

Supporting Ontology Driven Document Enrichment within Communities of Practice

John Domingue, Enrico Motta, Simon Buckingham Shum, Maria Vargas-Vera, Yannis Kalfoglou

Knowledge Media Institute

The Open University

Walton Hall, Milton Keynes, MK7 6AA, UK

+44 1908 655014

{j.b.domingue; e.motta; s.buckingham.shum; m.vargas-vera; y.kalfoglou}@open.ac.uk

ABSTRACT

Formative work by Lave and Wenger has articulated how *practices* emerge through the interplay of informal processes with symbolic codifications and artifacts. In this paper, we describe how ontologies can serve as symbolic tools within a community of practice supporting communication and knowledge sharing. We show that when a community's perspective on an issue is stable, it opens the possibility for introducing knowledge services, based on an ontology co-constructed by knowledge engineers with stakeholders. Using a case study we describe our approach, ontology driven document enrichment, looking at how ontology construction and population can be supported by web based technologies.

Keywords

Ontology, Semantic Web, Communities of Practice, Knowledge Management.

INTRODUCTION

Formative work by Lave and Wenger [14, 23] has articulated the nature of the *practices* from which the term *community of practice* derives its name. Practices emerge through the interplay of informal processes with symbolic codifications and artifacts:

...Such a concept of practice includes both the explicit and the tacit. It includes what is said and what is left unsaid; what is represented and what is assumed. It includes language, tools, documents, images, symbols, well-defined roles, specified criteria, codified procedures, regulations, and contracts that various practices make explicit for a variety of purposes. But it also includes all the implicit relations, tacit conventions, subtle cues, untold rules of thumb, recognizable intuitions, specific perceptions, well-tuned sensitivities,

embodied understandings, underlying assumptions, and shared world views. Most of these may never be articulated, yet they are unmistakable signs of membership in communities of practice and are crucial to the success of their enterprise. ([23], p. 47)

In this paper, we describe how ontologies [9] can serve as symbolic tools within a community of practice. We show that when a community's perspective on an issue is stable (i.e. there is reasonable consensus), it opens the possibility for introducing knowledge services, based on an ontology co-constructed by knowledge engineers with stakeholders. The ontology reflects a "shared world view", codifying "well-defined roles", "specified criteria" and "codified procedures." Throughout, we regard representations such as ontologies as *boundary objects* [2] whose role is to support communication and negotiation over meaning between stakeholders within and across communities of practice.

Once an ontology has been constructed a population phase uses the ontology to describe web documents from a communal viewpoint. Two key questions which arise in this type of enterprise and that we address in this paper are: who develops the ontology? and how is the ontology population phase supported?

We believe that knowledge engineers are crucial in the ontology development phase. The main reason for this choice is that a careful design of the ontology is crucial to ensure the success of any particular document enrichment initiative. The ontology specifies the selected communal viewpoint, circumscribes the range of phenomena we want to deal with and defines the terminology used to acquire domain knowledge. In our experience small errors/inconsistencies in any of these aspects can make the difference between success and failure. Moreover, ontology design requires specialist skills which are normally not possessed by the members of our target user communities.

Our approach is to develop the ontology using a participatory design methodology. The ontology is developed during a series of face-to-face meetings between

knowledge engineers, who are concerned with issues such as representational consistency and completeness, and a representative group of the target community.

In contrast it is essential that ontological enrichment occurs without the aid of knowledge engineers. Unless enriched web resources are a “living archive” the resultant services will soon fall into disuse. In describing the APECKS personal ontology server Tennison and Shadbolt [21] make a case for “living ontologies”.

In the rest of this paper we shall illustrate our approach, which we term ontology driven document enrichment [17], using a case study. We start by outlining the domain, the architecture of the application and one of the knowledge services that we created. We then describe the design of the ontology and four ways in which we support the ontology population process. Related work is briefly summarized before ending with some conclusions.

CASE STUDY AN OBSERVATORY ON LIFELONG LEARNING INITIATIVES

Case Study Background

In its Green Paper, ‘The Learning Age’, the UK Government set out its vision of ‘a learning society in which everyone, from whatever background, routinely expects to learn and upgrade their skills throughout life.’ To fulfill this vision the UK Government set up the University for Industry (Ufi) in the Autumn of 2000.

Working as a public-private partnership, Ufi provides flexible learning packages to improve the quality of life of individuals and to boost business competitiveness.

Promoting and facilitating lifelong learning is a non-trivial matter. As part of several support measures the Marchmont project (www.marchmont.ac.uk) was setup to create a *National Observatory* for the Ufi facilitating the dissemination of best practice on lifelong learning to researchers and policy makers.

Architecture

The overall architecture of the system is shown in Figure 1. At the centre of the architecture is a knowledge server, the components of which are as follows:

- *LispWeb* - a customised HTTP server [19] which offers a library of high-level Lisp functions to dynamically generate HTML pages.
- *WebOnto Server* - WebOnto [3], composed of a central server and a Java based client, enables knowledge engineers to collaboratively browse and edit knowledge models over the web.
- *OCML* - An operational knowledge modelling language [16], which provides the underlying representation for our ontologies and knowledge models.
- *Observatory Library* – the 13 knowledge models which represent the observatory domain. The library is used

to index the summaries of lifelong learning experiences contained in the good practice database.

Connected to the central server are:

- *The Good Practice Database* – a database containing several hundred summaries of documented examples of lifelong learning.
- *Named Entity Recognizer* – this uses the Marmot and Badger systems from Riloff [18] in combination with a regular expression matcher to support the automatic creation of OCML entities from text in web pages.
- *WebOnto Client* – a Java based client to the WebOnto server.
- *Observatory Services* – a set of knowledge services based on the observatory library. These services support the sharing of knowledge on how to promote and support lifelong learning.

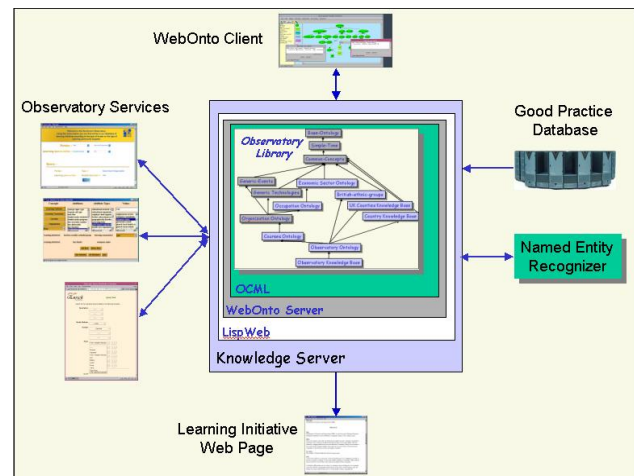


Figure 1. The architecture of the Observatory.

In contrast with other approaches to semantic annotation (e.g. SHOE project [11]) we decouple the knowledge structures from the web resources. This architecture allows us to provide multiple knowledge services, possibly for different communities of practice, over the same set of web documents. For example, a community of graphic designers may be interested in the typography and layout of a set of web pages whereas experienced website developers may be interested in the structure of the underlying HTML code. An additional benefit of this architecture is that changes to the ontology are immediately available to the knowledge services.

An Observatory Service

The Observatory services are designed to be easy-to-use by non-IT specialists and to provide answers to policy level questions. Figure 2 shows a screen snapshot of a web interface, constructed in Flash™, for finding learning initiatives according to the type of funder or the characteristics of the targeted learning community. In the

figure the user is asking for a government funded learning initiative which involved a socially excluded community.



Figure 2. A screen snapshot showing the query interface asking for a government funded learning initiative which involved a socially excluded learning community.

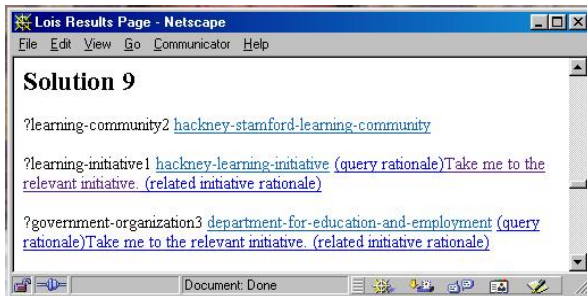


Figure 3. A screen snapshot showing the results of the query in figure 2.

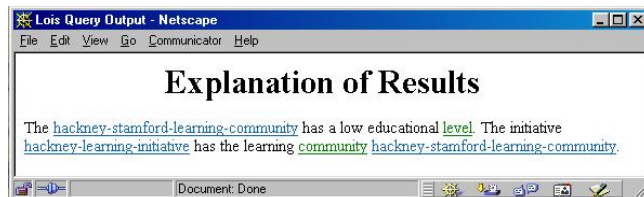


Figure 4. The explanation generated for the query formulated in figure 2.

The query is run in OCML on the knowledge server. A set of rules link OCML knowledge items to relevant learning initiatives within the good practice database. Figure 3 shows the 9th (of 11) solutions. Each solution contains links to a knowledge item, a related learning initiative and links to an explanation of why they were returned. The explanation, shown in figure 4, describes why the target learning community, the members of the Stamford housing estate, were considered to be socially excluded. A screen snapshot showing the Hackney Learning Initiative referred to here is shown in figure 13.

Ontology Design

There were several constraints which had to be satisfied when creating the observatory ontology. The ontology,

obviously, had to characterise the domain such that a) the types of questions posed by policy makers and researchers could be answered and b) there was a mapping to the existing database of learning initiatives. The ontology was constructed by a number of experienced knowledge engineers within our lab. The population of the ontology was to be carried out by a number of project officers within a distance learning group at The Open University who did not have a computing background. It was important that all of the 'observatory team' understood and had ownership of the ontology. Also as outlined in [5] in their analysis of the KA² initiative, and in [12] in their description of a SHOE case study, ontology development and representing specific resources are intertwined activities.

The conceptual design of the ontology was developed in a series of weekly meetings involving the whole observatory team. A number of the meetings included external policy makers and lifelong learning researchers who would be end users of the observatory. Once an initial version of the ontology had been implemented in WebOnto, by experienced knowledge engineers, a sample population phase followed. In the early part of this phase the knowledge engineers and populators collaboratively coded 10 practices in the database. Coding difficulties would either result in immediate changes to the ontology or be logged and changed later. The populators then coded a further 20 practices on their own reporting problems by phone or email. Additionally, the team continued to meet face-to-face weekly to discuss problems and changes to the ontology. These discussions would invariably result in changes to the ontology and occasionally in the addition of new tools. WebOnto's architecture meant that any changes to the ontology (or to WebOnto itself) were immediately available to the populators.

Because the domain, the intersection of learning and social policy, was relatively broad we created and reused a number of higher level ontologies. Figure 5 shows the structure of the relevant portion of our library. The knowledge base currently indexes several hundred good practice case studies.

The core of the ontology is based on a learning initiative class which represents a single documented case in the good practice database. As we can see from table 1 the main attributes of learning initiatives are the title, location, date, learning rationale, funders, organisations involved, target learners and the tangible results. Most of these types of objects that can fill these slots are fairly obvious. Often the descriptions of learning initiatives describe generic rather than specific entities. For example, involved parties are sometimes described using phrases such as "a local college" or "a few mechanical engineering SMEs". A generic-organisation class is used to capture these types of statements.

Slot Name	Documentation	Slot Attributes
Has-title	The title of the initiative.	This is a string.
Has-location	The location of the initiative. This includes information on the social geography of the area.	The value of the slot must be of type learning location.
Has-initiative-date	The starting date for the initiative.	This value is an integer representing the year (used within the existing database).
Has-rationale	The underlying rationale for the initiative.	The value of type rationale-for-learning.
Has-funder	The funding organisation or person.	The value is of type organisation or person.
Other-involved-parties	Organisations, individual people and communities which take part in the initiative.	The value is either of type organisation, generic-organisation, person or community.
Has-learner	The target audience for the initiative.	The value is of type learning-community.
Has-deliverable	The tangible results of the initiative.	A document, technology or organisation.

Table 1. The definition of the learning-initiative class.

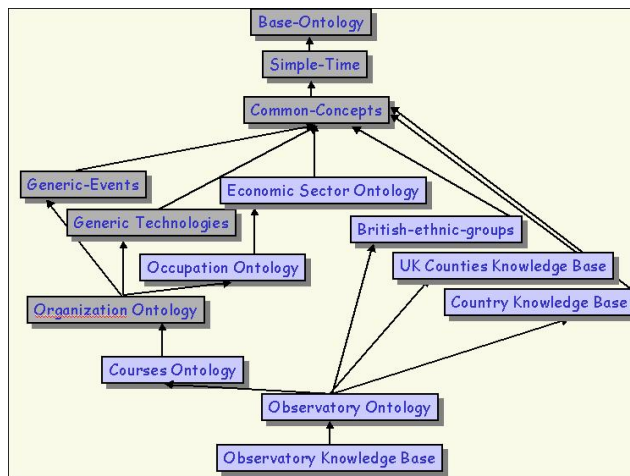


Figure 5. The ontologies and knowledge bases used by the observatory ontology. The darkly shaded knowledge models pre-existed the project.

The other key definition within the ontology is the learner-community class. We do not have space here to include this definition but the key attributes include the affiliation, ethnic group, occupation, gender, age, skill level and dependents. This broad range of slots reflects the diverse attributes that learning and social policy researchers argue can affect access to learning within a community.

Ontology Population

Although WebOnto is primarily aimed at expert model builders we have recently provided a number of tools to allow non-experts to populate ontologies. Integrating support for ontology creation and population within WebOnto contrasts with the approach taken in tools such as Protégé [8] where ontology construction and population are separated.

Help in WebOnto is provided in four main ways:

- *Multiple visualizations* – aid in reviewing what has been created.

- *Automatically generated instance forms* – support the addition of instances.
- *Knowledge items from web pages* – information extraction techniques have been coupled with direct manipulation techniques to enable OCML entities to be created from web pages.
- *Automatic type checking* – automatically checking for undefined values and constraint violations.

Multiple Visualizations

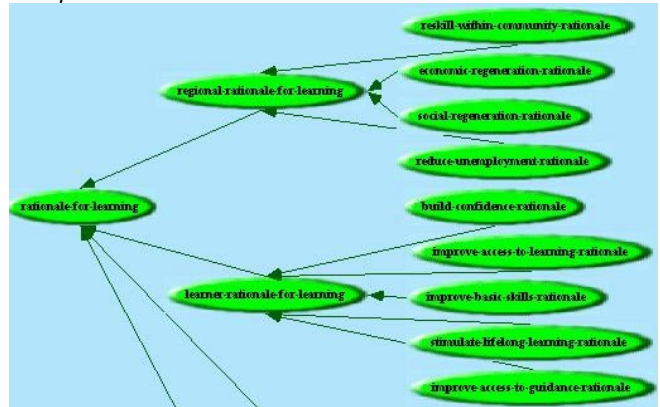


Figure 6. A screen snapshot of WebOnto showing part of the rationale-for-learning class hierarchy.

The use of visualizations has long been acknowledged to be important in the creation of knowledge models [4]. The key is to provide support for high level or coarse grained views which are tightly coupled to multiple fine grained views. This became a design requirements for WebOnto. Figures 6 and 7 show a coarse grained and fine grained visualization of the rationale-for-learning class.

A significant task where visualizations can aid populators is in validation. Populators need easy-to-read detailed descriptions of the added knowledge structures. Often the ontological enrichment of a web resource is based on a single class or on one of a set of related classes - typically

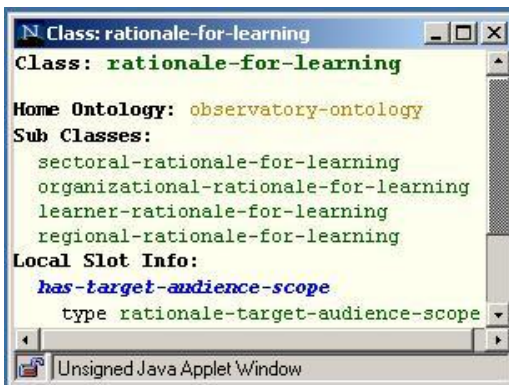


Figure 7. A screen snapshot of the fine grained view of the rationale-for-learning class. Colour coding is used to differentiate between the types of items displayed (e.g. class are coloured green, relations are coloured blue). Each OCML item is mouse sensitive and can be selected and inspected.



Figure 8. above shows a screen snapshot of a connected instance based visualization. Items in bold were defined by the ontology populators. Colour coding distinguishes between instances, classes and relations. Individual instances are picked out with background shading (enhanced for this paper).

class A constrains the type of a slot in class B. This heuristic provides the basis for the design of a connected instances visualization. This view displays all the instances connected to a selected instance. Figure 8 shows a connected instances view of the hackney-learning-initiative. Within this view instance names are shown in black, classes in green and slot names in a light blue. Knowledge items which were entered by the user are shown

in bold. Any slot values which are instances are expanded. Each instance is picked out using background shading.

Within figure 8 we can see that the hackney-learning-initiative is an instance of learning-initiative. The has-location slot has the value hackney-li-location which is an instance of learning-related-location. The has-premises-type slot of hackney-li-location has two values - the classes community-centre-premises and library-premises. The department-for-education-and-employment instance was created by the user but the values of its slots were not. The depth of the inline expansion is defined by the user. Selecting any instance in the view creates a new connected instances view. We elected to provide these visualizations in HTML format so that they could easily be printed and viewed in hardcopy format – a requirement from the ontology population community.

Automatically Generated Instance Forms

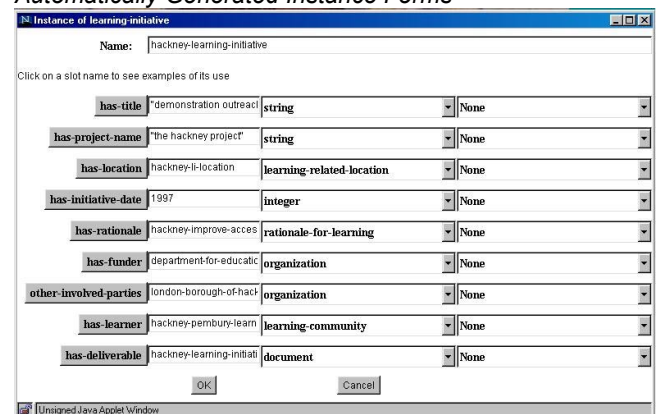


Figure 9. A screen snapshot showing an automatically generated learning-community instance edit form.

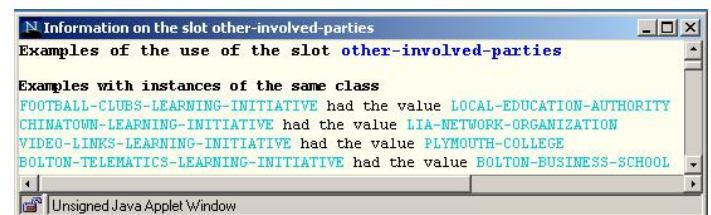


Figure 10. A screen snapshot of the help given when selecting the other-involved-parties button of the form shown in figure 9.

Many errors in semantic annotation occur because of errors in naming existing entities and in selecting the class of new instances [5]. The forms in WebOnto seek to alleviate this by prompting users with the names of relevant knowledge items.

An example of an automatically generated form for editing an instance of a learning community is shown in figure 9. Each slot is displayed as a row. The slot name is a button which displays examples of the values that have been given to the slot has other instances. Figure 10 shows the result of selecting the 'other-involved-parties' button.

The second column is a simple text field into which the name of a value can be entered. Within our underlying knowledge modelling language OCML [16] slots can be typed using a class or a combination of classes (e.g. (or organization person)). These classes and all of their descendants appear in alphabetical order the third column of the form. Figure 11 shows a user selecting the training-organization class for the other-involved-parties slot. When a class is selected in the instances of the class appear in the menu in the fourth column. Figure 12 shows a user selecting the focus-central-london instance.

Figure 11. A screen snapshot showing a user selecting the training-organization class for the other-involved-parties slot of a learning-initiative instance.

Figure 12. A screen snapshot showing a user selecting the focus-central-london instance for the other-involved-parties slot of a learning-initiative instance.

The forms here are in some respects similar to the forms provided in Protégé-II [8]. The key difference is that instance forms in WebOnto are generated directly from the ontology whereas the forms in Protégé-II use an extra set of form specific definitions. The extra information means that the generated forms can use non-trivial layouts but require an extra compilation cycle. Within WebOnto any changes to the ontology are immediately reflected within the forms.

Knowledge Items from Web Pages

As with the majority of our application domains a proportion of the elements referred to in the observatory knowledge base appear within web documents, specifically, within the entries within the good practice database. To aid in the generation of knowledge items from web documents WebOnto contains an interface to a named entity recognizer. Figure 13 shows the interface in use. Using a Netscape™ web browser the user specifies the source web document and the knowledge base to be used. The document is then displayed as usual alongside a WebOnto client.

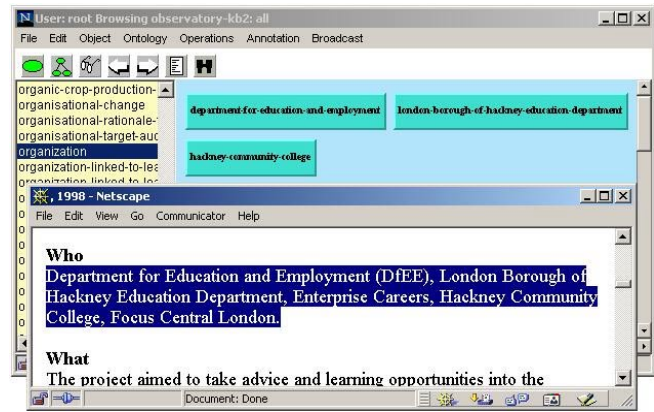


Figure 13. A screen snapshot showing the creation of three new instances of organizations from a highlighted segment of text within the Hackney Learning Initiative document.

In figure 13 the user has selected the organization class, within the WebOnto client, and using a menu selected a 'Create Instances' option. She then highlights the text within the 'Who' paragraph in the Hackney Learning Initiative displayed in the Netscape browser. This text is passed via the knowledge server to the entity recognizer. Within the entity recognizer, Marmot [18] tokenizes the text identifying nouns. The results of this analysis are then passed to Badger [18] and a regular expression matcher (written in Perl) in parallel to identify named entities. We use a regular expression matcher because Badger relies on the input text being composed of grammatical sentences (nouns, verbs and prepositions). Once the text analysis has been completed the entity recognizer passes back to the knowledge server the three segments of recognized text with corresponding entity types. These three segments are used to create the three new instances shown in figure 13.

The interface between OCML and the entity recognizer is implemented with two types of constructs: pattern definers and templates. A pattern definition consists of the name of an OCML class or instance and a set of strings which represents patterns using the using the standard notation for regular expressions. The pattern for a college is:

```
(def-pattern college
  "(capital_word)* College"
  "(capital_word)* College of (capital_word)*")
```

Within the observatory case we have created patterns to identify organizations, ethnic groups, peoples' names and dates.

Templates are used to create new OCML structures from the results of the entity recognizer. Currently three types of template are used:

- *New class instance* – this specifies how text can be used to create a new instance of a class.
- *New class subclass* – this specifies how subclasses of a class can be created.

- *Fill instance* – specifies how an existing instance is filled.

A template consists of the name of a class or instance, a list of variables and the template body. Within the template body variables are denoted by the prefix '\$', and, \$class-name and \$instance-name are special variables which represent the name of the class and instance respectively. The template used to create the `hackney-community-college` instance was:

```
(def-new-instance-template organization (name)
  (def-instance $name $class-name))
```

Other examples of how we have combined our knowledge modelling infrastructure with information extraction technologies can be found in [22].

Automatic type checking

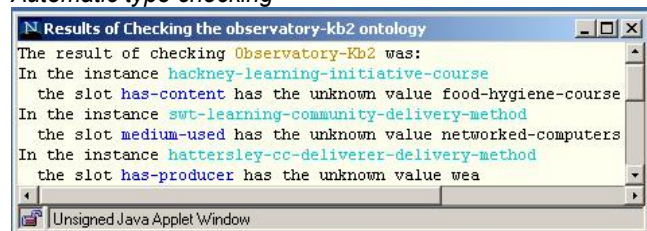


Figure 14. The result of carrying out consistency checking on the observatory knowledge base. Items within the knoweldge base are highlighted using colour and can be selected and inspected. Colour is used to distinguish between instances and relations.

The late 80s and 90s saw a considerable effort into creating tools for validating and verifying knowledge bases [15]. We have found that even relatively simple tools can aid ontology populators. Within WebOnto we have a consistency checker which reports on slot values which have not been defined. OCML also contains a real-time constraint checker. The output of checking the observatory knowledge base is shown in figure 14. Any of the instances or relations shown in figure 14 can be inspected by simply clicking on them.

RELATED WORK

The SHOE project [11] has proposed an extension to HTML to allow the specification of ontological information. The project team has also developed an editor to support the page annotation process. This work is mainly at the infrastructure level. That is, they suggest a mechanism to allow the representation of information and provide tools to edit and retrieve it, whereas we take a holistic approach to document enrichment looking at the wider issues concerning usability and sustainability.

The KA² initiative [1] shares a number of commonalities with our work. As with the case described here the aim of KA² is to allow a community to build a knowledge base collectively, by populating a shared ontology. Similarly to the approach used in SHOE the knowledge base is constructed by annotating web pages with special tags,

which can be read by a specialised search engine cum interpreter, Ontobroker [6]. In this paper we have described an approach which learns from the early problems reported in that initiative [5].

A number of tools such as the CEDAR toolkit [10] and OntoAnnotate [20] provide support based on a web browser integrated with a view of an ontology. The CEDAR annotation tool allows segments of text from web pages to be associated with OCML structures stored on a WebOnto server.

Within OntoAnnotate text can be selected from a web page and dragged to fill in the value of an instance. OntoAnnotate also contains mechanisms for managing annotations after an ontology is altered, a text pattern matcher similar to the one described here and links to an ontology based information extraction system. Both the CEDAR annotation tool and OntoAnnotate are designed to use ontologies to annotate web pages whereas goal of the technologies described here are to facilitate the population of ontologies. Hence, rather than create a separate tool we elected to extend WebOnto thus tightly coupling the ontology development and resource description activities.

In terms of the underlying architecture, as we stated earlier the main difference between our approach and the above approaches to adding semantic information to web pages is that we decouple the web pages from the knowledge model. We should state however that the WebOnto server is now able to export knowledge models in OIL RDF syntax [7]. This facility was used to incorporate parts of our library into an OIL based ontology server as part of a dynamic link service (see [13] for more details).

CONCLUSIONS

In this paper we have described how ontologies can support knowledge sharing within communities of practice. To be successful it is important that all stakeholders are able to participate in the ontology development process and that this process is ongoing and integrated with ontology population. Moreover, ontology population requires support from a mixture of technologies and as far as possible should be integrated into existing working practices.

We have now been using this approach over a number of years in a range of projects, in domains ranging from managing best practice in the aerospace industry, to supporting the application of medical guidelines. Our experience to date suggests that our approach appears to provide both the technology and the methodological framework required to minimize risk and ensure the participating community's acceptance.

ACKNOWLEDGMENTS

This work was funded by the Marchmont Project under the Adapt Programme and the Advance Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK EPSRC Council under grant number GR/N15764/01.

REFERENCES

1. Benjamins, R., Fensel, D. and Gomez Perez A. Knowledge Management through Ontologies. In U. Reimer (editor), *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management*,. Basel, Switzerland, 29-30 October, 1998.
2. Bowker, G. C., and Star, S. L. *Sorting Things Out: Classification and its Consequences*. MIT Press: Cambridge, Mass., 1999.
3. Domingue, J. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In B. Gaines and M. Musen (editors), *Proc 11th Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, April, 1998.
4. Eisenstadt, M., Domingue, J., Rajan, T. and Motta, E. Visual Knowledge Engineering. *IEEE Transactions on Software Engineering* Special Issue on Visual Programming, 16(10), 1164-1177, October, 1990.
5. Erdmann, M., Maedche, A., Schnurr, H.P., Staab, S. From Manual to Semi-Automatic Semantic Annotation: About Ontology-based Text Annotation Tools. *COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, Centre Universitaire, Luxembourg, 5-6 August, 2000
6. Fensel, D. Decker, S., Erdmann, M. and Studer, R. Ontobroker: The very high idea. *Proc 11th Annual Florida Artificial Intelligence Research Symposium (FLAIRS-98)*.
7. Fensel, D., Horrocks, I., van Harmelen, F., Decker, S., Erdmann, M. and Klein, M. OIL in a Nutshell, *Proc. 12th Int'l Conf. Knowledge Engineering and Knowledge Management*, Lecture Notes in Computer Science, vol. 1937, 2000, Springer Verlag, New York, 1-16
8. Grosso, W. E., Eriksson, H., Fergerson, R. W., Gennari, J. H., Tu, S. W., and Musen, M. A. Knowledge Modeling at the Millennium (The Design and Evolution of Protege-2000). *Proc 12th Knowledge Acquisition Workshop*, Banff, Alberta, Canada, October, 1999.
9. Gruber, T. R. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2). 1993.
10. Hatala, M., and Hreno, J. Annotation of documents with knowledge model concepts. *Enrich Report ID44-O-K*, 2000. (Available at <http://kmi.open.ac.uk/projects/enrich/id44.pdf>).
11. Heflin, J., Hendler, J. and Luke, S. Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. *AAAI-98 Workshop on AI and Information Integration*. (Available at <http://www.cs.umd.edu/projects/plus/SHOE/shoe-aaai98.ps>)
12. Heflin, J. Hendler, J., and S. Luke. Applying Ontology to the Web: A Case Study. In *Proc of the International Work-Conference on Artificial and Natural Neural Networks, IWANN'99*.
13. Kalfoglou, Y., Domingue, J., Carr, L., Motta, E., Vargas-Vera, M., and Buckingham Shum, S. On the integration of technologies for capturing and navigating knowledge with ontology-driven services. *Submitted to the 1st International Conference on Knowledge Capture (K-Cap '01)*, Victoria, BC, Canada, 2001.
14. Lave, J. and Wegner, E. *Situated Learning: Legitimate Peripheral Participation*, Cambridge University Press, Cambridge, UK, 1991.
15. Meseguer, P. and Preece, A. Verification and validation of knowledge-based systems with formal specifications. *The Knowledge Engineering Review*, 10(4), 331-343, 1995.
16. Motta E. *Reusable Components for Knowledge Models*. IOS Press, Amsterdam, 1999.
17. Motta, E., Buckingham Shum, S. and Domingue, J. Ontology-Driven Document Enrichment: Principles, Tools and Applications. *International Journal of Human Computer Studies*. 52(5), 1071-1109, 2000.
18. Riloff, E. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *The AI Journal*, 85, 101-134, 1996.
19. Riva, A. and Ramoni, M. LispWeb: a Specialized HTTP Server for Distributed AI Applications, *Computer Networks and ISDN Systems*, 28 (7-11), 953-961, 1996.
20. Staab, S., A. Mädche, S. Handschuh. An Annotation Framework for the Semantic Web. In: S. Ishizaki (ed.), *Proc. of The First International Workshop on MultiMedia Annotation*. January, 30 - 31, 2001. Tokyo, Japan.
21. Tennison, J. and Shadbolt, N. R. APECKS: a Tool to Support Living Ontologies. *Proc. of the 11th Banff Knowledge Acquisition Workshop*, Banff, Alberta, Canada, April 18-23, 1998.
22. Vargas-Vera, M., Domingue, J., Kalfoglou, Y., Motta, E. and Buckingham-Shum, S. Template-driven information extraction for populating ontologies. *Proc of the IJCAI'01 Workshop on Ontology, Learning*, Seattle, WA, USA 2001.
23. Wenger, E. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press: Cambridge, 1998.