial</uri>
  <submitted_by>#Profile_1 </submitted_by>
  <summary>doc summary goes here</summary>
  <isAbout rdf:resource="#RDF" ontoshare:ID="7" />
  <annotation>RDF tutorial for beginners...</annotation>
</Document>

<!--  PROFILES -->
<Profile rdf:ID="Profile_1">
  <user_name>John Smith</user_name>
  <email>john.smith@bt.com</email>
  <interestedIn rdf:resource="#Sesame" ontoshare:ID="5" />
  <interestedIn rdf:resource="#Tools" ontoshare:ID="2" />
</Profile>
     ... ... ...
```

2.3 Retrieving explicit knowledge in OntoShare

In this section, we discuss the ways in which OntoShare facilitates access to and the automatic sharing of the information.

Email notification. As described above, when information is shared in OntoShare, the system checks the profiles of other users in the community of which the user is a member. If the information matches a user's profile sufficiently strongly, an email message is automatically generated and sent to the user concerned, informing the user of the discovery of the information. Thus in cases where a user's profile indicates that they would have a strong interest in information shared, they are immediately and proactively informed about the appearance of the information.

Searching the community store – accessing information and people. Via their OntoShare home page, a user can supply a query in the form of a set of key words and phrases in the way familiar from WWW search engines. OntoShare

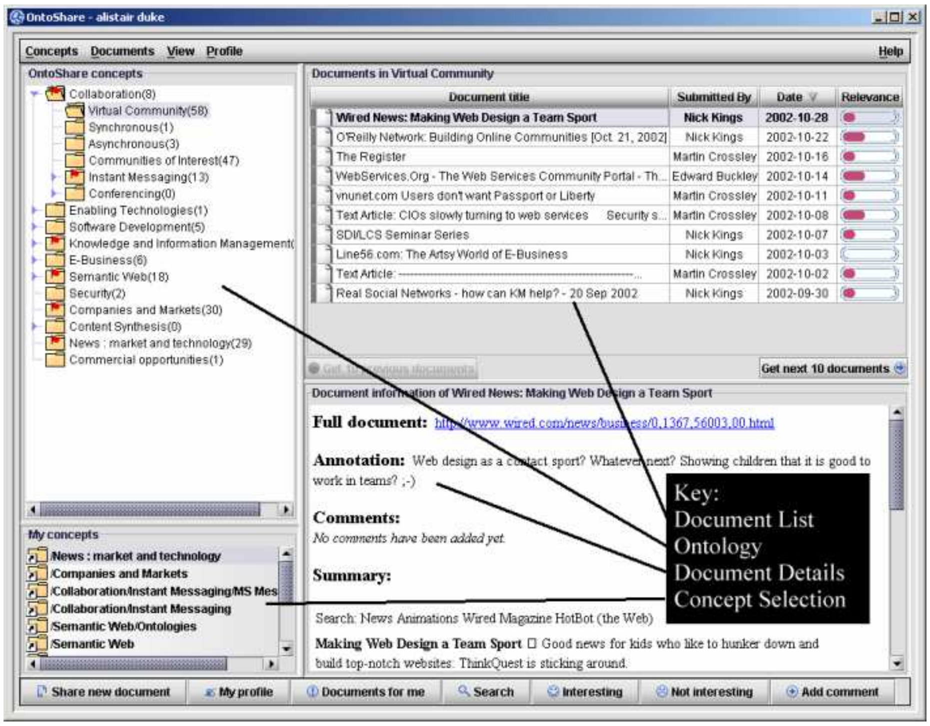

Figure 2: Typical OntoShare Home Page

then retrieves the most closely matching pages held in the OntoShare store, using a vector space matching and scoring algorithm [Har92]. The system then displays a ranked list of links to the pages retrieved and their abridgements, along with the scores of each retrieved page and any annotation made by the original sharer is also shown. Importantly, the user can elect to simultaneously search for other users by selecting the appropriate check box. We will have more to say about this capability to identify other users as well as information in Section 4 when we look at accessing tacit knowledge via other users using OntoShare.

Personalised Information. A user can also ask OntoShare to display “Documents for me” as shown in the top right pane of Figure 2 below. The system then interrogates the OntoShare store and retrieves the most recently stored information. It determines which of these pages best match the user’s profile. The user is then presented with a list of links to the most recently shared information, along with a summary, annotations where provided, date of storage, the sharer and an indication of how well the information matches the user’s profile (the thermometer-style icon in Figure 2).

In addition, 2 buttons are provided (on the button bar at the bottom of the screen in Figure 2) so that the user can indicate interest or disinterest in a particular piece of information – this feedback will be used to modify the user's profile. At this point, the system will match the content of the current document against each concept (ontological class) in the community's ontology. As described above, each ontological class is characterized by a set of terms (keywords and phrases) and the shared information is matched against he term set of each concept using the vector cosine ranking algorithm [Har92]. The system then identifies the set of zero or more concepts that match the information above a given ranking threshold and suggests to the sharer that this set of concepts be added to or removed from their profile in the cases of user interest or disinterest respectively. The user is then free to accept the system recommendation or to modify it by selecting from the set of suggested concepts.

Two further operations are possible on documents presented to the user. These operations are selected from the "Documents" menu. Firstly, a user can add their own annotation to information stored by another user. Secondly, a user can request that OntoShare identifies other users with an interest in the information under consideration.

This "Documents for me" information is in fact displayed on the user's OntoShare home page, so that whenever they access the system, they are shown the latest information. Figure 2 is a typical OntoShare home page.

3 Creating Evolving Ontologies

So far we described how, when a user shares some information, the system will match the content being shared against each concept (class) in the community's ontology. Recall that each ontological class is characterized by a set of terms (keywords and phrases) and that following the matching process, the system suggests to the sharer a set of concepts to which the information could be assigned. The user is then able to accept the system recommendation or to modify it by suggesting alternative concept(s) to which the document should be assigned. It is at this point that an opportunity for ontology evolution arises.

Should the user indeed override the system's recommended classification of the information being shared, the system will attempt to modify the ontology to better reflect the user's conceptualization, as follows. The system will extract the keywords and key phrases from the information using the ViewSum system mentioned above. The set of such words and phrases are then presented to the user as candidate terms to represent the class to which the user has assigned the information. The user is free to select zero or more terms from this list and/or type in words and phrases of his own. The set of terms so identified is then added to the set of terms associated with the given concept, thus modifying its characterization.

We call this approach usage-based ontology evolution and in this way the characterization of a given concept evolves over time, this evolution being based on input from the community of users. We believe that this ability to change as users' own conceptualization of the given domain changes is a powerful feature which allows the system to better model the consensual ontology of the community. Clearly, this level of evolution is limited to changing the semantic characterization of ontological classes and does not support, for example, the automatic suggestion of new classes to be added to the ontology. More advanced ontology evolution is the subject of ongoing research and is described briefly in Section 5. It is also worth noting that we have not concerned ourselves with ontology versioning (tracking and managing changes to an ontology) here. This is an important issue with regard to ontology evolution and the reader is referred to, for example, [SMMS02] for details of work in this area.

As well as usage-based evolution, we have seen above how users also indirectly annotate the information as a side-effect of sharing it with the community and we discuss and motivate this approach below. Pragmatically speaking, it is the case at the time of writing that only a very small proportion of WWW- and intranet-based information resources are annotated with RDF (meta)data. It is therefore beneficial to provide a system wherein such annotation effectively occurs as a side-effect of normal usage. Another important observation is that it is in the general case impossible to cover the content of a document exhaustively by an RDF description. In practice, RDF descriptions can never replace the original document's content: any given RDF description of a set of resources will inevitably give one particular perspective on the information described. Essentially, a metadata description can never be complete since all possible uses for or perspectives on data can never enumerated in advance. Our approach accommodates this observation however in the sense that each community will create its own set of metadata according to its own interest in and perception of information that is added to its store. It is very possible that the same information could be shared in two separate communities and emerge with different metadata annotations in each.

4 Expertise Location and Tacit Knowledge

In Section 2, we focused on the *technical* aspects of OntoShare and on the sharing and storing of explicit knowledge. Explicit knowledge we take to be that knowledge which has been codified in some way. This codification can take place in many different media (paper, WWW page, audio, video, and so on). In the context of OntoShare, by explicit knowledge, we mean the information shared in OntoShare, along with the meta-information associated with it such as the sharer, the annotations attached to it, and so forth. We now turn to the social aspects of the system and tacit knowledge. A large amount of the knowledge within an organization may of course not be codified: it may be personal, context-specific and difficult to write down, and may be better transmitted through a master-apprentice "learning by watching and copying" arrangement. Such knowledge is referred to as tacit knowledge [Pol58]. When tacit knowledge is difficult to make explicit (codify), we need to find new ways of transmitting the knowledge through an organization. Failure to do so can lead to loss of expertise when people leave, failure to benefit from the experience of others, needless duplication of a learning process, and so on.

One way in which a system such as OntoShare can encourage the sharing of tacit knowledge is by using its knowledge of the users within a community of practice to put people who would benefit from sharing their (tacit) knowledge in touch with one another automatically.

One important way we gain new insights into problems is through 'weak ties', or informal contacts with other people [Han97]. Everyone is connected to other people in social networks, made up of stronger or weaker ties. Stronger ties occur between close friends or parts of an organization where contact is maintained constantly. Weak ties are those contacts typified by a 'friend of a friend' contact, where a relationship is more casual. Studies have shown that valuable knowledge is gathered through these weak ties, even over an anonymous medium such as electronic mail and that weak ties are crucial to the flow of knowledge through large organizations. People and projects connected to others through weak ties are more likely to succeed than those not [Han97].

User profiles can be used by the OntoShare system to enable people to find other users with similar interests. The user can request OntoShare to show a list of people with similar interests to themselves. OntoShare then compares their profile with that of every user in the store and a list of names of users whose interests closely match their own. Each name is represented as a hypertext link which when clicked initiates an email message to the named user. Recall that profiles in OntoShare are a set of phrases and thus the vector space model can be used to measure the similarity between two users. A threshold can then be used to determine which users are of sufficient similarity to be deemed to 'match'.

This notion is extended to allow a user to view a set of users who are interested in a given document. OntoShare determines which members of the community 'match' the relevant document above a predetermined threshold figure and presents back to the user a list of user names. As before, these names are presented as hypertext links, allowing the user to initiate an email message to any or all of the users who match the document. In addition, as already mentioned in Section 2.3, a user can carry out a keyword search on other users and thus identify users with an interest in a particular subject. In this way, OntoShare, while not claiming to actually capture tacit knowledge, provides an environment which actively encourages the sharing of tacit knowledge, perhaps by people who previously would not otherwise have been aware of each other's existence.

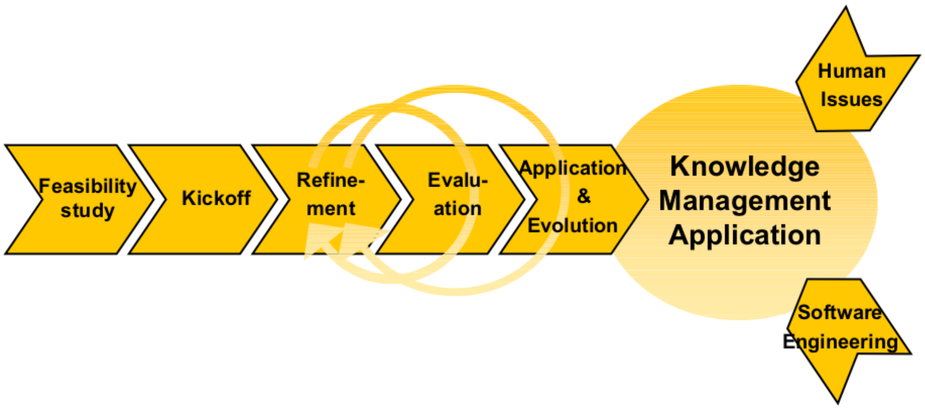

Figure 3: Core Steps of the On-To-Knowledge Methodology

5 Case Study & Evaluation

OntoShare has recently been applied and evaluated using the On-To-Knowledge methodology [SSS04]. Core to the process oriented methodology are "Knowledge Meta Processes" and "Knowledge Processes". While the former support the setting up of ontology based knowledge management applications, the latter one support their usage. A core element of the former that can not be shown in detail here is a lifecycle for ontology engineering. In the following we will (i) briefly sketch main steps of the "Knowledge Meta Process" (cf. Figure 3) and (ii) illustrate the way in which each of the steps were implemented together with the results of the exercise.

5.1 Methodology Feasibility Study

Here the problem and opportunity area and potential solutions are identified. These are then put into a wider organizational perspective. The intention is to select the most promising focus area and target solution for the envisaged application.

Kickoff. In this phase, the actual development of an ontology begins. A requirements specification of the ontology is developed which includes identifying the resources that will contribute to the ontology e.g. existing ontologies that may be re-used & valuable personnel that could contribute to ontology building. The domain and goal of the ontology should also be defined at this stage e.g. the ontology will be used in a knowledge sharing application in the domain of Knowledge Management. Design guidelines should be defined to help users who are not familiar with modeling ontologies. These might include an estimation of the number of concepts and the level of granularity required. The outcome of the kickoff phase should be an ontology requirements specification.

Refinement. This phase involves developing a taxonomy (i.e. modeling is-a relations) and then adding additional relations (other than is-a) to form a more rich ontology. This is generally a cyclical process with knowledge about the domain captured from the domain experts using brainstorming techniques.

Evaluation. There are three forms of evaluation in the methodology. These are technology-focused evaluation, user-focused evaluation and ontology-focused evaluation. (i) Technology-focused evaluation is concerned with evaluating both the properties of the ontology developed and the tools and applications used. Criteria such as language conformance, scalability, performance and interoperability are used. (ii) User-focused evaluation should encompass feedback from users of a prototype tool in the target application environment, usage patterns of the ontology and most importantly whether the ontology based technologies used are at least as good as already existing technologies. (iii) Finally ontology-focused evaluation is concerned with formally evaluating ontologies against a set of rules.

Application & Evolution. The final phase of the methodology is concerned with applying the ontology to its intended domain and managing its evolution. Both centralized and distributed strategies can be employed. As shown in Figure 3, the final three phases of the methodology are cyclical. It is to be expected that improvements to the tools and changes to the ontology will be introduced in an iterative manner.

5.2 Feasibility Study

This involved identifying both an appropriate tool-set and a suitable user group. The user group for the study consisted of approximately 30 researchers, developers and technical marketing professionals from the research and development arm of a large telecommunications firm. The interests of the users fell into 3 main groupings: conferencing, knowledge and information management and personalization technologies. It was felt that three separate yet overlapping topic areas would constitute an interesting mix of interests for the purposes of the trial. The tool-set consisted of OntoShare (which was the only tool to be used by the users), OntoEdit [SAS03] and Sesame [BKv02]. OntoEdit was used to capture the domain ontology produced by the experts. Sesame was used to store this ontology and the OntoShare data store (of added items, annotations, etc.) to provide access for other ontology-based tools. OntoShare allowed the ontology to automatically evolve and extend over the course of the study as documents were added. The effectiveness of this evolutionary process will be considered in the evaluation exercise. An interesting secondary outcome was to look at is the extent to which the ontology built up by the community is useful to other users in other contexts. Work on a searching and browsing facility over the community's information for other users outside the community is described elsewhere [Dv02].

5.3 Kickoff and Refinement

The Kickoff and Refinement stages of the methodology were encompassed by an ontology building process. This majority of this process was carried out at a workshop held at the case study company's premises and run by a knowledge engineer. A selection of 6 key people from the user group were invited to attend the workshop. It was felt that as a whole, they would be able to cover the domain of interest for the whole of the user group. The workshop included a presentation that described in basic terms what an ontology is and how they can be used. It also included a demonstration of the OntoShare tool which introduced the tool to the users as well as showing the use of an Ontology in a practical application. Following this, the ontology building process took place. It was very much brainstorming oriented during in the kickoff [SSS04, SAS03]. The following is a description of the steps that were carried out in the build process.

1. A brief discussion took place on the number of concepts that was felt to be appropriate. Two suggestions of 20 and 50 were made. These were made in the context of the OntoShare tool and how its interface dealt with the Ontology.
2. Each participant was asked to come up with 5+ topics that they felt were important to them. These were then collected. Some early organisation of the hierarchy took place at this stage.
3. Further organisation into a hierarchy then took place. This included adding and removing concepts and moving whole sub-branches of the proposed ontology.
4. The depth of the hierarchy was considered to be too deep. The UI aspects were considered at this stage and a decision taken to restrict the depth to a maximum of three levels where possible.
5. It was remarked that organisational groupings (i.e. knowledge management, conferencing, etc.) had been introduced at the top level of the hierarchy. It

was decided that this might introduce unwanted boundaries for the users and that they might feel that they could only add documents to their particular part of the hierarchy. Also, some of the sub-branches straddled the top-level areas. The top-level was then removed.

6. The number of concepts across each level was considered in terms of the UI. It was felt that 10–15 concepts at each level would be manageable by the user. Concepts of minor importance were then removed or combined at this stage in order to achieve this.

7. The suitability of the overall ontology was then considered and a few additional refinements were made.

The resulting ontology contained 10 top-level concepts and a total of 52 concepts with a maximum depth of three levels.

The group were able to produce this ontology at the workshop which meant that most of the refinement stage (i.e. the organization of concepts and relationships) had been carried out in tandem with the kickoff stage. Complex relationships (i.e. those outside of the simple taxonomical ones) are not included although these automatically develop as data items are added to the OntoShare store and relationships between that data is inferred by OntoShare itself. The over-riding message here is that user groups such as the one in this trial can be expected to produce lightweight ontologies.

5.4 User-Focused Evaluation

Of the three forms of evaluation in the On-To-Knowledge methodology, the most appropriate for use in the case study is user-focused. The tools rely on a high degree of user interaction and as such the users are the best resource for determining whether they meet their objectives. Various user-focused evaluation methods were employed. This section will describe these and the rationale for their use. The objectives of the evaluation were to determine:

- What the users think of sharing knowledge in an environment such as that used in the case study;
- whether the use of an ontology helps with the storing and sharing of knowledge;
- whether the ontology evolution process is effective;
- whether the ontology developed as part of the case study was effective; and
- the good and bad points of the knowledge sharing environment.

The principal means of evaluating the views of the users was with the use of questionnaires. Questionnaires have the benefit of allowing the views of a high proportion of the users to be canvassed without burdening them a great deal.

The questionnaires consisted of a mixture of open questions that required a qualitative response and a series of statements that required a quantitative response indicating the level to which the respondent agreed or disagreed. This mixed approach is endorsed by Eason [Eas88] who states that "Structured questions have the virtue of easy analysis and direct comparability. Their weakness is that they pre-define the answers it is possible to give and may not therefore permit the user to report the most important issues. We have always found it useful to use a structured approach to reveal issues and, once an issue is located, to use an unstructured method to explore the nature of the issue". A 'pre-trial' questionnaire and a post-trial 'questionnaire' were developed.

The 'pre-trial' questionnaire was intended to determine the nature of the users in the case study in terms of the way (and how often) they access, receive and share information. The 'post-trial' questionnaire was intended to extract the user's views of and experiences with the OntoShare system. Particular focus was placed upon the users' views on the usage of an ontology within the tool and the evolution of that ontology.

An additional form of evaluation involved the analysis of usage statistics that were collected by an OntoShare module developed exactly for this purpose. This was able to record every interaction that occurred on the OntoShare server along with the user who performed it. This allows analysis to take place on the use of different OntoShare functions by the group as a whole as well as the behavior of individuals (which can then be cross-referenced with the questionnaire responses). The combination of methods should allow an evaluation of the individual OntoShare functions to be made. The usage of the ontology can also be analyzed by recording the distribution of documents added to the concepts in the ontology and the evolution of the ontology over the course of the case study period.

The final form of evaluation was an expert usability analysis. This involved an assessment of the OntoShare user interface and allowed a thorough inspection to be made by an independent expert. It generally results in a more objective and far reaching analysis than would be the case if it was carried out by someone connected to the development of the tool.

5.5 Application & Evolution

This part of the methodology is concerned with use of the ontology in its intended application and its evolution. It is cyclical with results from the evaluation stage providing impetus for evolution. The delivery of the application to the user group was carefully managed in order to achieve the highest possible level of usage. The actual delivery process is outside the scope of this paper. The evolution of the ontology that was enabled by OntoShare was a topic of the evaluation exercise.

6 Results of Case Study

This section will suggest and discuss a number of recommendations for the evolution of the case study tools and deployment process. A set of lessons learnt from the experience of carrying out the case study is presented. Firstly, recommendations concerned with improving OntoShare are considered. These have come to light as a direct result of the user-focused evaluation exercise.

Give careful consideration to the nature of the virtual community. The evaluation showed that the majority of users were passive in their use of OntoShare i.e. they were happy to receive e-mails and read documents but did not add items to the system. As a result, the experience of the majority of OntoShare users is determined by the actions of the few who actually add items to the system. If those few users did not exist, the knowledge sharing benefits would not be forthcoming. Depending on the local organizational culture, dependence on a relatively small proportion of the user community may or may not be appropriate. Many successful communities are of this nature but in some cases alternative strategies may be required to reduce the dependence upon active users. In their responses to questionnaires, users made some useful suggestions in this regard. One user made the suggestion that OntoShare needed to be "regularly seeded with potentially relevant information or gain critical mass usage to offer a positive benefit and justify the effort of maintaining profiles and entering articles or information". This could be carried out manually by a knowledge engineer or automatically by an information agent and could benefit the system by ensuring sufficient data was added that was of interest to a wider cross section of the users. Both methods have drawbacks in that the manual process is time-consuming and the automatic method introduces the risk of downgrading the quality of information that is shared. The recommendation is to experiment with a combination of user, knowledge engineer and agent added data that can be varied depending upon user input and the nature of the community.

Provide better interface access. A commonly occurring criticism of the system was its use of a Java applet to provide the interface. This proved to be slow to load and required a login step in order that the user could be recognized. It was originally chosen over a HTML-based interface because displaying the Ontology (with a collapsing folder structure) would have proved difficult in that format. An alternative would be to provide an application, however this needs to be installed and re-installed every time a change is required which would have been disruptive for a system in its infancy. Also, different versions are required for different operating systems. The Java applet was so unpopular that using one of the alternatives now seems more attractive. The use of JavaScript and DHTML should be considered to allow the interface to operate in a standard browser. If this is not appropriate then the use of an installed application would probably be the best course of action.

Provide wider access to functions. Another drawback with the system that was widely mentioned was the need to login to the system to provide comments, add items, etc. Alternative methods of accessing individual functions of OntoShare should be explored. These might include direct links in notification e-mails to a comment adding facility and support in a web browser for dropping a URL into the system. The intention should be to reduce the effort that is required to use each function.

Richer ontological representation. The OntoShare ontology currently contains concept relationships of the topic–subtopic type. Other relations are inferred by OntoShare and the user can request to see these. A better way of presenting these to the user is required. This might be a toggle where the relationships are indicated on screen when it is turned on.

Provide better support to new users. When users first login they are often daunted by the interface. Better support should be provided to help them set up their profile and gain familiarity with the available functions. This could be in the form of a 'splash screen' that is shown on the first use of the tool or that can be disabled once users become familiar. This would show tips on usage and a short description of each function.

Inform users about an ontology change. When changes to the ontology are accepted or rejected by the users, they should be notified of the outcome and invited to adjust their profiles accordingly. This will ensure that users can gain access to items added to new areas without having to login and browse the concepts. The following lessons in relation to the development of ontologies have been learnt.

Physical presence is required. The approach used to produce a domain ontology i.e. a group of experts in a focused workshop led by a knowledge engineer who is physically present, proved to be fruitful. The domain experts have limited time available, hence it is necessary to be very focused. Had this capture been carried out over a period of time involving a number of disparate people it would have probably been a drawn out process, lacking in focus.

Domain experts can be expected to produce a taxonomy. A simple ontology in the form of a taxonomy is the most likely outcome from a group of domain experts asked to contribute in this way. More complex ontologies require considerably more effort. The OntoShare system is suited to a simple ontology with topic–subtopic relations.

The On-To-Knowledge methodology provides an effective framework for the introduction of an ontology-based application. The application of the methodology to the case study resulted in the rapid development of an ontology that performed well in its intended application. Users reported that they found it to be an appropriate ontology for their domains despite the limited amount of time that was available for ontology development.

In terms of the evaluation objectives introduced in Section 5.3, the following can be stated:

- Users are generally happy to receive shared knowledge from OntoShare and will often read it if it appears interesting and they have enough time. Only a minority of users actively share documents in OntoShare (mainly due to time pressures) so steps should be taken to make it as easy as possible to share. User shared data should be augmented with data added by a Knowledge Engineer or an information agent.
- Users reported that they found it useful to have the ontology available both when browsing for items and when adding items to the system.
- The ontology evolution process proved to be effective. Users were happy to create new concepts and these were accepted by the other users. Users reported that they found this facility useful although they were less inclined to make use of the interesting – not interesting features to evolve individual concept characterizations. Better access to facilities such as these has been recommended in order to further their usage.
- The ontology that was developed proved to be effective in the case study. It performed well in its intended application – measured by the distribution of documents that were added to its concepts. Users reported that they found it to be an appropriate ontology for their domains of interest.
- A number of good and bad points were identified which have been discussed. These have led to a set of recommended improvements being made.
- The On-To-Knowledge methodology proved to be an effective framework for the introduction of an ontology-based application. The application of the methodology to the case study resulted in the rapid development of an ontology that performed well in its intended application. Users reported that they found it to be an appropriate ontology for their domains of interest despite the limited amount of time that was available for ontology development. The results of this evaluation will be used in the continued development and exploitation of OntoShare and related tools.

7 Further and Related Work

Research and development of OntoShare is ongoing. A particular area of focus currently is the ontological structure: a strict hierarchy (taxonomy) of concepts about which the communities wants to represent and reason may prove ultimately limiting and various possibilities for allowing a more expressive concept map are under consideration. One such is that OntoShare will be developed beyond the subtopic/supertopic concept hierarchy with *IsRelatedTo* properties, allowing "horizontal" links between concepts. The exploitation of this additional information is again matter for further research. One proposal is that when seeking to match users to other users, the system can use some notion of tree-matching, taking into account the concepts in the users' profiles as well as not only the subtopic links but also the *IsRelatedTo* links. These richer ontologies may be better represented in a more expressive language such as OWL, the W3C recommendation [SWM04]. A further research area is the automatic identification and incorporation of new concepts as they emerge in the community. Work on this is however at a very early stage and is beyond the scope of this paper.

Turning to related work, Staab et al. [SAD$^+$00] describe a system for building and maintaining community web portals. As with OntoShare, a ontology-based is taken and an ontology is used to structure and access information, using F-Logic as its underlying language for ontology representation and querying. Relatively sophisticated querying is supported, offering a degree of inferencing in the query engine not offered in OntoShare. Semi-structured information provision is supported by the use of wrappers. User profiling and automatic alerting are not supported, neither is the ability to change the semantic characterization of a class as in OntoShare.

RiboWeb [ABC$^+$99] is another example of an ontology-based community portal RiboWeb holds information about ribosome data and computational models for the processing thereof. Most data are scientific papers manually linked to the appropriate ontological categories. Knowledge engineers maintain the data and metadata, rather than the data being provided by the community itself as in OntoShare.

8 Concluding Remarks

We have described OntoShare, an ontology-based system for sharing information among users in a virtual community of practice. We motivated the use of Semantic Web technology for KM tools and described how ontologies in OntoShare are defined in RDFS. Communities are able to automatically share information and create RDF-annotated information resources as a side-effect of this activity. Furthermore, these information resources are then of course available to other RDF-based tools for processing: the community semi-automatically creates an ontology-based annotated information resource for use by itself and others.

Importantly, the ontology used by a given community in OntoShare can change over time based on the concepts represented and the information that users choose to associate with particular concepts. This is a significant advantage over a community attempting to reach consensus on a set of concepts and how they relate to another at the outset that is then difficult or impossible to change. Much remains to be done in this area however, particularly with regard to the introduction of new concepts. In addition, users have personal profiles according to the concepts in which they have declared an interest and these profiles also evolve automatically, seeking to match more closely a user's information needs and interests based on the usage they make of the system.

The use of OntoShare in a case study and its subsequent evaluation has been described. The results of the evaluation were presented which have led to a number of recommendations for the continued development of OntoShare. We indicated some further directions of research. OntoShare exemplifies the much-improved knowledge management tools that the advent of the Semantic Web and its support for ontologies makes possible.

Acknowledgements

The research described was partly funded by the EU IST-1999-10132 project On-To-Knowledge (see `http://www.ontoknowledge.org`). We thank all our partners for their fruitful cooperations, discussions and the fun we had during the project.

References

[AB95] K. Ahmed and M. Benbrahim. Text summarisation: the role of lexical cohesion analysis. *New Review of Document and Text Management*, 1:321–335, 1995.

[ABC+99] R. Altmann, M. Bada, X. Chai, M.W. Carillo, R. Chen, and N. Abernethy. RiboWeb: an ontology-based system for collaborative molecular biology. *IEEE Intelligent Systems*, 15(5), 1999.

[BKv02] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF Schema. In Horrocks and Hendler [HH02], pages 54–68.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 2001(5), 2001. available at http://www.sciam.com/2001/0501issue/0501berners-lee.html.

[Dav00] N.J. Davies. *Supporting Virtual Communities of Practice*, chapter 3, pages 199–212. Springer-Verlag, 2000.

[Dv02] A. Duke and J. van der Meer. User profile construction. On-To-Knowledge deliverable 14, BT and AIdministrator Nederland b.v, 2002.

[Eas88] K. Eason. *Information technology and Organisational Change.* Taylor & Francis, London, 1988.

[Fen01] D. Fensel. *Ontologies: Silver bullet for knowledge management and electronic commerce.* Springer-Verlag, Berlin, 2001.

[GPB02] A. Gómez-Pérez and V. R. Benjamins, editors. *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web (EKAW 2002)*, volume 2473 of *Lecture Notes in Artificial Intelligence (LNAI)*, Siguenza, Spain, 2002. Springer.

[Han97] M.T. Hansen. The search-transfer problem: The role of weak ties in sharing knowledge across organisation subunits, 1997. Working Paper, Harvard Business School.

[Har92] D. Harman. *Ranking Algorithms.* Prentice-Hall, NJ, USA, 1992.

[HH02] I. Horrocks and J. A. Hendler, editors. *Proceedings of the First International Semantic Web Conference: The Semantic Web (ISWC 2002)*, volume 2342 of *Lecture Notes in Computer Science (LNCS)*, Sardinia, Italy, 2002. Springer.

[LS99] O. Lassila and R. Swick. Resource Description Framework (RDF). Model and Syntax Specification. W3C Recommendation 22 February 1999, 1999. available at http://www.w3.org/TR/REC-rdf-syntax.

[Pol58] M. Polanyi. *The Tacit Dimension.* Doubleday & Co., Garden City, NY, 1958.

[SAD+00] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Ontology-based community web portals. In *Proceedings of the AAAI-2000.* MIT Press, Menlo Park, USA, 2000.

[SAS03] Y. Sure, J. Angele, and S. Staab. OntoEdit: Multifaceted inferencing for ontology engineering. *Journal on Data Semantics*, LNCS(2800):128–152, 2003.

[SBD91] J. Seely-Brown and P. Duguid. Organisational learning and communities of practice. *Organisational Science*, 2(1), 1991.

[SMMS02] L. Stojanovic, A. Maedche, B. Motik, and N. Stojanovic. User-driven ontology evolution management. In Gómez-Pérez and Benjamins [GPB02], pages 285–300.

[SS04] S. Staab and R. Studer, editors. *Handbook on Ontologies in Information Systems.* International Handbooks on Information Systems. Springer, 2004.

[SSS04] Y. Sure, S. Staab, and R. Studer. *On-To-Knowledge Methodology.* In Staab and Studer [SS04], 2004.

[SWM04] M. K. Smith, C. Welty, and D. McGuinness. OWL Web Ontology Language Guide, 2004. W3C Recommendation 10 February 2004, available at http://www.w3.org/TR/owl-guide/.