Ontology-Driven Document Enrichment: Principles and Case Studies

Enrico Motta, Simon Buckingham Shum and John Domingue

Knowledge Media Institute
The Open University
Walton Hall, MK7 6AA
Milton Keynes, UK
{e.motta, s.buckingham.shum, j.b.domingue}@open.ac.uk

Abstract. In this paper we present an approach to *document enrichment*, which consists of associating formal knowledge models to archives of documents, to provide intelligent knowledge retrieval and (possibly) additional knowledge services, beyond what is available using 'standard' information retrieval and search facilities. The approach is *ontology-driven*, in the sense that the construction of the knowledge model is carried out in a top-down fashion, by populating a given ontology, rather than in a bottom-up fashion, by annotating a particular document. In the paper we give an overview of the approach and discuss its application to the domains of electronic news publishing, scholarly discourse and medical guidelines.

1. INTRODUCTION

An important activity in knowledge management is "to convert text to knowledge" (O'Leary, 1998). This activity is central to knowledge management for two reasons: i) work practices and information flow in organisations tend to be *document-centred* and ii) documents themselves do not normally exhibit the amount of structure required to support semantically-aware search engines or other forms of intelligent services. For these reasons there has been much interest in technology to support the specification of structured information in textual documents, especially web pages. Most of the work so far has focused on the underlying representational infrastructure: XML (XML, 1999) has been proposed as the basic annotation formalism to support the specification of structured information in web pages, while approaches such as Ontobroker (Fensel et al., 1998) and Shoe (Heflin et al., 1998) provide formalisms and associated interpreters to embed knowledge representation structures in web pages and use them to perform inferences.

In this paper we look at the wider issues concerning "the conversion of text to knowledge" and discuss an comprehensive approach to *document enrichment* (Sumner et al., 1998), which we are trying out in a number of projects here at the Knowledge Media Institute. In particular, we discuss the application of our approach to three application domains: *electronic news publishing* (Domingue and Motta, 1999), *scholarly discourse* (Buckingham-Shum et al., 1999) and *medical guidelines*.

The paper is organised as follows: in the next section we give an overview of our approach, in terms of the underlying methodological assumptions and the associated process model. In section 3 we describe the technology we have developed to support the approach. In sections 4,

5 and 6 we discuss the application of the approach to the three aforementioned domains. Finally, in sections 7 and 8 we discuss related work and reiterate the main contributions of this paper.

2. ONTOLOGY-DRIVEN DOCUMENT ENRICHMENT

Our approach is *ontology-driven*, in the sense that the construction of the knowledge model is carried out in a top-down fashion, by populating a given *ontology* (Gruber, 1993), rather than in a bottom-up fashion, by annotating a particular document - see figure 1. For this reason, we prefer to use the term "enrichment", rather than "conversion" or "annotation". That is, in our approach we do not aim to 'translate' or 'annotate' a document, but to 'enrich' it (Sumner et al., 1998). In general, a representation, whether formal, graphical or textual, can be enriched in several different ways - e.g., i) by providing information about the context in which it was created, ii) by linking it to related artefacts of the same nature, or iii) by linking it to related artefacts of a different nature. Although in our document-centred knowledge management work we provide multiple forms of document enrichment, such as associating *discussion spaces* to documents (Buckingham-Shum and Sumner, 1998), in this paper we will concentrate on the association of formal knowledge models to documents.

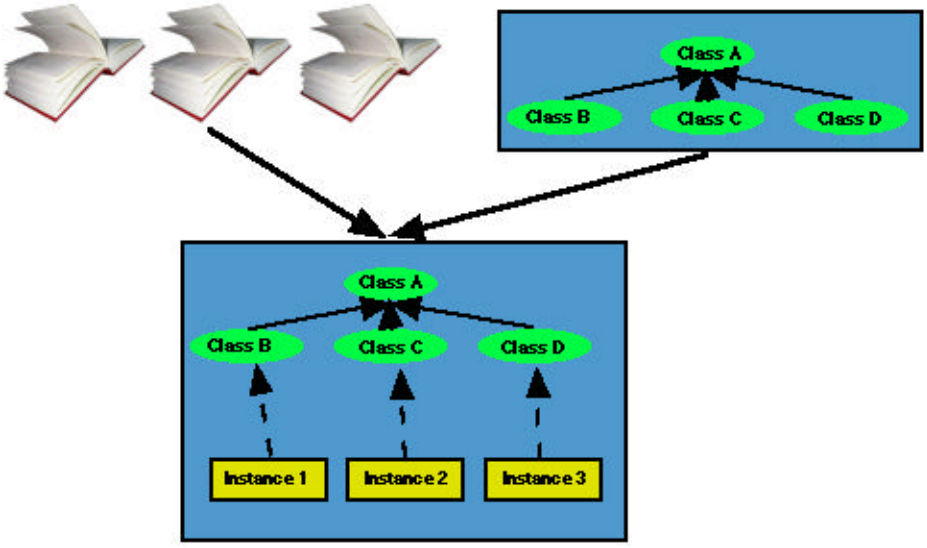

Figure 1. Ontology-driven Document Enrichment

Thus, an important facet of an ontology-centred approach to document enrichment is that the formalised knowledge is not meant to be a translation of what is informally specified in the associated document. Hence the knowledge model typically plays a different role from the associated text. For instance, in the medical guideline scenario the instantiation of the knowledge model helps to verify that all the kinds of knowledge expected to be found in a document describing a medical guideline are indeed there. In the scholarly discourse scenario the knowledge model is meant to capture the meta-knowledge required to structure academic debates (e.g., theory X contradicts theory Y), which is often expressed only implicitly in publications (i.e., acquiring it typically requires some interpretation effort) and is not modelled at all in traditional libraries. In a nutshell, the emphasis in our approach is in identifying the *added value* (in terms of enabling semantic retrieval capabilities or other reasoning services), which can be provided by a formalised knowledge model. Our methodology comprises the following six steps.

1. Identify use scenario.
2. Characterise viewpoint for ontology.
3. Develop the ontology.
4. Perform ontology-driven model construction.
5. Customise query interface for semantic knowledge retrieval.
6. Develop additional reasoning services on top of knowledge model.

These steps are briefly described in the next sub-sections.

2.1 Identify Use Scenario

At this stage the services to be delivered by the knowledge management system are defined. In particular, issues of feasibility and cost are investigated. Addressing the latter involves answering questions such as: "What is the added value provided by the knowledge model, considering the non-trivial costs associated with the development and instantiation of an ontology?", "Is there the need for a 'full-blown' knowledge model and for going beyond the facilities provided by off-the-shelf search engines?", "What additional reasoning services will be provided, beyond deductive knowledge retrieval?". Addressing feasibility issues requires assessing (among other things) whether or not it is feasible to expect the target user community to perform document enrichment or whether specialized human editors will be needed. There are subtle trade-offs involved here, which we address in section 2.4, when discussing the ontology instantiation task.

2.2 Characterise viewpoint for ontology

The previous step was concerned with assessing the feasibility of the approach and defining the functionalities of the envisaged system. This step focuses on the ontology: having decided on an ontology-based approach, it is important to characterize the particular viewpoint that the envisaged ontology will impose on the documents. For instance, in the electronic publishing domain we focus on the events characterizing academic life, while in the scholarly discourse domain the focus is on academic debate. Clearly the distinction between this step and the steps immediately before and after can in some cases be fuzzy. For instance, the specification of the viewpoint in the scholarly discourse domain is tightly integrated with the characterization of the use scenario. However, we believe that it is useful to explicitly distinguish a viewpoint specification task for two reasons. The first one is that it is feasible to envisage scenarios in which different viewpoints can be taken as the starting point for the ontology development process. Examples abound in the literature, especially with respect with highly generic ontologies - compare for example the CYC ontology (Lenat and Guha, 1990) with the work by Sowa (1995). Another reason is methodological: it is useful to separate the issues concerning the functionalities, scope and purpose of the envisaged system from the specification of the functionalities, scope and purpose of the ontology. For instance, in the electronic publishing scenario the knowledge management system is concerned with providing semantic retrieval capabilities and services supporting story identification and personalised news feeds - see section 4 for more details. Within this scenario the selected viewpoint focuses on modelling academic events and the key 'actors' in these events: technologies, people, organizations and projects.

2.3 Develop the Ontology.

Having defined a particular viewpoint over a domain (in this case, a set of documents), ontology development is largely a technical enterprise - i.e., the issues here concern modelling and formalisation, rather than scope and purpose. Several approaches to ontology development have

been proposed in the literature, which introduce distinctions along different dimensions - contrast, for instance, the *bottom-up* development style of van der Vet and Mars (1998) with Sowa's (1995) *top-down* approach. Uschold and Gruninger (1996) argue that a *middle-out, purpose-driven* approach is most effective, in which the basic concepts in a domain are identified first (e.g., dog), and later generalised (mammal) and/or specialised (cocker spaniel).

In our scenarios we have followed a task-independent, middle-out approach, and we use the selected viewpoint to help us to identify the *key concepts* in the class of models we want to construct. For instance, in the electronic newsletter scenario, a starting point was the notion of *news item*, which in turn was characterised as relating a number of *events*. Thus, our main modelling effort centred on identifying and modelling the various types of events relevant to our scenario (academic life). In the scholarly discourse domain, the viewpoint is 'academic debate' and the ontology then concentrates on characterising the most common relations which exist between academic theories, methodologies, models, etc.

Another important issue concerns who constructs the ontology? As pointed out earlier, we want to support scenarios in which knowledge models are constructed collaboratively. But what about the ontology itself? Is this constructed collaboratively? Our answer is negative. In all the projects we have carried out so far, we have centralised the ontology development. 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 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 or failure. Moreover, ontology design requires specialist skills which are normally not possessed by the members of our target user communities

Our ontology design approach is informed by two main modelling guidelines.

- *Minimal ontological definitions*. That is, to try and provide only the minimal set of attributes needed to define a particular class. This approach has the advantage that, when populating the ontology, users don't have to face lots of irrelevant attributes. For instance, in the electronic news publishing domain we initially reused the definition of class `event` given in the HPKB upper layer (HPKB, 1997), but we then removed 75% of its slots. The reason for this was that the HPKB definition aims to cover all potential attributes which *can* be relevant to a *generic instance* of class `event`. However, typically only a few slots will *actually* be relevant for any *specific subclass* of the class. For instance, slot `damages` is only relevant to events which can cause damage. Hence, we would introduce this slot when characterising a subclass of class `event`, such as `damaging-event`, rather than associating it with class `event` itself.
- *User-centred definitions*. This guideline requires that the terminology used by the ontology needs to be easy to understand for a user who is not necessarily a knowledge engineer. There are two aspects here: heavily technical modelling concepts – e.g., sophisticated modelling solutions for representing time – ought to be avoided. Moreover, the terminology should be as context-specific as possible. For instance, while we can talk about "agents performing events", when describing events in general, we should use the class-specific terminology, "awarding body assigns awards", when talking about an award-giving type of event. This latter guideline implies that the underlying modelling language should support slot renaming along *isa hierarchies* – i.e., inherited slots should get subclass-specific names. The importance of domain-specific, user-oriented

terminology has been recognised in knowledge acquisition for a long time (Musen, 1989) and arguably provides an important difference between the criteria associated with modelling for knowledge acquisition and those associated with modelling for system development.

2.4 Perform ontology-driven model construction

We are acutely aware that many schemes for registering shared resources and providing structured descriptions founder on the crucial 'capture bottleneck' - the envisaged beneficiaries of the system simply do not have the motivation or time to invest in sharing resources to reach a critical mass of useful material. Sobering lessons on this theme have been drawn for group and organisational memory systems (Selvin,1999), and indeed, for any system that requires users to formalise information (Shipman and Marshall, 1999). Why should we succeed where others have failed? Our working hypothesis is that our domains have unique characteristics lacking in domains in which collaborative development has failed.

- *Co-operation rather than competition*. We are selecting domains where co-operation, meant here as "willingness to share knowledge", is either a basic work premise or is enforced by external constraints. For instance, academic analysis and publishing require scholars to read, refer to and praise/criticise each other's work. In sum, the dynamics of academic publishing requires co-operation. A similar situation occurs in the medical guidelines scenario. Institutions and individual scientists may compete with each other, but the outcome of consensus conference (by definition) are shared knowledge resources. In other scenarios, for instance when constructing an organisational memory, other forces (e.g., directives from higher management) may force co-operation, even when competition would be the norm.
- *Benefits outweigh costs*. Motivation is a crucial aspect. Motivation essentially boils down to a cost-benefit analysis. For instance, in the scholarly discourse scenario, we assume that the basic motivation of an academic is to disseminate his/her work. Hence, having completed a new document, the author will want to maximise its 'digital presence' on the net by carefully encoding its contributions and connections to the existing literature.
- *Compatibility with organisational work practices.* This requires the seamless integration of our document enrichment model with existing work practices. For instance, in the case of electronic publishing, the ontology is used to enrich news items, which are submitted either through email or through a web-based form. Hence, at least for those users who submit through the latter mechanism, instantiating the ontology becomes an additional form-filling activity, carried out using the same medium (i.e., the web browser) and at the same time. Analogously, in the case of scholarly discourse, at least in some academic communities, authors are used to submitting papers to digital repositories and providing metadata. Filling an ontology-derived form should then be perceived as a small, additional step.

2.5 Customise query interface for semantic knowledge retrieval

At this stage the appropriate query interface is designed, in accordance to the use scenario and the expected functionalities. To support this step we have developed a flexible form-based interface, called *Lois*, which can be customised for each specific application domain. Lois is described in section 3.3.

2.6 Develop additional reasoning services on top of knowledge model

This final step is where the real benefit of the approach lies. Once a knowledge model has been produced, then it becomes possible to provide additional intelligent functionalities and ensure that the benefits outweigh the costs. These reasoning services tend to be application specific. For instance, in the scholarly publishing scenario, we are planning to develop specialised agents, whose goal is to identify emerging scholarly perspectives, using heuristic knowledge and machine learning techniques. For instance, an agent could discover a 'European perspective' on a particular issue, if a structural pattern in the knowledge model - e.g., use of formal methods - also matched the geographic location of the relevant researchers. In the electronic publishing domain we have designed two agents, which reason about the contents of the knowledge model to identify new, potentially 'hot' stories and to provide personalised news feeds - see section 4.2.5.

3. TECHNOLOGIES FOR ONTOLOGY-DRIVEN DOCUMENT ENRICHMENT

In this section we describe the main technologies we have developed to support document enrichment. These are as follows:

- *OCML*. An operational knowledge modelling language, which provides the underlying representation for our ontologies and knowledge models.
- *WebOnto*. A tool providing web-based visualisation, browsing and editing support for developing and maintaining ontologies and knowledge models specified in OCML.
- *Lois*. A flexible form-based interface for knowledge retrieval.
- *Knote*. A form-based interface for populating an ontology.

These technologies are described in the next sub-sections.

3.1 OCML, an operational knowledge modelling language

OCML (Motta, 1999) is an operational modelling language, which provides constructs for specifying relations, functions, classes, instances, rules and control structures. Operationality is supported by means of a function interpreter, a control interpreter and a proof system. The latter integrates inheritance and function evaluation with a backward chaining inference engine. OCML modelling is supported by a library of reusable definitions, which is structured according to the basic categories of our application modelling framework: *task*, *method*, *domain* and *application* (Motta, 1999). The library also relies on a number of *base ontologies*, which provide definitions for basic modelling concepts, such as numbers, sets, relations, tasks, methods, roles, etc. OCML has also been designed to be compatible with established standards such as Ontolingua (Gruber, 1993). In particular, OCML supports the following relation specification keywords used by Ontolingua: `:iff-def`, `:def`, `:sufficient` and `:axiom-def`. Moreover, the OCML base ontology also includes the constructs specified in the Ontolingua frame ontology, thus ensuring compatibility between frame-based specifications in the two languages. These capabilities allow Ontolingua users to use OCML as a kind of 'operational Ontolingua', providing function evaluation and deductive facilities for a subset of Ontolingua constructs. Such facilities are interactive and therefore support incremental model construction, rather than the 'batch mode' style of interaction associated with the translation approach used for operationalizing Ontolingua models in other languages (Gruber, 1993). Constructs for which

OCML does not provide operational support - e.g., axioms - are simply added to the model, but they are not used in the reasoning process.

Our library of OCML models can be accessed through the WebOnto browser at URL http://webonto.open.ac.uk.

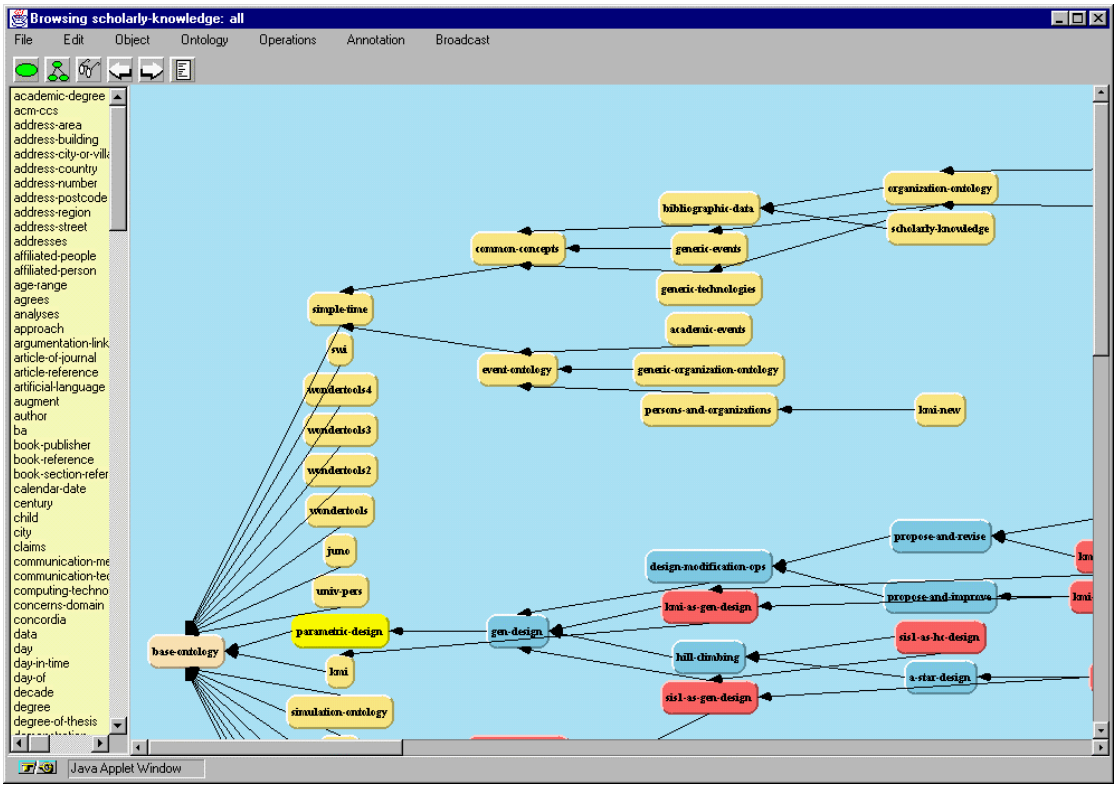

Figure 2. WebOnto visualization of part of the KMI library of knowledge models.

3.2 WebOnto: browsing and editing knowledge models on the web

WebOnto (Domingue, 1998) enables knowledge engineers to collaboratively browse and edit knowledge models over the web, using a standard web browser. The architecture is composed of a central server and a Java applet. WebOnto's central server is built on top of a customised web server, *LispWeb* (Riva and Ramoni, 1996) and uses OCML as the underlying modelling language. The WebOnto Java applet provides multiple visualisations of OCML knowledge models, a direct manipulation and forms interface for creating new knowledge structures, and a groupware facility which supports both synchronous and asynchronous model building by teams of knowledge engineers. A snapshot of WebOnto is shown in figure 2.

3.3 Lois, a flexible form-based interface for knowledge retrieval.

The aim of Lois is to provide an interface for posing queries to a knowledge model at a level which abstracts from the underlying modelling language. This goal has been accomplished by developing a form-based interface which allows users to select 'key concepts' in the ontology and then construct a query by navigating the structure of the ontology (i.e., by following relations between concepts). This navigation leads to the creation of a query as a *list of rows*, which are linked by logical connectives. For instance figure 3, which is taken from our electronic

publishing domain, shows a query which asks for a KMI researcher involved in the development of software visualisation technology. This query was constructed by selecting class kmi-member first, specialising it to kmi-researcher, selecting the relation develops-technology and then circumscribing the range of this relation to kmi-software-visualisation-technology. More specifically, the first row of the query was created by a) selecting the "Member of KMi" button, b) selecting "type" in the "Index Aspects" window, c) selecting "kmi-researcher" in the "Aspect Type" window, d) selecting the "Add Row" button, and finally e) selecting "develops-technology" from the menu at the end of the row. The second row was created in a similar fashion.

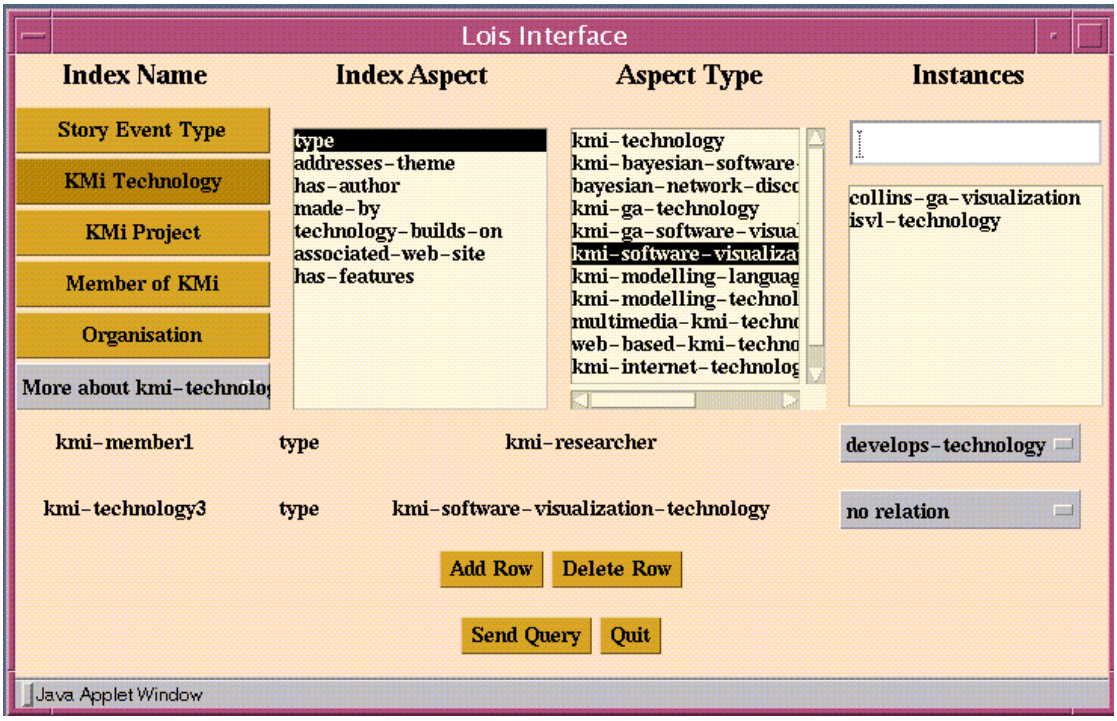

Figure 3. Finding a KMI researcher who works on software visualisation.

The Lois interface is created automatically once the key classes for a knowledge model have been specified. In the example, these are: story-event-type, kmi-technology, kmi-project, kmi-member and organization. The "Index Aspect" window shows the slots of the selected class. When the slot is constrained to have values of a certain type, the "Aspect Type" window displays the type (which is an OCML class) and its descendants. Thus the usability of the Lois interface is dependent on the underlying ontology. The knowledge engineer is required to choose class and slot names which can be understood by the Lois user. If this requirement is satisfied, then the user only needs to learn how to construct queries through the accumulation of rows.

3.4 Ontology instantiation using Knote

Our goal is to enable as wide an audience as possible to take part in the ontology-driven construction of a shared knowledge model. Knote was therefore designed to be 'low entry', so that users would not necessarily require a background in knowledge modelling. At the same time Knote should allow experienced ontology engineers the freedom to create arbitrarily complex OCML expressions. Knote provides instance forms which are modelled on the *dynamic forms* of

Girgensohn et al. (1995). The key difference between these two types of forms is that instance forms in Knote are generated directly from the ontology and not from a user description.

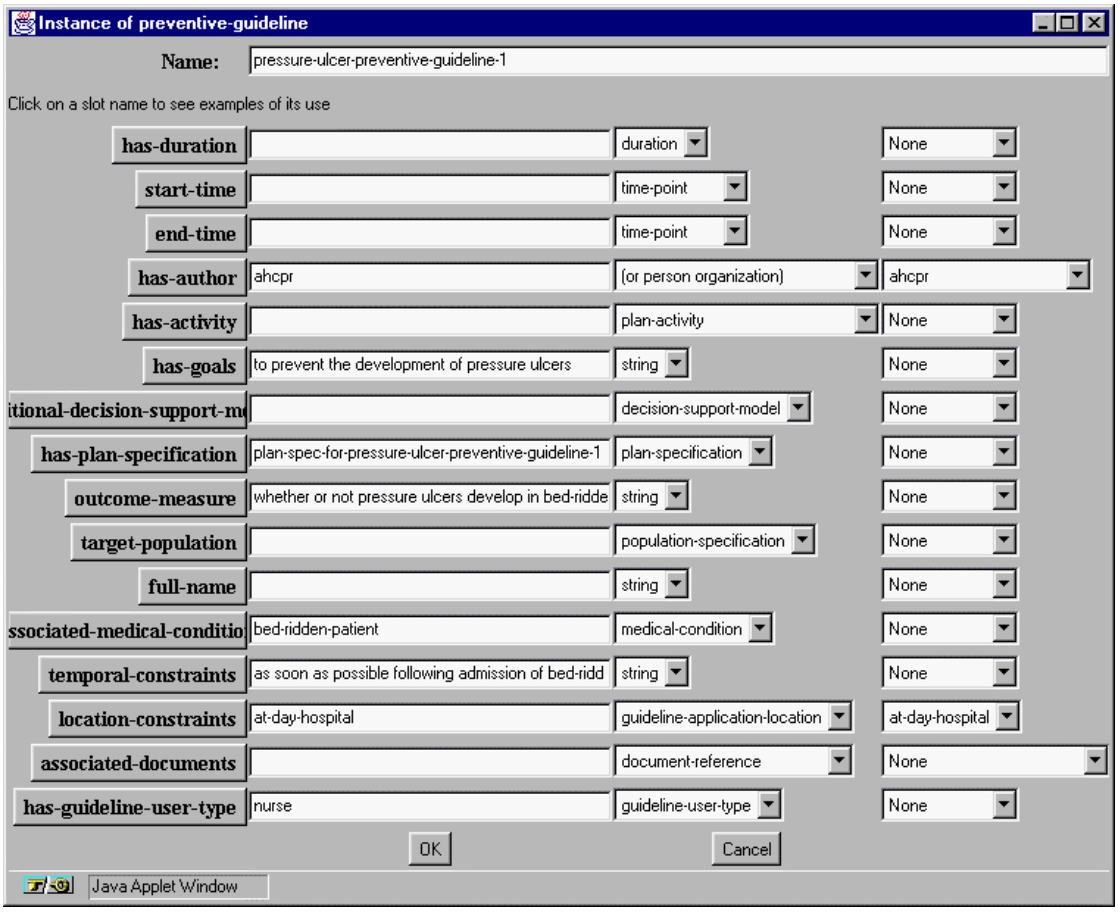

Figure 4. Class instantiation with Knote.

The Knote forms are created on the fly from the current class definition and are thus domain independent. This approach has the benefit that it is possible to edit classes and instances at the same time, which is often required during application development. This was not possible in systems such as Protégé-II (Eriksson et al., 1994), where the instance forms were created by a distinct meta tool. Recent work on Protégé-2000 (Eriksson et al., 1999) seeks to alleviate this by allowing classes to be viewed as instances of meta classes.

An example of a Knote form is presented in figure 4. The figure shows a form instantiating a preventive medical guideline. The structure of the form was derived from the definition of class `preventive-guideline` in the ontology modelling medical guidelines. Knote provides the user with quite a lot of help in filling the form. When a slot is typed, Knote allows the user to navigate the subclasses and the instances of the given type, to select an existing instance or to create a new one. In the latter case, the appropriate Knote form is then generated. The user can also click on a slot name (listed in the leftmost column of the form) to get examples of the use of the slot. Our experience suggests that this example-centred support tends to be more useful than the generic documentation associated with a slot.

With the description of the Knote form-filling support we have completed the brief overview of our technology for knowledge modelling. In the next sections we will illustrate examples of the approach in three application domains.

4. PLANET-ONTO: ENRICHING NEWS STORIES

4.1 The scenario

KMI Planet (Domingue and Scott, 1998) is a web-based news server which facilitates communication within our laboratory (the Knowledge Media Institute) and allows the wider community to access lab-related items of interest. Planet is used as the 'front door' to our laboratory, both metaphorically and physically (a dedicated machine running Planet is stationed at the entrance of the laboratory, so that visitors to KMI tend to be shown or 'play with' Planet first). Our archive is growing steadily and now contains about a hundred stories, submitted by 13 journalists. We currently have 480 registered 'readers', i.e., users who subscribe to the Planet alert services. Planet has been a 'success story' and numerous versions of the newsletter have now been produced for other organisations, both within and outside the Open University.

For all this success, it is apparent that, as the Planet archive and readership grow, more sophisticated mechanisms supporting semantic searches and individualised presentations and alerts are needed. Users of Planet often come across interesting news items but they cannot easily follow-on with obvious queries. For instance, having read a story about an award for a paper about visualising genetic algorithms, a user may want to find out who else works on software visualisation, what other projects are going on in this area, etc. Of course, many answers can be found by browsing our web pages and through standard search mechanisms. However, even in well-organised sites, many important relations between people, technologies, projects and organisations are often missing, leading to the 'standard' knowledge management problem of finding out who does what and who knows what.

In addition to the need for better search and retrieval facilities, the experience of a day-to-day use of Planet over more than two years has highlighted a number of other issues. An obvious one is the need for individualised news feeds and presentations – currently, registered readers periodically get a standard news update message. A less obvious issue is the need for Planet to move away from simply being a passive news archive and become a 'real newspaper', where news is not just passively received but proactively identified and requested, in accordance with events in the department and the observed interests of the readership. To address these issues we have developed an integrated suite of tools, *Planet-Onto*, which extends the original Planet news server by providing support for ontology-driven document enrichment, integrated browsing and deductive knowledge retrieval, personalised news feeds and alerts, and proactive identification of potentially interesting news items.

4.2 Instantiating the approach

4.2.1 Characterise viewpoint for ontology.

The viewpoint is centred on the range of academic events which characterise the life of an academic department. Hence, we have developed a rich taxonomy of events. In addition, we have identified five key classes, which have driven the development of the ontology: *people*, *organisations*, *stories*, *projects* and *technologies*.

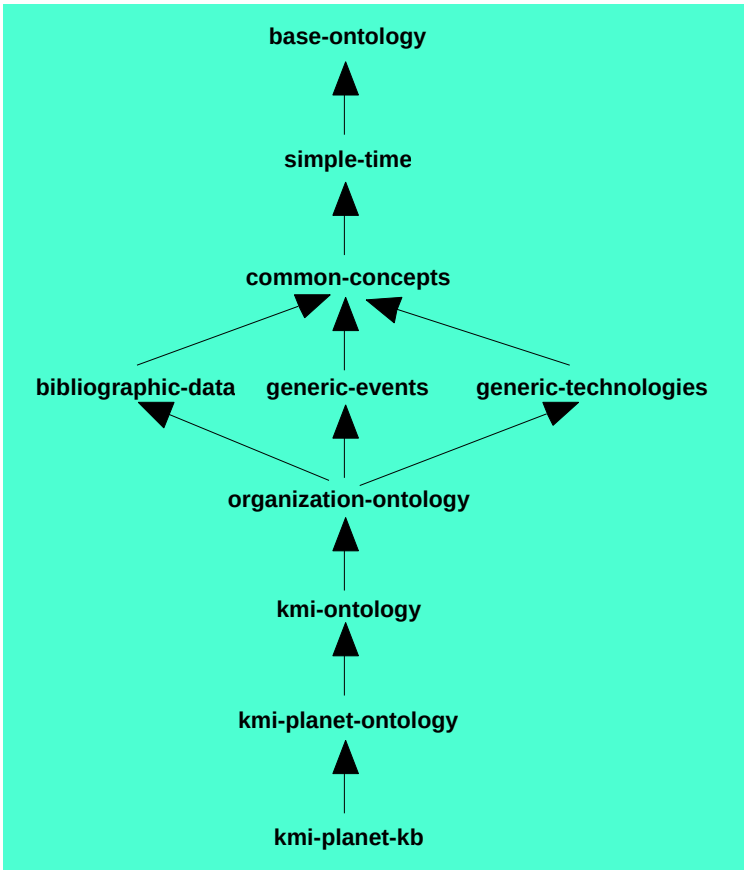

Figure 5. Ontology inclusion in the Planet Ontology.

4.2.2 Develop the Ontology.

The Planet ontology builds on a number of other ontologies included in the library, as shown in figure 5. Some of these ontologies were reused from existing repositories - e.g., simple-time and bibliographic-data, others were developed from scratch. It is difficult to quantify the overall development effort, as these ontologies are held in the WebOnto central repository and grow steadily. As an indication, we can point out that the first version of the Planet ontology took about two weeks to complete and proved adequate to support the range of queries envisaged in the use scenario.

4.2.3 Perform ontology-driven model construction.

A prototype model was built using a subset of the stories. This initial prototype was used to gather initial feedback about the technology and the approach, and used as a showcase demo to publicise the approach to KMI visitors. The experience from the first model construction exercise led to the implementation of renaming mechanisms in OCML. A new, complete version of the Planet knowledge base is currently being developed and is expected to be completed by the end of September, 1999. Full-scale trials of the Planet-Onto system will begin shortly after.

4.2.4 Customise query interface for semantic knowledge retrieval.

This step consisted of developing the Lois form shown in figure 3. As pointed out earlier, the design of the form was driven by the identification of the key concepts in the ontology.

Experience so far shows that the design covers the type of queries which are asked to Planet-Onto - i.e., queries tend to be centred around key concepts.

4.2.5 Develop additional reasoning services on top of knowledge model.

We have designed (but not yet implemented) two intelligent agents, whose purpose is to provide additional services by reasoning about the formalised knowledge model. These are described below.

4.2.5.1 Story chasing with NewsHound

NewsHound will periodically gather data about 'popular' news items and will use these data to solicit potentially popular stories from the appropriate journalists. This will be accomplished by identifying 'gaps' in the knowledge base - e.g., projects about which there is no information. The design of NewsHound is consistent with one of the main goal in the design of KMI Planet: the system should try and emulate a news room team. One of the tasks that a news editor carries out is to identify potentially popular stories and assign them to one of the journalists in the staff. NewsHound is meant to emulate this behaviour. In order to identify potentially interesting stories, NewsHound will use two main types of data: statistics on access to individual stories in KMI Planet and records of the queries posed through Lois. Each story within Planet keeps a record of its popularity by counting the number of times the full text is requested from the KMI Planet server. Once NewsHound identifies a story as 'popular', then it tries to identify related stories which have the potential to be popular. To perform this task NewsHound analyses the knowledge base trying to find items of interest that have not yet been covered by Planet stories. Typically, these would be projects and technologies which i) are known to NewsHound, ii) are 'related' to 'popular' projects and technologies, but iii) have not yet been covered by a story. The term 'related' is the key here. NewsHound will use various heuristics to define 'relatedness'. For instance direct subclasses of the same class are considered related; technologies are related if they build on the same underlying technology; projects are related if they tackle the same areas. These heuristics are of course completely 'soft' and modular and therefore any new one can be added without affecting the existing ones.

An interesting feature of NewsHound has to do with the unique scenario in which it examines a knowledge base for gaps. Typically, completeness in a knowledge base is defined with respect to logical or task-related properties (van Heijst, 1995). In our scenario incompleteness is defined in pragmatics terms: publications need popular stories.

4.2.5.2 Providing personalised alerts with NewsBoy

Lois has been designed (among other things) to help users to track down Planet stories with very specific characteristics. However, a significant number of users prefer to work with *push technology*, that is they prefer to be automatically notified about potentially interesting stories, rather than having to query Lois about them. We have therefore designed an agent, *NewsBoy*, to provide a mode of use that is complementary to the one supported by Lois. NewsBoy enables users to create a personalised front-page to which interesting stories are 'pushed'.

When a new story is formalised and added to the Planet-Onto repository, NewsBoy matches the story against the specified interests of registered readers. Readers whose interests match that of the story are notified by email that a new story has been added to their personal Planet page. To make an explicit declaration a reader simply specifies a number of queries using the Lois interface. The reader is then updated when a new story matches at least one of the logged queries. Alternatively, a reader can state that she would like NewsBoy i) to log all the queries she

makes using Lois and ii) to create a user profile from the log. The resulting user profile is simply the logical disjunction of the queries contained within the log.

It is interesting to compare NewsBoy to other approaches which attempt to infer user profiles from analysing patterns of access to documents – e.g., see (Liebermann, 1995; Krulwich and Burkey, 1997). These approaches try to induce user interests using empirical methods. Our approach is semantic-centred: the user herself specifies the range of documents of potential interest through unambiguous declarative specifications.

5. SCHOLONTO: SUPPORTING SCHOLARLY DEBATE

5.1 The scenario

Contextualising ideas in relation to the literature is a fundamental task for authors and readers—are they new, significant, and trustworthy? Scholars accomplish this task firstly by bringing to bear their own knowledge of the field. This process leads to *commentary and discourse* of various kinds, which reflect the extent to which peers regard an author's work as authoritative. These can take the form of private exchanges, formal peer review of conference/journal submissions, or published reviews of literatures and books. We can think of conventional scholarly publication and debate as a document-centred, text-based process. Text is a rich medium in which to publish and discuss ideas in detail and with subtle nuances, but the corresponding disadvantage is that it takes a long time to read and is hard to analyse computationally.

A complementary approach focuses on the conceptual models implicit in textual documents and discourse. The goal is to provide a *summary* representation of ideas and their interconnections, in order to assist literature-wide analysis. We believe that this has advantages over textual media for tracing the intellectual lineage of a document's ideas, and for assessing the subsequent impact of those ideas, that is, how they have been challenged, supported and appropriated by others. In addition, the availability of explicit conceptual models opens possibilities for automatic analysis of a community's collective knowledge.

We begin with the idea that an author's goal is to persuade the reader to accept his/her *perspective*, which constitutes a set of claims about the world. Usually, the author has some new ideas that s/he is contributing, and asserts particular relationships between these and existing ideas already published in order to demonstrate both the reliability of the conceptual foundation on which s/he is building, and the innovation and significance of the new ideas. The scholarly reader's task is to understand which ideas are being claimed as new, and assess their significance and reliability. Let us switch from a reading scenario to the scenario of literature search and analysis. In this case, the scholar has some ideas and relationships in mind that s/he is trying to locate in the literature—has anyone written about them, or perhaps these ideas exist but not yet in a single document? The interpretative task includes i) formulating the ideas of interest in ways that may uncover relevant documents, ii) reading the documents (as just described), and then iii) interpreting them to characterise any patterns that appear to emerge. This is a similar scenario to that of a newcomer to a scholarly community - e.g., a student, librarian, lecturer or researcher from another discipline, who wants to know, for instance, what the seminal papers are, or if there are distinctive perspectives on problems or techniques that define that community.

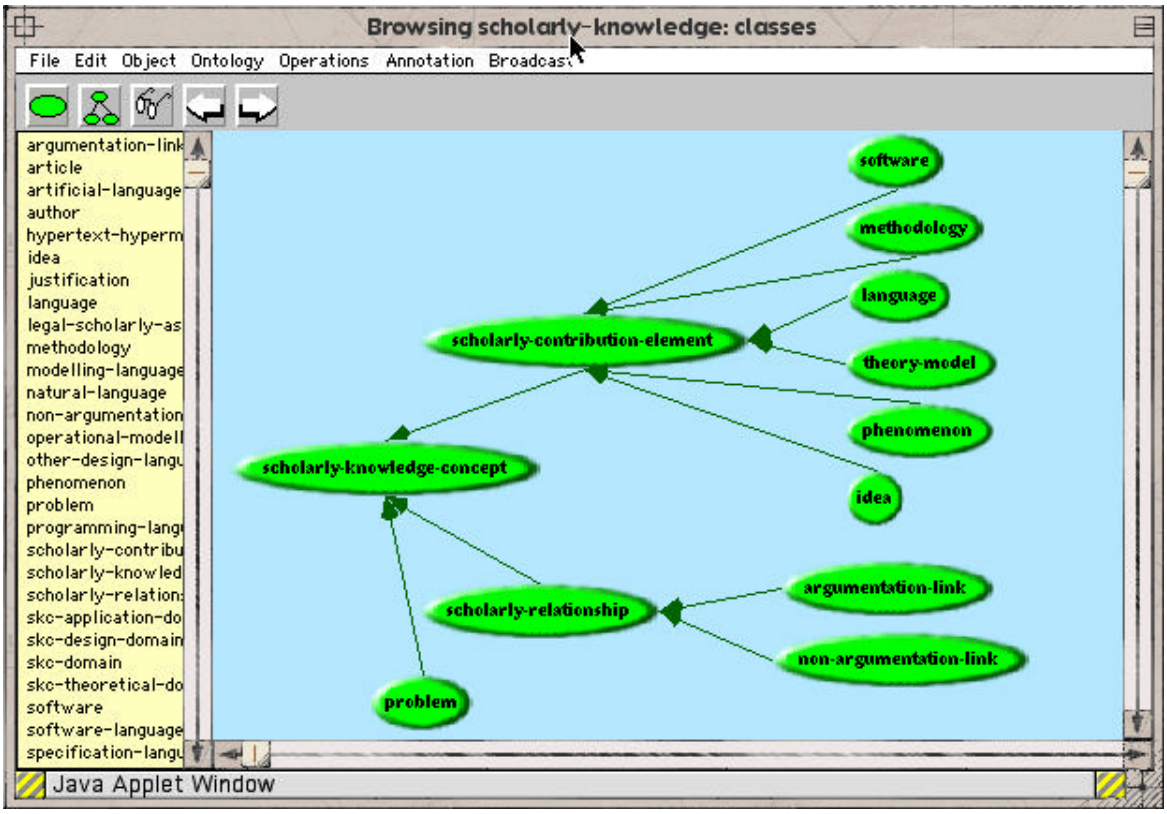

Figure 6. Main taxonomy of the ScholOnto ontology.

We contend that scholars are very poorly supported in these tasks by conventional library and technological environments, but that digital libraries open up new possibilities which have yet to be exploited. Consider the document interpretation scenario. In the non-digital world, there is currently no way beyond following citations (only those provided by the author), or using citation indices (to find others citing him for some reason), to ask questions such as: "Has anyone built on the ideas in this paper, and in what way?", "Has anyone challenged this paper?", "Has anyone proposed a similar solution but from a different theoretical perspective?". These are arguably the kinds of phenomena of most interest to scholars when they read or write papers, engage in debate or search the literature. These are also the kinds of questions asked by researchers unfamiliar with a literature. Our goal is to support these kinds of queries by means of our enriched-document approach. Specifically, we have developed an ontology to support scholarly debate and we plan to use this ontology to characterise scholarly relations between documents.

5.2 Instantiating the approach

5.2.1 Characterise viewpoint for ontology.

It might appear paradoxical to propose the use of ontologies to support scholarly communities in managing their knowledge, since conflicting worldviews, evidence and frames of reference lie at the heart of research and debate. Of course, the key issue is what is represented. It is hard to envisage when scholarly communities will no longer need to make claims about, or contest, the nature of a document's *contributions* - e.g., "this is a new theory, model, notation, software, evidence", or its *relationships* to other documents - e.g., "it applies, modifies, predicts,

refutes…". Our approach builds on this relatively stable dimension of what are otherwise constantly evolving research fields, by representing scholars' *claims about the significance* of ideas and concepts—a focus on *discourse* and *argumentation* (how scholars support and contest claims), and on *context* (the conceptual network in which an idea is embedded). Representing concepts separately from claims about them is critical to supporting multiple perspectives.

5.2.2 Develop Ontology.

Figure 6 shows the top level structure of the ontology. Both nodes and links in the semantic network created by scholars' submissions are *scholarly knowledge concepts.* Nodes are *scholarly contribution elements*, and links *scholarly relationships.* These are subdivided into *argumentation* and *non-argumentation links*. The ontology is designed to support scholars in making *claims* by asserting relationships between concepts. Other scholars may *support*, *raise-issues-with*, or *refute* these claims. A claim is formally defined as a relation between a set of authors, who make a `legal-scholarly-assertion`, with some justification.

The design of the ontology was based on the analysis of scholarly articles from a range of different fields, and took about two person weeks' effort. Once the top level structure stabilised, it required only two days to develop the first version of the ontology in OCML.

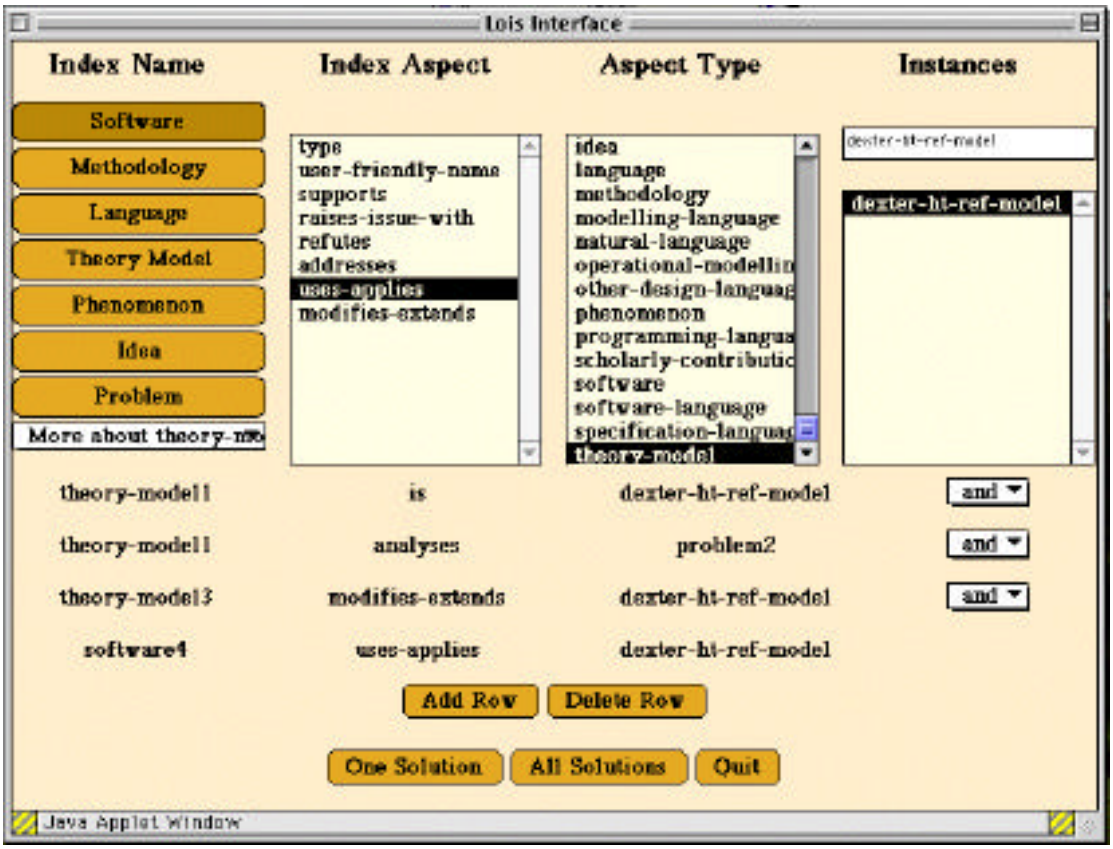

Figure 7. A forms-based interface generated automatically from the *ScholOnto* ontology.

5.2.3 Perform ontology-driven model construction.

We are currently seeding a knowledge base with document descriptions to test the ontology's ability to support the scenarios described in this paper.

5.2.4 Customise query interface for semantic knowledge retrieval.

A prototype form has been designed, which is shown in figure 7. In the example the user is trying to assess the motivation behind, and impact of, a theoretical model called the *Dexter Hypertext Model*. The queries specify, respectively, i) what problems does it analyse? ii) are there any theory-models which modify-extend it? and iii) is there any software which uses-applies it?

5.2.5 Develop additional reasoning services on top of knowledge model.

A knowledge model enables inference-based searching and alerting. It will be possible to ask the system questions such as "What impact did Theory T have?", since "impact" can be defined, for example, in terms of the number of subsequent documents using or modifying it, the number of different domains in which it has been applied, the number of problems addressed which drew on the theory, and so forth. Our knowledge modelling environment makes it simple for us (as system maintainers) to write heuristics that could assist in finding relevant documents - e.g., "if Method Y extends Method X, and Method X is challenged, then Method Y may be challenged". Moreover, as already mentioned, it will be possible to develop specialised agents whose goal is to identify emerging perspectives, using heuristic knowledge and machine learning techniques. As these machine-discovered assertions are added to the knowledge model, software agents effectively become actors in the scholarly debate. This scenario provides an example of a hybrid agent community and therefore raises a whole host of interesting issues, from the dynamics of social interaction to the design of epistemic agents - see (Masterton, 1998) for an another example and for a detailed discussion of the relevant issues. More in general, the scenario is an example of the general trend towards reducing the boundaries between humans and machines (Stutt and Motta, 1998).

6. KNOWLEDGE MANAGEMENT OF MEDICAL GUIDELINES

In the EC-funded PatMan project (PatMan, 1998) we are developing a number of technologies to support guideline-centred patient management. As part of this work, one of the project partners, the Medical Informatics group at the University of Pavia, has produced an editor and an interpreter to support the specification and execution of medical guidelines. However, an important problem which has arisen is that the formalised model of a medical guideline, to be used in the context of the guideline editor/interpreter, necessarily represents only a small subset of the knowledge typically expressed in a document describing a medical guideline. To ameliorate this problem we have started work on integrating the guideline editor with a knowledge model, developed according to the approach described in this paper. The aim is to provide intelligent knowledge retrieval facilities which can augment the information provided by the editor. At the same time, we expect to use the ontology to evaluate the completeness of the knowledge expressed in the guideline document.

This work is still in the early stages and so far we have only worked on the design of the ontology, which builds on a pre-existing generic medical ontology developed in an earlier project, HC-ReMa (HC-ReMa, 1997). We took the notion of medical guideline as our starting point and most of the effort was spent in characterising this class and the associated classes and relations - e.g., *outcome-measure*, *guideline-user-type*, etc. Because the users are expected to be health professionals we do not envisage the need to provide 'deep' medical models, beyond the formalization associated with a medical guideline. A first version of the ontology was developed in four person-days.

As shown in figure 8, a medical guideline is modelled as a subclass of class `plan`, which is in turn characterised as a `temporal-thing`. Hence, the slots associated with a plan specification (i.e., the slots typically used by guidelines interpreters) are kept separate from additional information about the guideline. The OCML definition of class `medical-guideline` is given in figure 9.

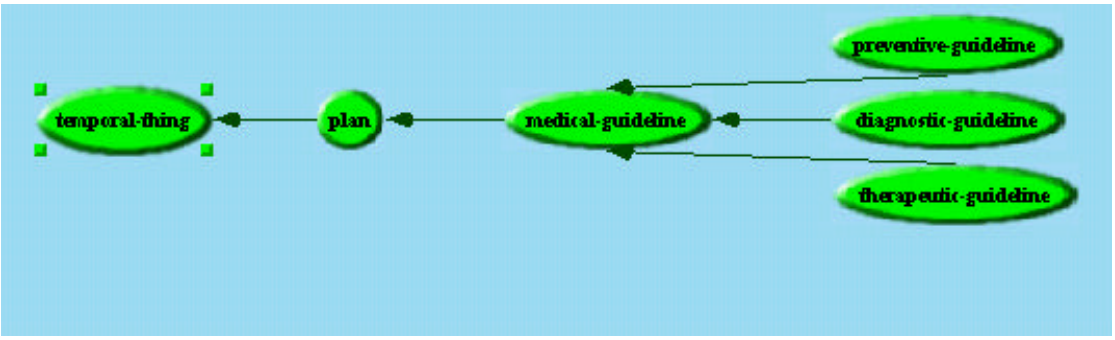

Figure 8. Isa hierarchy for class `medical-guideline`.

```
(def-class medical-guideline (plan)
  "Each guideline is associated with a medical condition. It also targets a
particular population"
  ((outcome-measure :type string)
   (target-population :type population-specification)
   (full-name :type string)
   (associated-medical-condition :type medical-condition)
   (temporal-constraints :type string)
   (location-constraints :type guideline-application-location)
   (associated-documents :type document-reference)
   (has-guideline-user-type :type  guideline-user-type)))
```

Figure 9. OCML definition of class `medical-guideline`.

We have started testing the adequacy of the ontology on a guideline document on the prevention, detection and treatment of high blood pressure. Early results are encouraging although more data are needed to fully evaluate the ontology.

7. RELATED WORK

From a representational and technical perspective, our approach differs from metadata-centred approaches - e.g., W3C (1997), in that ontologies support more sophisticated modelling - e.g., for specifying relations. Moreover, languages such as OCML and Ontolingua also provide metalevel modelling support which makes it possible to reason about the ontology itself.

The Shoe project (Heflin et al., 1998) 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. We take a holistic approach to document enrichment and we look at the wider issues concerning usability and sustainability. Thus, we are not just concerned with providing a mechanism for associating knowledge structures to text but we wish to develop a comprehensive architecture addressing all the relevant issues, from the 'right' approach to ontology development to the required visualisation and interface tools needed to facilitate the development of

knowledge models. Having said so, the technical solutions provided by Shoe could be easily integrated within the Planet-Onto framework. For instance OCML structures could be represented in terms of the relevant Shoe tags.

The (KA)2 initiative (Benjamins et al., 1998) shares a number of commonalties with our work. As in the case of Planet-Onto the aim of (KA)2 is to allow a community to build a knowledge base collectively, by populating a shared ontology. In the case of (KA)2 the knowledge base is meant to document the activities of the knowledge acquisition community. 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 (Fensel et al., 1998). In this paper we have emphasised that the feasibility of the idea of a collective construction of a knowledge base crucially depends on a number of features, including: i) a carefully defined ontology; ii) an underlying modelling language providing user-oriented facilities, such as context-dependent renaming; iii) a user-friendly ontology instantiation environment; and iv) the right motivational stimuli for the participants. In their paper on the (KA)2 initiative, Benjamins et al. (1998) mainly focus on the latter issue. However it seems to us that a careful analysis of all the issues associated with collaborative knowledge modelling is required, in order to manage the risks associated with such enterprises. In particular we believe that a careful design of the underlying ontology is particularly important. For this reason, in contrast with the case of (KA)2, the design of the ontology (but not the ontology population process) is centralised in our approach.

In terms of the underlying architecture the main difference between our approach and other approaches to adding semantic information to web pages is that we decouple the web pages from the knowledge model. This means that we do not directly annotate web pages but the collaboratively constructed knowledge base is held centrally in a server. As a result, changes made to either the ontology or the knowledge base are immediately available to the users. Moreover, decoupling knowledge models from documents also emphasizes that the latter are not the exclusive source of knowledge for the former. We see formalized knowledge servers as playing a similar role to audio, video and 'vanilla web' servers. Each type of server plays a distinct role and provides a distinct set of services. A key to successful knowledge management is to integrate these different media to provide the appropriate services in the relevant scenarios. We refer to the collection of these services as *knowledge media*.

8. CONCLUSIONS

In this paper we have described an ontology-centred approach to knowledge management. The approach and the underlying technology have attracted considerable interest and at the moment we have seven ongoing projects which are testing out this approach in domains ranging from students' help desks to managing best practice in the aerospace industry, to engineering design. Nevertheless, it is still early days and we do not yet have enough data to evaluate the approach fully. In this paper we have tried to highlight the key issues and we have emphasised that the success of these enterprises crucially depends on the successful management of a number of user-centred, task-centred and model-centred issues. The next few months will tell us whether we have been able to tackle these issues successfully in the domains described in this paper.

ACKNOWLEDGEMENTS

Thanks to the KAW-99 reviewers for the many insightful and stimulating comments. This research is partially supported by the CEC-funded Enrich (P29015) and PatMan (P4017) projects. Enrich is part of the ESPRIT programme on IT for Learning and Training in Industry; PatMan is part of the Healthcare Sector of the Telematics Application Programme.

REFERENCES

Benjamins, R., Fensel, D. and Gomez Perez A. (1998). Knowledge Management through Ontologies. In U. Reimer (editor), *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management.* 29-30 October, 1998, Basel, Switzerland

Buckingham Shum, S., Motta, E., & Domingue, J. (1999). Representing Scholarly Claims in Internet Digital Libraries: A Knowledge Modelling Approach. In S. Abiteboul and A.-M. Vercoustre (Ed.), *Proc. of ECDL'99: Third European Conference on Research and Advanced Technology for Digital Libraries*. Paris, France, September 22-24, 1999: Springer-Verlag (Lecture Notes in Computer Science). Available at: http://kmi.open.ac.uk/projects/scholonto/

Domingue, J. (1998). Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In B. Gaines and M. Musen (editors), *Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop*, April 18th-23th, Banff, Canada. Available online at http://kmi.open.ac.uk/people/domingue/banff98-paper/domingue.html.

Domingue, J. B. and Motta, E. (1999). A Knowledge-Based News Server Supporting Ontology-Driven Story Enrichment and Knowledge Retrieval. In D. Fensel and R. Studer (editors), *Proceedings of the 11th European Workshop on Knowledge Acquistion, Modelling, and Management (EKAW '99)*, LNAI 1621, Springer-Verlag..

Domingue, J. and Scott, P. (1998). KMI Planet: Putting the Knowledge Back into Media. In M. Eisenstadt, and T. Vincent, (editors), *The Knowledge Web: Learning and Collaborating on the Net,* Kogan Press, pp. 173-184.

Eriksson, H., Puerta, A. R. and Musen, M. A. (1994). Generation of knowledge-acquisition tools from domain ontologies. International Journal of Human Computer Studies, 41, 425-453.

Eriksson, H., Fergerson, R. Shahar, Y. Musen, M. A. (1999). Automatic Generation of Ontology Editors. Proceedings of the 12th Banff Knowledge Acquisition Workshop, Banff, Alberta, Canada, October 16-22, 1999.

Fensel, D. Decker, S., Erdmann, M. and Studer, R. (1998). Ontobroker: The very high idea. *Proceedings of the 11th Annual Florida Artificial Intelligence Research Symposium (FLAIRS-98).*

Girgenshohn, A., Zimmermann, B., Lee, A., Burns, B. and Atwood, M. E. (1995). Dynamic Forms: An Enhanced Interaction Abstraction Based on Forms. *Proceedings of Interact '95*, pp. 362-367.

Gruber, T. R. (1993). A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition, 5(2).*

Heflin, J., Hendler, J. and Luke, S. (1998). Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web. *AAAI-98 Workshop on AI and Information Integration*. Available online at http://www.cs.umd.edu/projects/plus/SHOE/shoe-aaai98.ps.

van Heijst, G. (1995). *The Role of Ontologies in Knowledge Engineering*. PhD thesis, University of Amsterdam.

HC-ReMa (1997). Health-Care Resource Management. *Telematics Applications Project HC 3103*. Project Description available at http://aim.unipv.it/projects/hcrema/.

HPKB (1997). High Performance Knowledge Bases. Darpa Project. Project Description available from http://www.teknowledge.com:80/HPKB/.

Krulwich, B. and Burkey, C. (1997). The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction. *IEEE Expert Intelligent Systems and their Applications, 12(5)*, pp. 22-27.

Lenat, D.B. and Guha, R.V. (1990). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Addison-Wesley, Reading, MA.

Lieberman, H. (1995). Letizia: An Agent That Assists Web Browsing. *International Joint Conference on Artificial Intelligence, IJCAI '95*. Montreal, August 1995.

Masterton, S. (1998). The Virtual Participant: A Tutor's Assistant for Electronic Conferencing. In Eisenstadt, M. and Vincent, T. (editors), *The Knowledge Web*, Kogan Press.

Motta E. (1999). *Reusable Components for Knowledge Models*. IOS Press, Amsterdam.

Musen, M. A. (1989). *Automated Generation of Model-Based Knowledge Acquisition Tools*. Research Notes in Artificial Intelligence, Pitman, London.

O'Leary, D. E. (1998). Knowledge Management Systems: Converting and Connecting. *IEEE Intelligent Systems, 13(3)*, pp. 30-33. May/June 1998.

Patman (1998). Patient Management Workflow Systems. *Telematics Applications Project HC 4017*. Project Description available at http://aim.unipv.it/projects/patman/.

Riva, A. and Ramoni, M. (1996). LispWeb: a Specialized HTTP Server for Distributed AI Applications, *Computer Networks and ISDN Systems, 28 (7-11)*, 953-961. (also available at http://kmi.open.ac.uk/~marco/papers/www96/www96.html).

Selvin, A. (1999). Supporting Collaborative Analysis and Design with Hypertext Functionality. *Journal of Digital Information*, 1, (4). Available at: http://jodi.ecs.soton.ac.uk/Articles/v01/i04/Selvin/.

Shipman, F. M., & Marshall, C. C. (1999). Formality Considered Harmful: Experiences, Emerging Themes, and Directions on the Use of Formal Representations in Interactive Systems. *Computer Supported Cooperative Work*, In Press.

Sowa J. F. (1995). Top-Level Ontological Categories. *International Journal on Human-Computer Studies*, 43(5/6), pp. 669-685.

Stutt, A. and Motta, E. (1998). Knowledge Modelling: An Organic Technology for the Knowledge Age. In Eisenstadt, M. and Vincent, T. (editors), *The Knowledge Web*, Kogan Press.

Sumner, T., and Buckingham Shum, S. (1998). From Documents to Discourse: Shifting Conceptions of Scholarly Publishing. *Proc. CHI 98: Human Factors in Computing Systems*, (Los Angeles, CA), 95-102. ACM Press: NY. Available at: http://kmi.open.ac.uk/techreports/papers/kmi-tr-50.pdf

Sumner, T., Domingue, J. Zdrahal, Z., Hatala, M. Millican, A., Murray, J., Hinkelmann, K., Bernardi, A., Weiss, S., Traphoner, R. (1998). Enriching Representations of Work to Support Organisational Learning. In *Proceedings of the Interdisciplinary Workshop on Building, Maintaining, and Using Organizational Memories (OM-98)*. 13th European Conference on Artificial Intelligence (ECAI-98), 23-28 August, Brighton, UK. Available at: http://kmi.open.ac.uk/projects/enrich/enrich-oms98-paper.html.

Uschold M. and Gruninger M. (1996). Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review*, 11(2), pp.93-136.

van der Vet, P. E. and Mars, N. J. I. (1998). Bottom-up Construction of Ontologies. *IEEE Transactions on Knowledge and Data Engineering, 10(4)*, pp. 513-526.

XML. (1999). http://www.oasis-open.org/cover/xml.html.

W3C (1997). Metadata and Resource Description. World Wide Web Consortium. http://www.w3.org/Metadata/.