myPlanet: an ontology-driven Web-based personalised news service

Yannis Kalfoglou, John Domingue, Enrico Motta,
Maria Vargas-Vera, Simon Buckingham Shum
Knowledge Media Institute(KMi),
The Open University,
Milton Keynes MK7 6AA, UK
{y.kalfoglou,j.b.domingue, e.motta, m.vargas-vera, s.buckingham.shum}@open.ac.uk

Abstract

In this paper we present *myPlanet*, an ontology-driven personalised Web-based service. We extended the existing infrastructure of the **PlanetOnto** news publishing system. Our concerns were mainly to provide lightweight means for ontology maintenance and ease the access to repositories of news items, a rich resource for information sharing. We reason about the information being shared by providing an ontology-driven interest-profiling tool which enable users to specify their interests. We also developed ontology-driven heuristics to find news items related to users' interests. This paper argues for the role of ontology-driven personalised Web-based services in information sharing.

1 Introduction

Nowadays, we observe a trend in providing personalized Web-based services in order to accommodate the versatile needs of an ever increasing number of Web users. Recent advances in agent and Internet technology provide the technological means, however, equally important is to provide the means for semantic infrastructure. Towards this goal, [Huhns and Stephens, 1999] propose the use of "personal ontologies" where each Web user will be able to create his/her own ontology tailored to his/her view of the world. Although we found this idea fruitful, it bears a contradictory connotation. When we talk about ontologies, we can't really say "personal". Ontologies are - by definition - shared views of the world([Kalfoglou, 2000a]). We rather prefer to use the metaphor "personal views" of an ontology tailored to specific services. That is, each user will see - and eventually be able to edit - part of an ontology that is tailored to a specific service. The ontology itself will remain shared, in the sense that the creation, editing and maintenance tasks involve the efforts of many agents(let them be people or software). The way it will be exposed to users will depend on the kind of services they want. For example, in our domain of Web-based news services, a user is able to browse those contents of the ontology that are related to news items, like people who wrote them, projects mentioned, etc. This kind of Web-based news services enable users to access information tailored to their interests.

Valuable information is shared among the members of a community by using the lowest-common-denominator medium: an email message. Users send a story in the form of an email(hereafter, e-Story) to a news server from which designated systems redirect the e-Story back to targeted members of the community. This is an indirect form of communication(in comparison with a member-to-member form), however, we enrich it by an ontology-driven interest-profiling tool and deductive knowledge retrieval techniques. This allowed us to reason about the knowledge being shared and target it to certain people. The means for connecting knowledge to people were analyzed from the process point of view in [O'Leary, 1998]. His framework has been used in some ontology applications([Kalfoglou, 2000b]) and in [Domingue and Motta, 2000] the authors showed how these processes are realized in the context of **PlanetOnto**. In particular they focussed on the two connecting processes: *people to knowledge* and *knowledge to people*. The means which were used to connect people to knowledge in **PlanetOnto** were integrated visualisation, search, and query-answering facilities whereas the connection of knowledge to people achieved by pro-actively contacting people to solicit e-Stories and alert them when items of interest were published.

To deliver such an ontology-driven service we need to have flexible mechanisms for ontology maintenance, an area which is still in its infancy and hampers ontology applications([Kalfoglou *et al.*, 2000]). In this work, we deployed Information Extraction(hereafter, IE) systems to extract information from users using a service which could be used to update the underlying ontology. In that sense, the user becomes the main agent responsible for maintaining the ontology instances, lifting the burder from ontological engineers who can focus on structural and semantic issues related with ontology design and deployment. In our domain we experimented with extracting information from users' e-Stories in order to update the underlying ontology.

Our research goals are two-fold: (a) to improve and ease ontology usability for Web users by means of ontology-driven Web-based front-ends to personalized services; and (b) to provide lightweight means for ontology maintenance triggered by users' input by deploying IE techniques along with domain specific templates. This tight coupling of Web-based environments with underlying ontologies is a promising and appealing technology for the majority of users.

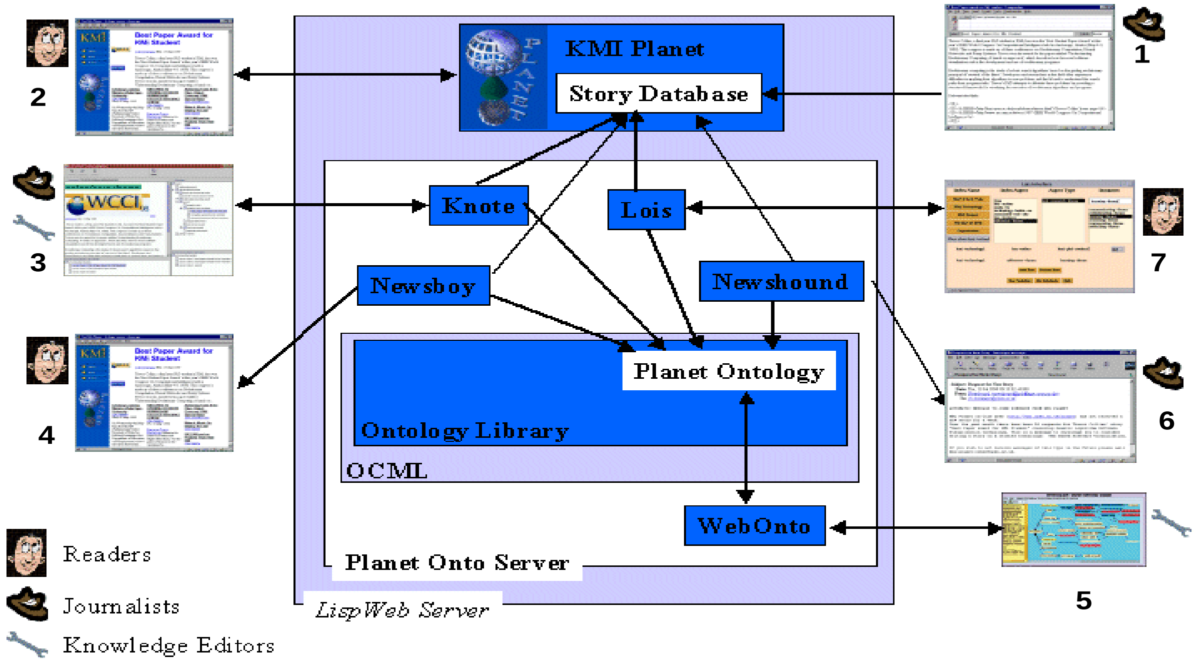

Figure 1: The **PlanetOnto** architecture.

We organize this paper as follows: in section 2 we describe the existing infrastructure, **PlanetOnto**, which we extend in section 3 with the personalized services provided by *myPlanet*. We report on related efforts in section 4 and we conclude the paper in section 5 by discussing future directions and implications of this work.

2 PlanetOnto

In this section we briefly describe the existing infrastructure, **PlanetOnto**, an integrated suite of tools developed over the last 4 years in the *Knowledge Media Institute(KMi)*. The whole infrastructure is described in detail in [Domingue and Motta, 2000]. Here we recapitulate on the important elements of the **PlanetOnto** architecture some of which were the focus of our work as we describe in the next section.

In the **PlanetOnto** domain we identify three types of users: journalists who send stories to *KMI Planet*, knowledge editors who maintain the *Planet* ontology and the *Planet* knowledge base, and readers who read the *Planet* stories. In figure 1, we illustrate the **PlanetOnto** architecture along with the activities that supports:

1. *Story submission:* Stories are submitted to *KMi Planet* in the form of email which is then formatted and stored in *KMi Planet's* story database;
2. *Story reading:* Stories can be read by using a standard Web browser;
3. *Story annotation:* A specialized tool, *KNote*, is used to help the journalist or the knowledge editor to associate the story with knowledge structure of the underlying ontology. This process was manual and we have semi-automated it as we describe in section 3.2;
4. *Provision of customized alerts:* An agent, *Newsboy*, builds user profiles from patterns of access to *PlanetOnto* and then uses these profiles to alert readers about relevant stories. While that tool uses statistical evidence to build profiles, in section 3.1 we present *myPlanet* which makes it possible for a user to build a profile by using an ontology-drawn structure;
5. *Ontology editing:* A Web-based ontology editor, *WebOnto*[Domingue, 1998], is used for constructing knowledge models in the OCML language[Motta, 1999];
6. *Story soliciting:* An agent, *Newshound*, gathers data about popular news items and then solicits potentially popular stories from the journalists. The ontology-driven heuristics of *myPlanet*, described in section 3, could extend this tool to solicit stories from journalists with similar interests;
7. *Story retrieval and query answering:* A Web-based interface, *Lois*, provides access to the story archive and the associated knowledge base by integrating Web-browsing and search with knowledge-based query retrieval.

3 *myPlanet*

PlanetOnto was originally conceived as an internal newslet-

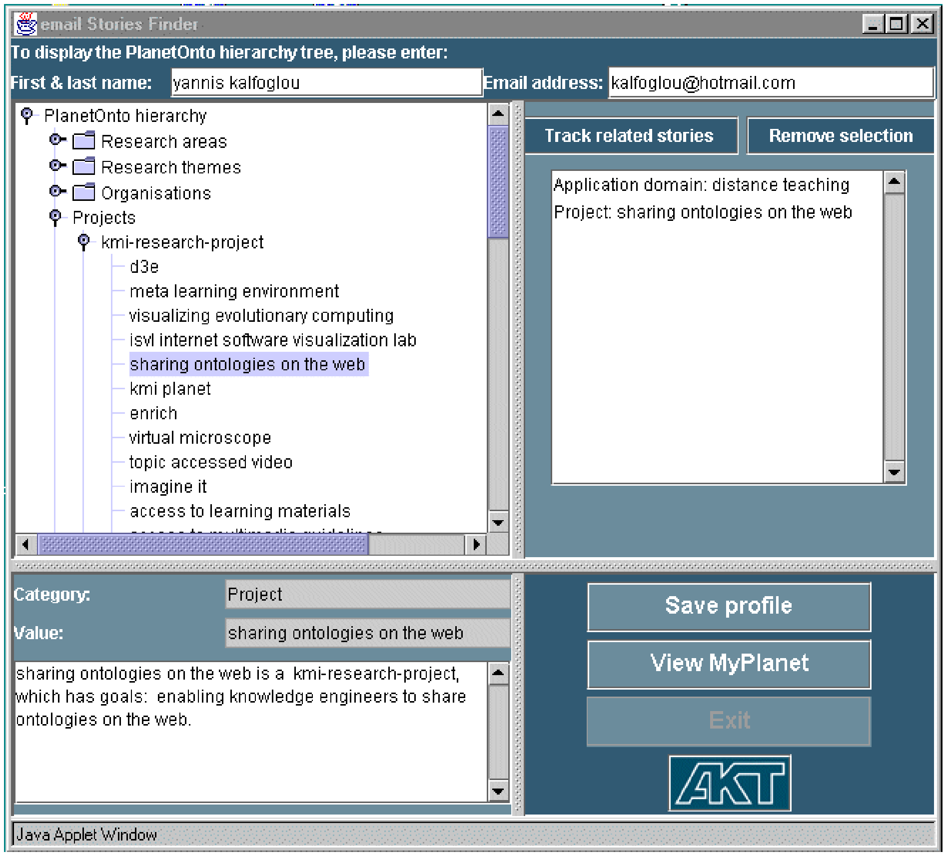

Figure 2: The e-Stories finder Java Applet.

ter and progressively became an integrated suite of tools for knowledge management. It is used as a mass communication medium from members of our lab but lacks the advantages of personalised, tailored-to-preferences, services. *myPlanet* aims to fill-in this gap by providing the means for easy navigation through the e-Stories repository, setting user preferences, and providing assistance to the knowledge editors for annotating e-Stories. We describe these tools in the following two sections.

3.1 Ontological interest-profiling

One of the limitations of the **PlanetOnto** suite of tools was the lack of an e-Stories retrieval method which would enable users to read only the e-Stories of their interest instead of forcing them to browse the e-Stories database for potentially interesting items. A possible fix to this problem would have been to provide a keyword-based search engine. This sort of solution, however, bears the known limitations that everyone of us has experienced with current keyword-based search engines(e.g, unrelated matches).

Consequently, we worked on a method which allows the user to specify his/her interests(crudely speaking, "the search criteria"), and then we search for e-Stories that match these interests. The difference of our approach when comparing it with a keyword-based search engine is that the structure of the interests is drawn from the underlying ontology. Hence, we deliberately impose a generic structure of interests to the user which contains the most important types of information one would typically find in the *KMi Planet* e-Stories. This structure is composed of the following items:

- `Research areas` that are investigated in KMi;
- `Research themes` that are investigated in KMi;
- `Organizations` that KMi collaborates with;
- `Projects` in KMi;
- `Technologies` used in KMi;
- `Application domains` that are investigated in KMi;
- `People` - members of the KMi lab.

All of these items are classes in the underlying `KMi Planet` ontology[1]. The advantage of this is that we can go beyond the expected category name matching: we can

[1] Accessible from the Web through the `WebOnto` browser on URL: http://webonto.open.ac.uk/

reason about the categories selected by applying ontology-driven deductive heuristics. For example, if someone is interested in `Research Area` *Genetic Algorithms*, we would normally return all the e-Stories that talk about that `Research Area` by employing the string-matching technique we describe in the sequel. However, by using the ontological relations that hold between these categories we can find which `Projects` have as `Research Area` *Genetic Algorithms* and then search for e-Stories that talk about these `Projects`. These would then be included in our answer set as potentially interesting e-Stories although they don't explicitly mention the *Genetic Algorithms* `Research Area`. In the same manner, we can apply more complex heuristics such as finding `Technologies` that have been used in `Projects` and `People` who are members or leaders of these `Projects` - which have as `Research Area` *Genetic Algorithms* - therefore inferring that these `People` might be a potential contact for information on `Technologies` for *Genetic Algorithms*. In terms of the underlying ontology structure, our aim is to take advantage of the rich definitions of classes in the OCML language. For example, the following OCML code is the definition of an instance of a KMi research and development project, the "sharing ontologies on the web" project:

(def-instance project-sharing-ontologies-on-the-web **kmi-r&d-project**
((**has-research-area**
res-area-ontologies res-area-knowledge-sharing-and-reuse)
(**project-application-domain** organisational-learning)
(**addresses-theme**
theme-collaborating theme-communicating theme-reasoning)
(**has-project-leader**
john-domingue enrico-motta zdenek-zdrahal)
(**funding-source** org-european-commission)
(**has-goals**
"Enabling knowledge engineers to share ontologies on the web.")
(**has-web-address**
web-page-project-sharing-ontologies-on-the-web)
(**uses-technology** lisp java tech-lispweb tech-ocml)
(**associated-products** tech-webonto tech-tadzebao)))

As we can see, this definition is sufficient for deducing facts related to the project's research areas, themes, application domain, leaders, etc. Most of these constructs are used directly in the browsable structure we imposed to the user in *myPlanet*'s interface. Thus, the deduction step involves a straightforward OCML query. Other slots, however, like funding source and technologies used, can be used to infer further links as in the scenario we described before. This rich representation of a project instance highlights the strengths of OCML as a knowledge modelling language([Motta, 1999]) which has been used in many projects over the last 6 years. Currently, there are over 90 models defined in the WebOnto library all of which are accessible with a Web browser from `webonto.open.ac.uk`. We also use relations to link people with projects such as:

```
(def-relation involved-in-projects (?x ?project)
   :constraint (and (person ?x)
                    (project ?project))
   :sufficient (or (has-project-member ?project ?x)
                   (has-project-leader ?project ?X)))
```

The OCML language provides support for defining operational options for each relation such as the `:sufficient` construct in our example above. Its purpose is to help characterize the extension of a relation. For the relation given above, it is sufficient to prove that a person is a member or leader of a project in order for the relation `involved-in-project/2` to hold. We also store the selections a user makes, that is, we save the user's profile with respect to the selected interests. This profile can be edited later on as well as used for finding pro-actively e-Stories that match it.

The matching of interests in a e-Story is based on string-matching but employs the notion of "cue phrases" and "cue words" which are associated with the instances of the categories given above. We use two meanings of "cue": evidence and abstraction. A cue phrase, in our approach, is both an abstraction of the category that is associated with and evidence that the e-Story which contains it is relevant to that category. For example, we define as a cue phrase for the `Research Area` *Ontologies*, the phrase "knowledge sharing and reuse". This is an abstraction of the term *Ontologies*. Whenever we find that phrase in an e-Story we assume that this e-Story is relevant to *Ontologies*. This finding is the evidence of relevance. This technique has been proved easy to apply and gave us a broader and more accurate answer set than the one we would get with a simple match of the category name. On the other hand, we need to be careful when we identify or devise cues for a particular category since a loosely defined cue phrase could result in loosely related e-Stories. For example, the cue phrase "survival of the fittest" could be argued that is an abstraction of the *Genetic algorithms* `Research Area` since it describes a common technique of molecular biology used in *Genetic algorithms*. It might be dangerous to use it though, since it is loosely connected to the term *Genetic algorithms* and the possibility to get unrelated e-Stories is high(e.g, e-Stories about a fighting contest might contain this phrase). We see this as a tradeoff: the more generic the cue phrases are the more phrases we can define or devise, the less generic the cue phrases are the less phrases we can define or devise. It is obvious that, with more cue phrases we can find more e-Stories but the phrases can't be too generic because this may result in unrelated e-Stories. To resolve this tradeoff, we had to follow a manual approach in identifying or even devising, whenever necessary, cue phrases for all the instances of the seven categories described above. That way, we were able to judge by ourselves the "closeness" of a cue phrase to a particular category by referring to literature resources, asking experts in that category for advice, etc. We are planning, however, to automate this process to the maximum degree possible as this is a desired requirement in order to scale-up this approach in a time-effective manner.

To illustrate the usage of this tool, we will go through a detailed scenario in which a user tries to find e-Stories related to his/her interests. As we can see from figure 2, a Java Applet is used as the front-end for choosing the categories upon which the search will be based. When this Applet is loaded over the Web it loads all the instances for the seven categories given above, hence it provides a partial view of the underlying ontology's contents. In our example, the user "yannis kalfoglou" has browse the hierarchy tree and chosen two categories: `Application Domain` *Distance teaching* and `Project` *Sharing Ontologies on the Web*. These two

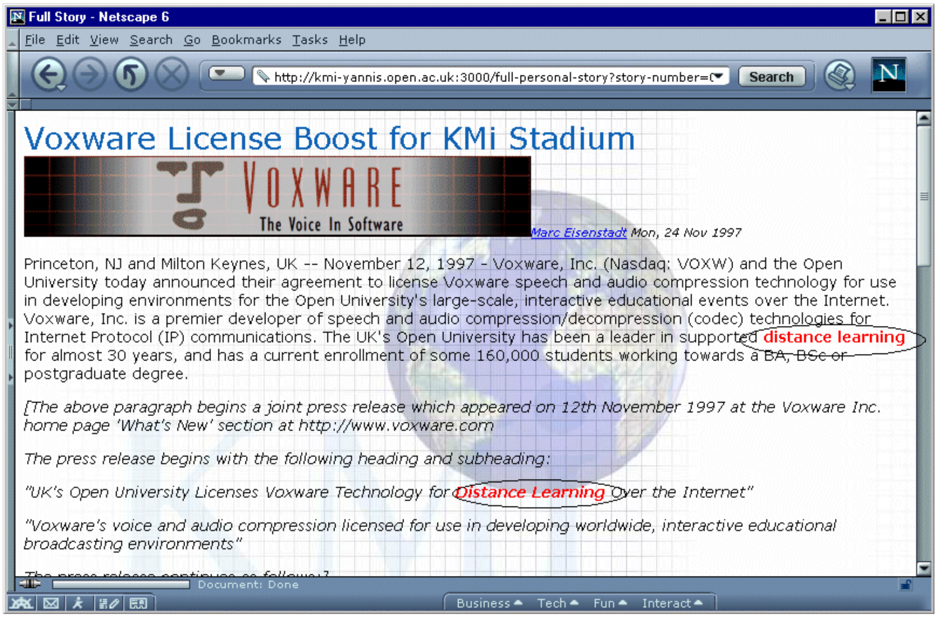

Figure 3: A e-Story of *myPlanet*.

are displayed in the upper right pane of the window in figure 2. The lower left pane is used for displaying additional information with respect to the category currently viewed in the tree. In our example, we see a textual description of the goals for the `Project` being viewed. This information is obtained by querying the underlying ontology for the project's goals. We display different types of textual information tailored to the type of category being viewed. For example, when an instance of `People` is viewed then we display the projects that this person is involved to. This information is obtained from the ontology after firing the relevant query.

After selecting the categories, user "yannis kalfoglou" can save his profile and initiate the search by pressing the `View myPlanet` button. This will display the results, if any, in a personalized Web-page which will be used in future sessions as the user's personal `Planet` Web-page(hence, *myPlanet*). Such a page contains the set of e-Stories that match the selected categories by employing the string-matching technique we described above. We include a snapshot of a e-Story that was found relevant to the user's interests in figure 3. As we can see, this e-Story contains the cue phrase "distance learning"(which is deliberately circled for the sake of this example) which is associated with the `Application domain` *Distance teaching*.

3.2 Populating the ontology

The e-Stories are formalized in terms of associating them with a formal representation which supports various forms of reasoning in **PlanetOnto**. This formalization process, as [Domingue and Motta, 2000] describe:

> "is driven by an ontology that defines the concepts needed to describe events related to academic life - for example, projects, products, seminars, publications and so forth. This means that we ignore parts of a news story that are not relevant to the ontology, much as in template-driven information extraction approaches."

In these approaches, IE systems focus only on portions of text that are relevant to a particular domain. From that perspective, IE can be seen as the task of pulling pre-defined relations from texts as we see in applications of IE in various domains(see, for example, [Proux and Chenevoy, 1997]). Furthermore, IE can be used to partially parse a piece of text in order to recognise syntactic constructs without the need of generating a complete parse tree for each sentence. This approach could be coupled with domain specific templates in order to identify relevant information. If no extraction template applies to the parsed sentence then no information is retrieved.

These characteristics of IE technology were appealing for our task: to populate the ontology with new instances of e-Stories in an automated manner. IE gave us the means to identify the part of an e-Story that will be processed, whereas domain specific templates made it possible to fill-in slots in ontology instances. For example, in a e-Story for the KMi domain one might be interested to extract only the name of KMi projects, KMi members, KMi funding organisations, KMi award bodies, money being awarded, etc., and ignore the rest. As it is described in [Vargas-Vera *et al.*, 2001], the kind of information that will be extracted is determined by the pre-defined templates which are based on the typology of events in our `KMi Planet` ontology. Examples of events are `visiting-a-place-or-people`, `academic-`

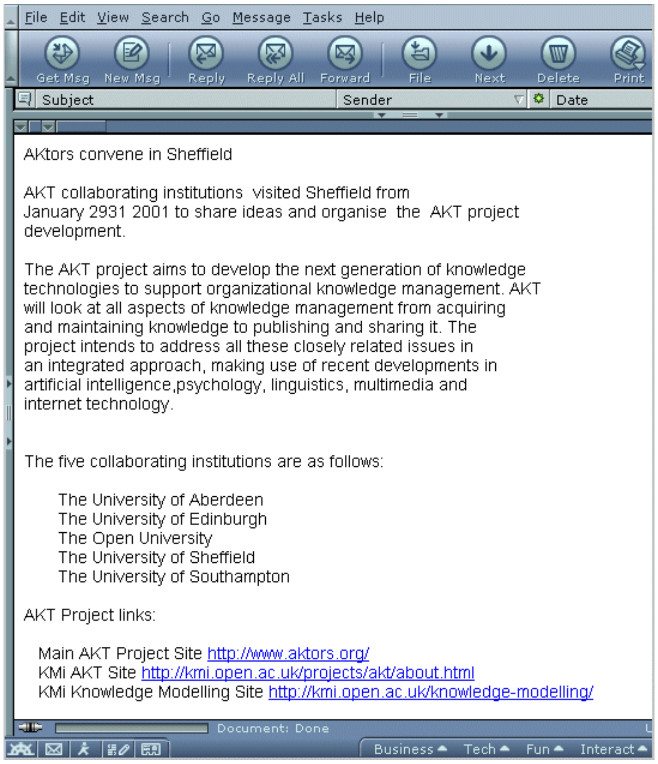

Figure 4: A e-Story send to KMi Planet.

`conference`, `event-involving-project`, and so forth. Currently, we have 40 event types defined in our ontology and we have devise templates for 10 of them. These are the domain specific templates used in IE systems.

An example template for the event type `visiting-a-place-or-people` is as follows:

`[_,X,_,`*`visited`*`,Y,`*`from`*`, Z,_]`

This template matches the sentence word list where X is recognisable as an entity capable of visiting, Y is the place being visited and cannot be a preposition, and Z is recognisable as a range of dates by virtue of their syntactic features. The remaining tokens in the sentence are ignored. We use the underlying `kmi-ontology` instances to identify proper names for visitors(if they are KMi employees) and whenever this fails we deploy a named entity recogniser to help us with identifying additional proper names for visitors and places. Each template is triggered by the main verb in any tense. In this template, the trigger word is the verb “visited”. As [Riloff, 1996] describes, linguistic rules could be deployed to help identify trigger words reliably. For example, if the targeted information is the subject or the direct object of a verb then the trigger word should be the main verb.

Assume that a KMi journalist submits a e-Story about an AKT meeting. We illustrate such a e-Story in figure 4. As we can see, the first sentence of the e-Story matches the template given above. It contains the trigger word “visited”. This will activate the template and variables X, Y, and Z will be instantiated to visitor, place being visited and range of dates, which give us the following information:

- visitor: “AKT collaborating institutions”
- place: “Sheffield”
- date: “January 29-31 2001”

This will be automatically converted to OCML code in order to fill-in the slots in the instance of the event type we are dealing with:

(def-instance visit-of-akt-collaborating-institutions
  visiting-a-place-or-people
   ((has-duration ‘3 days’)
    (start-time january-29-2001)
    (end-time january-31-2001)
    (has-location sheffield)
    (visitor akt-collaborating-institutions)))

In the sequel, a form-based interface is used to visualize the information extracted as shown in figure 5. Uninstantiated slots could be filled-in manually by the knowledge engineer. The main help of this semi-automatic instantiation of event type is the extraction of information from e-Stories, the partial slots-filling, and the identification of event type.

In some templates we can also make use of the underlying ontology to support the event identification. For example, the template for the `conferring-a-monetary-award` event type is:

`[X,_,`*`has been awarded`*`,Y,`*`from`*`,Z,_]`

where Y is amount of money, Z is a funding body, and X is either a person of a project. To decide which one, we traverse the instances of people and projects in the underlying `kmi-ontology` to find out which matches X.

4 Related work

Although we couldn’t find directly comparable projects with our domain - ontology-driven Web-based personalized news services - there several efforts described in the literature where ontologies and Web-based services were put together. We report on these in the sequel:

In the *FindUR* project[McGuinness, 1998], the means for knowledge-enhanced search by using ontologies were investigated. McGuinness describes a tool, deployed at the AT&T research labs, which uses ontologies to improve the search experiences from the perspectives of recall and precision as well as ease of query formation. Their tool is mainly targeted to the Information Retrieval research area and aims to improve the search engines technology. However, the idea of deploying ontologies to achieve these goals is similar to our approach which is mostly concerned with using ontologies to structure the search space(i.e., pre-selected categories of interests - section 3.1) and increase the answer set(i.e., heuristics deployed to select a relevant e-Story - section 3.1). In their work though, means for updating the topic sets used to categorize information(similar to our interests categories) were investigated. In contrast with our approach where the

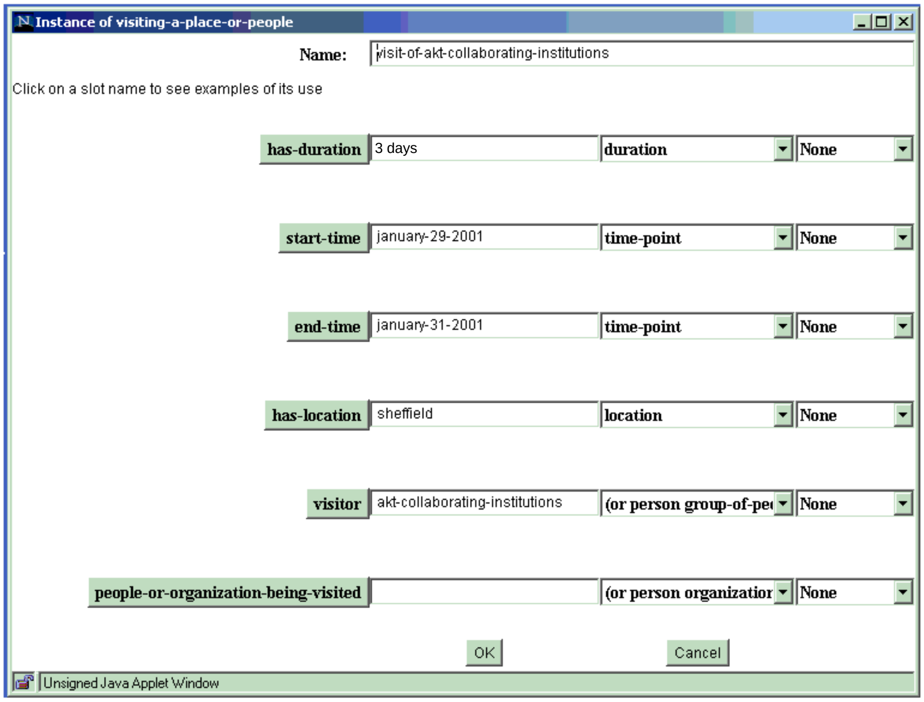

Figure 5: Semi-automatically annotate the e-Story of figure 4: a partial instantiation of the event type: *visiting a place or people*.

categories of interests are pre-defined and maintained internally, the *FindUR* team were "experimenting with a collaborative topic-building environment that allows domain experts to expand of modify topic sets directly"[McGuinness, 1998]. Although this approach has the advantage of speeding up the maintenance task, in our case we see the pre-selected categories as a stable piece of knowledge over time. If however, these categories need to be updated, we could use the `WebOnto`[Domingue, 1998] environment for editing and browsing the underlying ontology. We should also point out a similarity in the use of cue phrases and cue words to increase the number of related e-Stories. In the *FindUR* project, the notion of "evidence phrases" was used. However, their definition as "evidence" phrases highlights a difference in their application: as we described in section 3.1, we use cues both as abstractions of terms and as evidence whereas in the *FindUR* domain they used only as evidence. For example, as the authors describe, the company *Vocaltec* could be an evidence for the topic *Internet telephony* but certainly is not an abstraction of it. In particular, they defined a typology of evidence phrases: *synonyms*, *subclasses*, *products*, *companies*, *associated standards*, *key people*. These were then used to increase the number of related answers to a given query. They were deployed in the background along with rules that govern their interrelations. As in our approach, these were not automatically generated.

A similar approach which deploys *content matching* techniques is described in [Guarino *et al.*, 1999] where the authors present the *OntoSeek* system designed to support content-based access to the Web. As in the *FindUR* project, the target was the Information Retrieval area with the aim of improving recall and precision and the focus was two specific classes of information repositories: yellow pages and product catalogues. Their underlying mechanism uses conceptual graphs to represent queries and resources descriptions. As the authors argue, "with conceptual graphs, the problem of content matching reduces to ontology-driven graph matching, where individual nodes and arcs match if the ontology indicates that a subsumption relationship holds between them"[Guarino *et al.*, 1999]. However these graphs are not constructed automatically. The *OntoSeek* team developed a semi-automatic approach in which the user has to verify the links between different nodes in the graph via a designated user-interface. The similarity of this work with *myPlanet* lies in the usage of an ontology. However, as previously, we deployed our ontology in different phases: in structuring the search space and in increasing the answer set.

On a slightly different focus, the IMPS(Internet-based Multi-agent Problem Solving) system uses software agents to conduct knowledge acquisition on-line using distributed resources[Crow and Shadbolt, 1999]. One of these agents, *OCA*(Ontology Construction Agent), is used to facilitate the task of constructing an ontology at runtime, that is, querying various resources for filling in the gaps in the ontology. Although the goals of this work were different, the underlying idea for the *OCA* is similar to our efforts of populating the ontology by automatically instantiating classes as we described in section 3.2. *OCA* was used "to extract information from networked knowledge resources - like WordNet, the online thesaurus/lexical database and a plain text domain database in the field of geology, the IGBA dataset"[Crow and Shadbolt, 1999]. Our approach is different in that we deploy IE techniques along with domain specific templates to instantiate specific ontology classes whereas the *OCA* deploys heuristic methods for extraction and focuses on creating an hierarchy lattice of classes of concepts.

In the context of managing user profiles we should point to attempts that have been made to infer user profiles from analyzing patterns of access to documents [Krulwich and Burkley, 1997]. However, most of these approaches try to induce user interests by employing empirical methods. In our case, we deliberately impose an ontology-driven structure to the user profile which enabled us to reason about it.

Finally, [Roux *et al.*, 2000] and [Faatz *et al.*, 2000] discuss early ideas on the use of IE techniques coupled with ontologies in order to help them understand complex relationships, statements or terms in semi-structured or unstructured documents.

5 Summary and future work

In this paper we presented a system, *myPlanet*, which acts as the front-end to a news server. It is placed on the top of the existing infrastructure for ontology-driven Web-based news services, **PlanetOnto**. It aims to allow users browse e-Stories according to their preferences(i.e., search criteria). The usage of the underlying ontology allowed us to devise heuristics which make it possible to increase the answer set of related e-Stories. We also provide facilities for saving users' profiles, a feature vital for providing further services tailored to their preferences.

While the ease of accessibility to our e-Stories repository was a primary goal, equally important was the maintenance of this repository. Since we base our services on the enrichment of e-Stories in terms of annotating them with ontology-drawn knowledge structures we had to find ways of automating this process. We used IE techniques and developed domain specific templates to automatically identify the event type of a e-Story and extract specific information needed for instantiating it in the underlying ontology.

There are certain research issues which remain open in this work. In the area of personalised services we need to take the ontology-based reasoning to a further stage: reason about the kind of output that will be dispatched to the user by analysing his/her profile. Since we save the user's preferences we could apply deductive heuristics to find e-Stories that are related to these preferences by means of tracing their interrelations in the underlying ontology. A simple example could be to infer that technologies used in projects might be of interest to users that looking for e-Stories related to other projects with the same research area. Furthermore, we are investigating the possibility of extending the type of output. Currently, a related e-Story is the output of *myPlanet*. In the future though, we might want to provide other kind of output like, for example, suggestions about potential collaborators on a research topic, or organizations with a potential interest in the user's research areas. These could be inferred by applying the same style of deductive heuristics but changing the output to a designated "personal interests" Web-page. As in the existing system, editing facilities are vital to keep the system updated and let the user drive the reasoning process.

One of the advantages of our "lowest-common-denominator" medium(the email message) is that we make no commitments as to what the structure should be. Which means that we can apply exactly the same infrastructure to any kind of document, not necessarily email messages. The technology needs no changes, however we might need to edit or even create new ontologies to characterise the new domain. Towards this direction, we plan to extend the usage of IE techniques coupled with domain specific templates as it has been proved a fast way of instantiating our ontologies. In our ontology population task we had to manually construct the templates for each type of event. We are planning to automate this task by deploying inductive learning algorithms. The existing set of e-Stories could be used, potentially, as the training set to identify characteristics of event types which will eventually lead to automatically construct their templates. These templates can then be tested on the annotated e-Stories to judge their quality and appropriateness. In the same line of work, we intend to expand on IE techniques and include tools that allow detection of anaphora which is an important feature when dealing with large corpusses of text from the same organisation but different departments. In these cases, terms are often used in different formats(i.e., abbreviated names). Co-referrences between those are important to be identified prior to IE tasks in order to avoid duplications or omissions of information.

Finally, the use of cue phrases and cue words for increasing the answer set worked well in our approach. Although the set is relatively small(we have something like 200 cue phrases defined) their identification need to be automated. To do this we have begun to work with a technique borrowed from the data engineering domain [Krulwich, 1995], which applies heuristics to identify 'semantically significant phrases'. The underlying principle is to observe visual effects often used by authors to emphasize important concepts in their documents. For example, boldfaced or italicised words, heavily repeated phrases, compound noun phrases, list of items, etc. We have build a prototype tool which extracts a large set of potential cue phrases after applying a designated set of heuristics. The potential phrases will then be edited to construct the final set.

With this first version of *myPlanet* and the extensions we plan to make we are working towards the vision of the *Knowledge User* era where the user is the focal point in a setting with a plethora of knowledge-intensive systems aim to deliver intelligent services over the Web surrounding him. This metaphor, although in its infancy yet, is in contrast with the traditional view of knowledge-intensive systems being the focal point with users surrounding them acting as subscribers for knowledge services.

Acknowledgements

The research described in this paper is supported by the Interdisciplinary Research Collaboration(IRC) project: Advance Knowledge Technologies(AKT - http://www.aktors.org) funded by the UK government.

References

[Crow and Shadbolt, 1999] L. Crow and N. Shadbolt. Acquiring and Structuring Web Content with Knowledge Level Models. In R. Studer and D. Fensel, editors, *Proceedings of the 11th European Workshop on Knowledge Acquisition, Modelling and Management(EKAW'99), Dagstuhl, Germany*, pages 83–101. Springer Verlag, May 1999.

[Domingue and Motta, 2000] J. Domingue and E. Motta. Planet-Onto: From News Publishing to Integrated Knowledge Management Support. *IEEE Intelligent Systems*, 15(3):26–32, 2000.

[Domingue, 1998] J. Domingue. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. In *Proceedings of the 11th Knowledge Acquisition, Modelling and Management Workshop, KAW'98, Banff, Canada*, April 1998.

[Faatz *et al.*, 2000] A. Faatz, T. Kaamps, and R. Steinmetz. Background knowledge, indexing and matching interdependencies if document management and ontology maintenance. In *position paper in Proceedings of the ECAI2000 workshop on Ontology Learning(OL2000), Berlin, Germany*, August 2000.

[Guarino *et al.*, 1999] N. Guarino, C. Masolo, and G. Vetere. OntoSeek: Content-Based Access to the Web. *IEEE Intelligent Systems*, 14(3):70–80, May 1999.

[Huhns and Stephens, 1999] M. Huhns and L. Stephens. Personal Ontologies. *IEEE Internet Computing*, 3(5):85–87, September 1999.

[Kalfoglou *et al.*, 2000] Y. Kalfoglou, T. Menzies, K-D. Althoff, and E. Motta. Meta-knowledge in systems design: panacea...or undelivered promise? *The Knowledge Engineering Review*, 15(4), 2000.

[Kalfoglou, 2000a] Y. Kalfoglou. *Deploying Ontologies in Software Design*. PhD thesis, Department of Artificial Intelligence, University of Edinburgh, June 2000.

[Kalfoglou, 2000b] Y. Kalfoglou. On the convergence of core technologies for knowledge management and organisational memories: ontologies and experience factories. In *Proceedings of the ECAI 2000 Workshop on Knowledge Management and Organisational Memories(W11-KMOM'00), Berlin, Germany*, August 2000.

[Krulwich and Burkley, 1997] B. Krulwich and C. Burkley. The InfoFinder Agent: Learning User Interests through Heuristic Phrase Extraction. *IEEE Intelligent Systems*, 12(5):22–27, 1997.

[Krulwich, 1995] B. Krulwich. Learning document category descriptions through the extraction of semantically significant phrases. In *Proceedings of the IJCAI'95 Workshop on Data Engineering for Inductive Learning*, 1995.

[McGuinness, 1998] L.D. McGuinness. Ontological Issues for Knowledge-Enhanced Search. In N. Guarino, editor, *Proceedings of the 1st International Conference on Formal Ontology in Information Systems(FOIS'98), Trento, Italy*, pages 302–316. IOS Press, June 1998.

[Motta, 1999] E. Motta. *Reusable Components for Knowledge Models: Case Studies in Parametric Design Problem Solving*, volume 53 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 1999. ISBN: 1-58603-003-5.

[O'Leary, 1998] D. O'Leary. Knowledge Management Systems: Converting and Connecting. *IEEE Intelligent Systems*, 13(3):30–33, June 1998.

[Proux and Chenevoy, 1997] D. Proux and Y. Chenevoy. Natural language processing for book storage: Automatic extraction of information from bibliographic notices. In *Proceedings of the Natural Language Processing Pacific Rim Symposium(NLPRS'97)*, pages 229–234, 1997.

[Riloff, 1996] E. Riloff. An empirical study of automated dictionary construction for information extraction in three domains. *AI Journal*, (85):101–134, 1996.

[Roux *et al.*, 2000] C. Roux, D. Proux, F. Rechermann, and L. Julliard. An ontology enrichment method for a pragmatic information extraction system gathering data on genetic interactions. In *position paper in Proceedings of the ECAI2000 Workshop on Ontology Learning(OL2000), Berlin, Germany*. August 2000.

[Vargas-Vera *et al.*, 2001] M. Vargas-Vera, J. Domingue, Y. Kalfoglou, E. Motta, and S. Buckingham-Shum. Template-driven information extraction for populating ontologies. In *Proceedings of the IJCAI'01 Workshop on Ontology Learning, Seattle, WA, USA*, August 2001.