Open Research Online

The Open University's repository of research publications
and other research outputs

PlanetOnto: from news publishing to integrated knowledge management support

Journal Item

oro.open.ac.uk

A Knowledge-Based News Server Supporting Ontology-Driven Story Enrichment and Knowledge Retrieval

John Domingue and Enrico Motta

Knowledge Media Institute
The Open University
Walton Hall,
Milton Keynes, UK

{j.b.domingue, e.motta}@open.ac.uk
http://kmi.open.ac.uk/people/{domingue, motta}

Abstract. We consider a knowledge management scenario in which members of an academic community collaboratively construct and share a common archive of news items. Given this scenario, a number of knowledge management challenges arise: how to organize a speedy, low overhead publication process which can nevertheless yield high quality results; how to provide semantic search and knowledge retrieval facilities in an effective and sustainable way; how best to provide individualized presentations and news alerts. To address these questions we have drawn on a number of technologies, including knowledge modelling, autonomous agents, software visualization, knowledge acquisition and distributed computing. In the paper we describe the resulting *Planet-Onto* architecture, which provides an integrated set of tools to support news publishing, ontology-driven document formalization, story identification and personalized news feeds and alerts.

1   Introduction

Loosely speaking, *knowledge management* is about facilitating the generation, sharing and use of knowledge. Thus, any activity or tool which fosters communication and sharing in a community can be seen in principle as an exercise in knowledge management – for instance, bandwidth improvements to an Intranet. Having said so, when discussing knowledge management it is useful to try and refine the all-encompassing span of this research area, by circumscribing the range of issues and technologies under examination. In the context of this paper, rephrasing and refining the definition given by O'Leary [29], we say that we are interested in the *computer-*

mediated management of explicitly represented knowledge. That is, we focus on issues of organisation, formalisation and distribution of knowledge stored in a network of computers. It is important to emphasise that we do not impose any constraint on the form of the representation – e.g. whether the knowledge in question is represented as text, in a conventional database, or in some knowledge representation formalism.

In particular, in this paper we consider a scenario in which members of an academic community collaboratively construct and share a common archive of news items. Given this context, our main research goal is to develop a suitable computational infrastructure, which can effectively support the publishing process, as well as facilitating access to the archive of news items.

To characterise the activities implied by our scenario we can use the framework proposed by O'Leary [29], which proposes an elegant typology of knowledge management activities as specialised *connecting* and *converting processes.* Specifically, using O'Leary's terminology, our goal is to provide solutions for the following classes of knowledge management processes:

- *Converting individual to group knowledge.* In our context this means the provision of tools for supporting 'journalists' submitting stories to a news archive.
- *Converting text to knowledge.* That is, formalising the knowledge expressed by the news item and integrating it into a knowledge base associated with the archive.
- *Connecting people to knowledge.* That is, providing integrated visualisation, search and query answering facilities, to allow users of the archive to quickly home in on information at different levels of granularity, from (sub-)collections of stories to specific data (whether they are explicitly included in a document or implied by the collection of documents).
- *Connecting knowledge to people.* That is, pro-actively contacting journalists and readers. The former should be contacted to solicit stories useful to plug 'holes' in the archive; the latter should be contacted when items of interest are published.

Actually, while O'Leary's framework provides a useful typology for characterising knowledge management activities, we ought to point out that his use of the term "converting" is not necessarily the correct way to talk about knowledge transformation activities in our (and maybe any other) scenario. The problem is that the term "converting" has a translation-centred connotation, which is misleading. In particular, in our scenario the formalisation process is driven by an *ontology* [18], which defines the concepts needed to describe events related to academic life, e.g. projects, products, seminars, publications, etc. This means that the parts of a news story not relevant to the ontology are ignored, much as in template-driven *information extraction* approaches [9] [33]. That is, not all knowledge expressed in textual form in a news item is expected to be formalised. On the other hand, because i) the formalisation is ontology-driven and ii) instantiating an ontology might require the specification of knowledge not given in the story, the knowledge base associated with the news archive would normally contain knowledge that is not present in the archive. In sum, the knowledge base associated with the news archive provides neither a subset nor a superset of the knowledge expressed in the stories. This is of course not very surprising. We know that any translation or encoding process has more to do with reconstructing meaning than with replacing representations. This is true not only

for translations between different natural languages, but also for translations from text to code [6] and from code to code [8]. Hence, the metaphor of converting is not the right one when talking about formalising knowledge in a document, both for fundamental reasons and, in our case, because of the different roles played by the news stories and the associated representations. For these reasons we prefer to use the metaphor of *enriching a representation* [40].

The paper is organised as follows. In the next section we provide an overview of the overall architecture, which is called *Planet-Onto*. In section 3 we describe the *Planet* news server [11]. In section 4 we describe the ontology which we use to drive the representational enrichment of news items. In section 5 we illustrate the ontology-driven formalisation process. In section 6 we discuss the aspects related to "connecting people to knowledge" and "connecting knowledge to people". Specifically, we illustrate i) the interface which allows users to access the archive and the associated knowledge base and ii) the *push technology* [32], which alerts journalists to gaps in the archive and readers to new, relevant stories. Finally, in sections 7 and 8 we discuss related work in knowledge management and ontological engineering, we assess the current state of the architecture and we outline outstanding research issues.

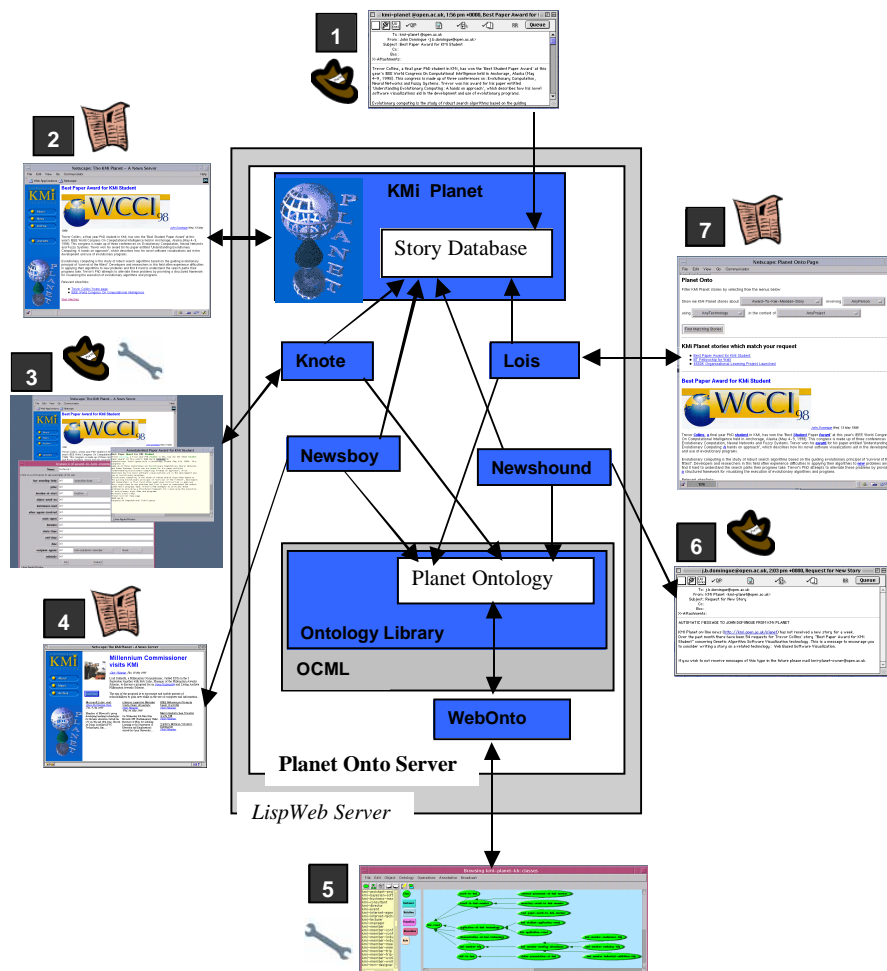## 2   The Architecture of Planet-Onto

The overall scenario introduced in the previous section is graphically shown in figure 1, which summarises both the architecture of Planet-Onto and the associated process model. The architecture of Planet-Onto extends that of the original Planet news server [11]. In the 'basic' scenario supported by the Planet news server a 'journalist', who is normally a KMI member, writes a story about some KMI-related event and emails it to the Planet server. The Planet agent formats the story and adds it to the news archive. Periodically an agent informs the Planet readership about new stories that have been added to the archive. Readers can browse the archive and access stories through a standard web browser, such as Netscape Navigator™ or Microsoft Explorer™.

This basic publish/find out/read scenario supported by KMI Planet has now been augmented in Planet-Onto, as we have developed i) tools which allow the specification and association of knowledge structures to stories, ii) an end user interface, *Lois*, which integrates web browsing with search and knowledge retrieval, and iii) specialised agents which, alert journalists to gaps in the archive and readers to relevant new stories.

Specifically, as shown in figure 1, Planet Onto supports seven main activities with respect to three types of users. These are:

Journalists. Those who send stories to KMI Planet.

Fig. 1. The Architecture of Planet Onto

Knowledge Editors. Those who are responsible for maintaining the Planet Ontology and the Planet knowledge base. In some cases they may annotate the stories in place of journalists.

Readers. Those who read the Planet stories.

The seven main activities supported by Planet-Onto are:

1. *Story submission*. A journalist submits a story to KMI Planet by plain email. The story is formatted and stored within KMI Planet's story database.

2. *Story reading*. A Planet reader browses through the latest stories using a standard browser.
3. *Story annotation*. Either the journalist submitting the story or one of the Planet knowledge editors uses *Knote* to perform ontology-driven association of knowledge structures to a story.
4. *Provision of customised alerts*. An intelligent agent, *NewsBoy*, which is able to build user profiles from patterns of access to Planet-Onto, uses these profiles to alert readers about relevant new stories.
5. *Ontology editing*. The Planet-Onto architecture also includes sophisticated support for knowledge modelling to be used by skilled knowledge editors. In particular, the *Web-Onto* tool [10] provides web-based visualisation, browsing and editing support for developing and maintaining knowledge models specified in *OCML* [27]. The latter is an operational knowledge modelling language, which supports the specification and instantiation of ontologies and the specification and execution of *reusable problem solving methods* [1].
6. *Story soliciting*. An intelligent agent, *NewsHound*, periodically gathers data about 'popular' news items and uses these data to solicit potentially popular stories from the appropriate journalists. This is accomplished by identifying 'gaps' in the knowledge base, e.g. projects about which there is no information, which can be filled by potentially interesting stories.
7. *Story retrieval and query answering*. An end-user interface, *Lois*, integrates traditional web browsing and search with knowledge-based query retrieval, to support integrated access to the archive of Planet stories and to the associated knowledge base.

All the tools included in the Planet-Onto architecture are web-based and can be accessed through ordinary web browsers. As a result, both the development and use of the archive and associated knowledge base are carried out in a distributed fashion, by a community of users playing specialised roles. The underlying technology is provided by an *HTTP server* written in Common Lisp, *LispWeb* [34].

In the rest of the paper we will discuss the Planet-Onto architecture and process model in detail, focusing in particular on the 'external' knowledge management activities and tools rather than on the 'internal' knowledge modelling support and web-based infrastructure.

3   KMI Planet: A Newsroom Agent on the Web

KMI Planet was designed to support the creation of high quality web based newsletters whilst minimising the load for journalists submitting stories. Our approach to attaining this objective was to develop a news server which accepts stories submitted in the lowest common-denominator medium - an email message - yet is able to create a high quality web page. Thus, the system minimises overheads for journalists and editors by making their contributions entirely lightweight: the web based news server takes on most of the work.

In our model a journalist sends an email message to the KMI Planet story account. The subject line of the message becomes the headline of the story, the body of the

message becomes the text of the story. If an image has been attached to the message, it is added into the story in an appropriate place. If no image has been attached, then Planet searches its database of images for a suitable one – e.g. a photo of the journalist in question, the logo of the project described in the story, a screen snapshot of a relevant system, etc.
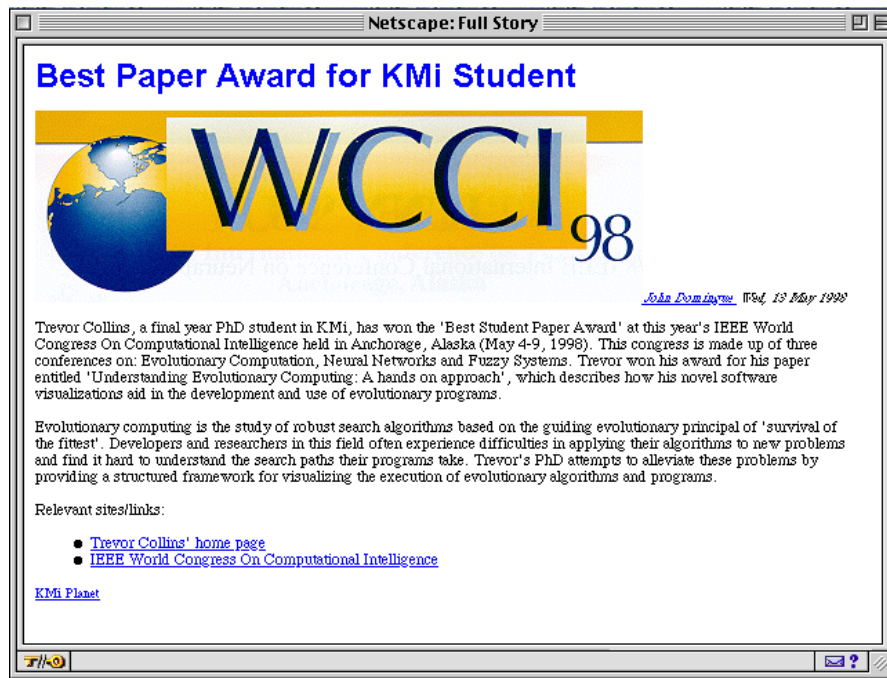
Fig. 2. A story presented in KMI Planet

In figure 2 we see a single story web page rendered from an email message. The message header is shown at the top and the attached image follows beneath. Next to the image are the name of the journalist who sent the message (linked to his home page) and the date of the submission. The message body is below the image, with web links appropriately anchored to their destinations. Although the journalist is responsible for sending in a good story, the rest of the process is handled by the software.

This low-overhead approach to news publishing has proven very successful. Our archive is growing steadily and now contains 73 stories, submitted by 13 journalists. We now have 480 registered readers – i.e. not just readers who have accessed the Planet server but users who subscribe to the Planet alert services. Moreover, the Planet technology has attracted interest from other organisations, both within and outside the Open University. However, it is apparent that, as the Planet archive and readership grow, more sophisticated mechanisms supporting semantic searches and individualised presentations and alerts are needed. In the rest of this paper we will discuss the tools that we are developing to fulfil these needs.

4   An Ontology for Characterizing Academia-Related Events

An ontology provides a partial specification of a shared conceptualisation, to be used for formulating knowledge-level theories about a domain [18] [20] [27]. Our domain comprises the range of events associated with a university department, the persons who take part in these events and the entities required to characterise these events and these persons in the context of academic life.

Several approaches to ontology development have been proposed in the literature, which introduce distinctions along different dimensions. For instance van der Vet and Mars [42] propose a *bottom-up* approach to concept identification, which contrasts with the *top-down* approach normally used by researchers and practitioners – e.g., see [39]. Uschold and Gruninger [41] argue that a *middle-out* 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)[1]. Another distinction is whether an ontology is developed in a *task-oriented* [27] or a *task-independent* style [5] [15]. However, hardly any approach affords detailed, prescriptive guidelines. A notable exception is provided by Uschold and Gruninger [41], who propose a *purpose-driven* skeletal lifecycle for developing ontologies.

In developing the Planet ontology we have followed a task-independent, purpose-driven approach. In particular, the main role of the Planet ontology is to drive the annotation of news items relating events in KMI. Thus, we have taken the concept of *news item* as our starting point. The essence of a news item is that of relating one or more *events*.

4.1   Modelling Events

The notion of event is central to problem solving and several ontological characterisations are available in the literature – e.g., see [25] [37]. Thus, rather than trying to reinvent a pretty complex wheel we looked at existing definitions with the aim of reusing them. In particular, the public version of the CYC upper level ontology [25], which is called *HPKB upper level*[2], is now available on the Stanford ontology server [13]. Thus, we decided it would be a useful exercise to try and reuse this one. Unfortunately this turned out to be a problem. The definition of class event in the UPKB upper level ontology contains 94 slots. Obviously this definition has been designed with the aim to maximise *reusability*, by trying to account for all features which can possibly be associated with an event. On the contrary, given the purpose of our ontology (to allow story annotation by users who are not necessarily knowledge engineers), we are mainly interested in *usability*: that is, our ontology ought to minimise the knowledge engineering overhead associated with story annotation. Otherwise, this task would have to be carried out by specialist knowledge

---

[1] Incidentally, this claim is consistent with much psychological literature, which shows that human subjects are much better are recalling information about basic categories, than they are at recalling information about superordinates or subordinates – see e.g., [35].

[2] HPKB stands for "High Performance Knowledge Bases" and is the name of an ongoing research project in the United States [23].

editors, resulting in an unsustainable approach. In practice, imposing a low overhead on the annotation process means adhering to two modelling guidelines:

- *Minimal ontological commitments*. The definition of class event in the HPKB upper layer provides an extreme case of a *coverage-centred* approach to reuse. That is, the definition aims to cover all potential attributes which *can* be relevant to a *generic instance* of the class. However, typically only few slots will *actually* be relevant for any *specific instance* of the class. An alternative approach consists of *minimising ontological commitments* [19]. That is, to try and provide only the minimal set of attributes needed to define the class. This approach has the advantage that, when populating the ontology, users don't have to face lots of irrelevant attributes.

- *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 [28] and arguably it provides an important difference between the criteria associated with modelling for knowledge acquisition and those associated with modelling for system development.

For these reasons we decided we could not just 'cut & paste' the definition in HPKB, but we needed to build our definitions by means of a more use-oriented approach – i.e. starting with minimalist concepts and then enriching them when defining specialised subclasses. The definition of class event used in our ontology is shown below.

The definition shown in the box defines the essential aspects of an event. For the sake of compatibility the terminology reflects the one used in the HPKB ontology. However, as already pointed out, this generically reusable terminology scores lowly on usability and therefore different refinements of this class provide specialised terminology. For example class award-to-kmi-member renames slot main-agent to awarding-body and slot object-acted-upon to awarded-prize.

In total, KMI-Planet-Ontology comprises 452 definitions, based around 6 epistemological building blocks: story, event, person, organisation, project and technology. We shall describe how the ontology has been deployed within Planet-Onto in the following sections.

```
(def-class event (temporal-thing)
 ((main-agent
   :min-cardinality 1
   :documentation
   "The agents causing the event to happen.
    At least one main agent is assumed but
    there can be others")
  (other-agents-involved
   :documentation
   "Other agents involved in the event")
  (instrument-used :documentation
                   "The instrument used by the main
                    agent to carry out the action")
  (Object-Acted-On
   :documentation "The things which are affected by
                   the event. e.g. in 'john broke the
                   stone with a hammer', the stone is
                   the object acted on")
  (recipient-agents
   :documentation
   "The agents which are affected by the event")
  (location-at-start
   :type location
   :documentation "The location at which an event
                   takes place - or starts in the case
                   of events which change the position
                   of something)")))
```

5   Ontology-driven Story Annotation using Knote

Our goal is to enable as wide an audience as possible to annotate stories. Encoding representations of even a moderately growing repository such as KMI Planet can only succeed if the process is 'farmed out' as much as possible. We thus envisage that users of Knote will not form a homogeneous group, but will range from regular Planet journalists to experienced ontology engineers. 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.

As we discussed in the previous section, the Planet ontology is based around the epistemological tenet that a KMI Planet story describes a number of KMI related events. Story annotation is therefore the description of one or more events occurring within the story to be described. The four main steps in annotating a story are:

1. Choosing a story to annotate,
2. Selecting a particular event in the story to describe,

3. Classifying the event in terms of the hierarchy of event types provided by the KMI-Planet-Ontology.
4. Filling in an automatically created *instance definition form* to characterise the new instance of class event.

Fig. 3. A partially filled in form for an instance of class award-to-kmi-member

Fig. 4. The appearance of figure 3 as the journalist fills in the value of the recipient-agents slot with the instance collins

We shall describe how Knote supports the annotation process through a small scenario. A Planet journalist decides to annotate the 'Best Paper Award for KMI Student' story shown in figure 2. The journalist elects to describe the main event in the story, that is Trevor Collins receiving a best paper award from IEEE. After a little consideration the journalist classifies the event in question as an instance of class award-to-kmi-member and hits the "Describe Event" button. An event instance definition form, partly shown in figures 3 and 4, is created which the journalist begins to fill in. We will illustrate the annotation support provided by Knote by showing how the tool helps the user fill in the slot recipient-agents.

The journalist can see from figure 3 that the value for the recipient-agents slot must be of type kmi-member. She decides to see which more specific types are currently available by clicking on the kmi-member menu. After choosing class kmi-phd-student, the journalist checks whether there are any instances of kmi-phd-student currently defined. She does this by clicking on the rightmost menu of the recipient-agents row – see figure 4. She chooses collins from the menu and the text "collins" is inserted into the recipient-agents value window. Alternatively, the journalist could have chosen to create a new instance of class kmi-phd-student, by clicking on the

menu item "New Instance". In this case a new instance form would be created and the name of the new instance inserted as the value of slot recipient-agents.

The instance forms described here are modelled on the *Dynamic Forms* of Girgensohn et al. [17] and provide a subset of the functionalities found in the Dynamic Forms system. The key difference between dynamic forms and instance forms in Knote, however, is that forms in Knote are generated directly from the ontology and not from a user description. The forms in Knote are similar to the forms found in the Mecano environment [31].

6   Connecting People to Knowledge

6.1   Lois: A Flexible Form-Based Interface for Knowledge Retrieval

As the number of stories in KMI Planet grows, it becomes harder for users to find relevant stories quickly. In addition, browsing and reading stories is but one way to find information about events in KMI. Readers may wish to know about specific technologies, specific projects or specific members of staff. For instance, after reading the story about the award to Trevor Collins, a reader might want to find out who else in KMI works on software visualisation, what papers have recently been produced, what projects tackle software visualisation issues, etc. An important feature of the Planet-Onto architecture is the integration of traditional web browsing, including lexical search, with deductive knowledge retrieval. In particular, various levels of knowledge retrieval support are provided. Experienced ontology engineers can directly access the Planet knowledge base and pose arbitrary queries expressed in OCML, using the Web-Onto tool [10]. However, our assumption is that most readers either are not experienced knowledge engineers, or, even if they are, might not want to interact with Planet-Onto at the OCML level.

To support semantic access to Planet by 'ordinary' readers we have developed a form-based interface, called Lois. The aim of this interface is to allow users to express a wide range of queries, ideally any query that can be expressed directly in OCML, while at the same time shielding them from formalism-related aspects.

The solution we have taken is to use the basic epistemological building blocks (people, organisations, stories, events, projects and technologies) of the Planet knowledge base to organise a form-based query interface. The rationale for this approach is that, almost without exception, any useful query to Planet-Onto must include one or more subclasses or instances of these six building blocks. For instance, figure 5 shows a query which asks for KMI researchers involved in software visualisation. This query was constructed by selecting the class kmi-member (pressing the button "Member of KMi"), specialising it to kmi-researcher (using the "Index Aspect" and "Aspect Type" windows), selecting the relation develops-technology and then circumscribing the range of this relation to kmi-software-visualisation-technology. To ensure that 'naïve' users can indeed construct queries out of these six building blocks, when designing the ontology we have ensured that all the 'obvious' binary relations between these six classes are *explicitly* included in the ontology (as opposed to be derivable through chains of inferences). Because stories

are only interesting with respect to the associated events, they are not linked to the other main classes (and correspondingly no button is provided for the story class). Hence, only 24 relations had to be specified.
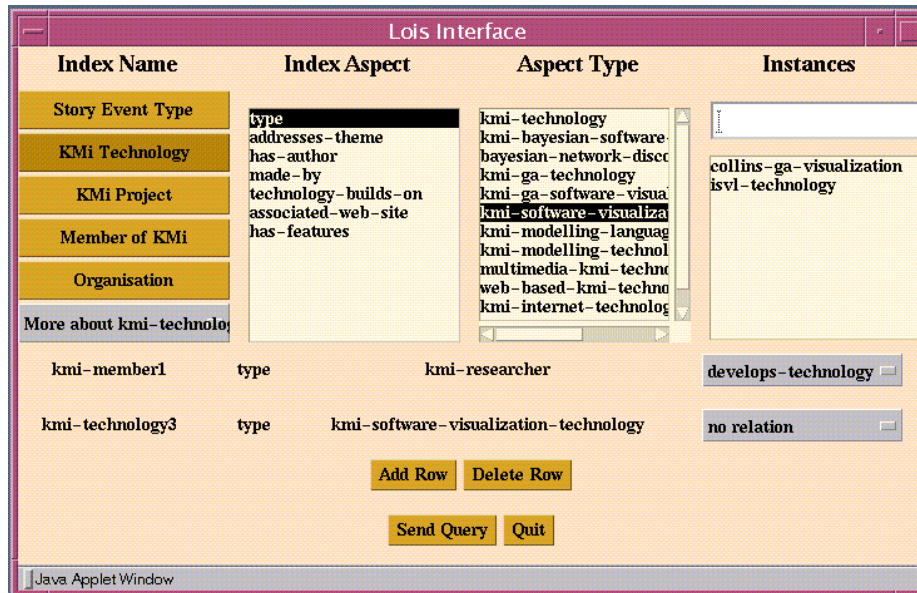

Fig. 5. Finding a KMI researcher who works on software visualisation

6.2   Story Chasing with NewsHound

An important goal in the design of KMI Planet was that 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. In order to emulate this behaviour we are developing an intelligent agent, NewsHound, whose job is to identify potentially popular stories and to assign them to the appropriate journalists. These requests for new stories are carried out simply by sending the relevant journalist an email.

   In order to identify potentially interesting stories, NewsHound uses 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 own 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 uses 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. However, in our view the most interesting feature of NewsHound has not so much to do with the specific adopted heuristics but rather 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 [22]. In our scenario incompleteness is defined in pragmatics terms: publications need popular stories.

6.3   Providing Personalised Alerts with NewsBoy

Lois is 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 therefore designed an agent, NewsBoy, to provide a mode of use that was 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 annotated using Knote, NewsBoy matches the story knowledge base against the specified interests of registered readers. Readers whose interests match that of the newly annotated story are notified that a new story has been added to their personal Planet page. For example, if a reader has specified that they would like to read stories about visitors to KMI, she would receive an email stating that her personal Planet page has been updated to look like figure 6.

The primary interface to NewsBoy is via a settings page where registered users can:

1. Set their name and password.
2. Set the criteria for a notification to be sent.
3. Declare what sort of stories their interests cover.

Passwords enable groups of users to create a secure communal newsletter where confidential information can be communicated. Readers can elect to be sent an automatic notification by NewsBoy every time a relevant story is found, or to be sent an update every week or month. Moreover, readers can either explicitly declare the types of stories which interest them or allow NewsBoy to infer them. One of the prime goals when designing Planet-Onto was to make the system as easy to use as possible. For this reason we decided to reuse the Lois interface as far as possible, rather than creating a whole new interface for readers to communicate their interests to NewsBoy.

To make an explicit declaration a reader simply specifies a number of queries using the Lois interface. The reader is then informed 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 – see e.g., [26] [24]. 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.

Fig. 6. A personalised Planet Web page showing stories concerning visits to KMI

7 Related work

The work described in this paper is related to research in a number of areas, including *information retrieval* [38] and *extraction* [33] [9], *knowledge management* [29], *ontological engineering* [18], *agent technology* [7] *knowledge engineering* [14] and *model-based knowledge acquisition* [22] [36] [12]. Because we have already compared individual tools in Planet-Onto to related technology and (more importantly) because this paper should not exceed the 18 page limit, we will confine ourselves to discuss related approaches to ontology-driven knowledge management.

The SHOE project [21] has proposed an extension to HTML to allow the specification of ontological information in web pages. 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. In Planet-Onto we take a holistic approach to the publish-annotate-retrieve process 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 publish-annotate-access process. 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 also shares a number of commonalities 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 [16]. In this paper we have emphasised that the feasibility of the idea of a collective construction of a knowledge base crucially depends on the availability of i) a carefully defined ontology; ii) an underlying modelling language providing user-oriented facilities, such as context-dependent renaming; iii) a user-friendly annotation environment; and iv) the right motivational stimuli for the participants. In their paper on the (KA)2 initiative, [2], the authors mainly focus on the latter issue. However it seems to us that a careful analysis of all the issues associated with collaborative ontology development and instantiation 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 is centralised in our approach. Members of the community are not expected to develop ontologies, only to populate existing ones.

Related work in knowledge management here in KMI has (naturally) many points of contact with Planet-Onto. The work in the *Enrich* project [40] aims to support organisational learning through the enrichment of web-based documents. This enrichment is carried out both through ontologies, as in our scenario, and through hypertext-based argumentation [4]. Although the Enrich scenario is very different from the one addressed by Planet-Onto, the underlying assumptions are the same for both projects: ontology-driven enrichment of documents play an important role in knowledge management. However, effective support for knowledge management requires a holistic approach, which carefully analyses both technological and organisational issues and emphasises the usability of the deployed technology and the sustainability of the overall process model.

Buckingham-Shum and Sumner [3] have produced an ontology to support the tracking of research within the Journal of Interactive Media in Education (http://www-jime.open.ac.uk/). Their emphasis is to support the development of shared viewpoints in a community and the discovery of relationships between documents. As for Planet-Onto populating the ontology is a collaborative process. In contrast with Planet-Onto only an informal ontology is provided.

8   Conclusions

The Planet-Onto architecture provides an example of an ontology-centred approach to knowledge management. With the exception of NewsBoy and NewsHound, which are still at a preliminary implementation stage, all components of the architecture are now in place and are undergoing preliminary user testing. Once this preliminary evaluation and testing phase has been completed, Planet-Onto will become fully operational. We also plan to apply the Planet-Onto technology to support semantic access to medical guidelines. This work will be done in the context of the *PatMan* project [30].

Obviously, several research issues are still open. In particular, the main obstacle to this kind of enterprises is provided by the collaborative construction of the knowledge base. While usability has been our main criterion when designing the ontology and the knowledge annotation tool, we realise that this approach is only sustainable in restricted scenarios, where users are reasonably motivated and skilled and the annotation process relatively lightweight. Thus, the major challenge for this and other similar enterprises remains to develop tools that take the burden of ontology annotation off the writers of the documents to be annotated. Investigating the feasibility of such tools will be one of our main research goals for the near future.

Acknowledgements

References

1.   Benjamins, V. R. and Fensel, D.: Special Issue on Problem Solving Methods. International Journal of Human-Computer Studies, **49**(4), (1998) 305-650

2 .   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 (1998)

3.   Buckingham-Shum, S. and Sumner, T.: Publishing, Interpreting and Negotiating Scholarly Hypertexts: Evolution of an Approach and Toolkit. Technical Report KMI-TR-57, Knowledge Media Institute, The Open University, Milton Keynes, UK (1997. Available from http://kmi.open.ac.uk/kmi-abstracts/kmi-tr-57-abstract.html.

4 .   Buckingham-Shum, S. and Sumner, T.: New Scenarios in Scholarly Publishing and Debate. In M. Eisenstadt, and T. Vincent, (editors) The Knowledge Web: Learning and Collaborating on the Net, Kogan Press, (1998) 135-152

5.   Beys, P., Benjamins, R., and Van Heijst, G.: Remedying the Reusability-Usability Trade-off for Problem-Solving Methods. In B. R. Gaines and M. Musen (editors), Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW´96), Banff, Canada, (1996)

6. Bowker, G. C.: Lest We Remember: Organizational Forgetting and the Production of Knowledge. Accounting, Mangement and Information Technologies, **7**(3) (1997) 113-118. Available from http://www.lis.uiuc.edu/~bowker/forget.html

7. Bradshaw, J.: An Introduction to Software Agents. In J. Bradshaw (editor), Software Agents. AAAI Press/MIT Press, Menlo Park, California (1996)

8. Clark, P., Healy, M., Uschold, M., Williamson, K. and Woods, S.: An Experiment in Ontology Reuse. In B. Gaines and M. Musen (editors), Proceedings of the 11th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada (1998)

9. Craven, M., Di Pasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K. and Slattery, S.: Learning to Extract Symbolic Knowledge from the World Wide Web. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98) (1998)

10. Domingue, J.: 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 (1998).
Available from http://kmi.open.ac.uk/people/domingue/banff98-paper/domingue.html

11. Domingue, J. and Scott, P.: 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, (1998) 173-184

12. Eriksson, H., Puerta, A. R. and Musen, M. A.: Generation of Knowledge Acquisition Tools from Domain Ontologies. In B. Gaines and M. Musen (editors), Proceedings of the 8th Knowledge Acquisition for Knowledge-Based Systems Workshop (1994)

13. Farquhar, A., Fikes, R. and Rice, J.: The Ontolingua Server: A Tool for Collaborative Ontology Construction. In B. Gaines and M. Musen (editors), Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop. Banff, Alberta, Canada (1996)

14. Feigenbaum, E. A.: The Art of Artificial Intelligence: Themes and Case Studies of Knowledge Engineering. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA (1977)

15. Fensel, D., Motta, E., Decker, S. and Zdrahal, Z.: The Use of Ontologies for Specifying Tasks and Problem Solving Methods: A Case Study. In R. Benjamins and E. Plaza (editors), Knowledge Acquisition, Modeling, and Management. Proceedings of the 10th European Workshop, EKAW '97. Lecture Notes in Artificial Intelligence 1319, Springer-Verlag (1997)

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

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

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

19. Gruber, T. R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-Computer Studies **43**(5/6) (1995) 907-928

20. Guarino, N. and Giaretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars (editor), Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing. IOS Press, Amsterdam (1995) 25-32

21. 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 (1998). Available from http://www.cs.umd.edu/projects/plus/SHOE/shoe-aaai98.ps

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

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

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

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

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

27. Motta E.: Reusable Components for Knowledge Models. PhD Thesis. Knowledge Media Institute. The Open University. UK (1997). Available from http://kmi.open.ac.uk/~enrico/thesis/thesis.html

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

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

30. Patman: Patient Management Workflow Systems. Telematics Applications Project HC 4017 (1998). Available from http://aim.unipv.it/projects/patman/

31. Puerta, A. R., Eriksson, H., Gennari, J. H. and. Musen, M. A.: Beyond Data Models for Automated User Interface Generation. Proceedings of the HCI'94, People and Computers, The University of Glasgow, Knowledge Systems Laboratory (1994)

32. PointCast: http://www.pointcast.com/ (1998)

33. Riloff, E.: An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. AI Journal **85** (1996)

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

35. Rosch, E.: Principles of Categorisation. In E. Rosch and B. B. Lloyd (editors), Cognition and categorisation. Lawrence Erlbaum Associates, Hillsdale, NJ (1978)

36. Runkel, J. T., Birmingham, W. B. and Balkany, A.: Separation of Knowledge: a Key to Reusability. Proceedings of the 8th Banff Knowledge Acquisition Workshop. Banff, Canada (1994)

37. Schank, R. C.: Conceptual Dependency: A theory of natural language understanding. Cognitive Psychology, **3** (1972) 552-631

38. Schatz, B. R.: Information Retrieval in Digital Libraries: Bringing Search to the Net. Science **275** (1997) 327-334

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

40. Sumner, T., Domingue, J., Zdrahal, Z., Hatala, M., Millican, A., Murray, J., Hinkelmann, K., Bernardi, A., Weiss, S. and Traphoner, R.: 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 (1998)

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

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