Personal Ontology Creation and Visualization for a Personal Interaction Management System

Akrivi Katifori, Costas Vassilakis, Ilias Daradimos, George Lepouras, Yannis Ioannidis
Dept. of Informatics, University of Athens,
Panepistimioupolis, Ilissia, 157 84,
{vivi,costas,drid,glepoura,yannis}@di.uoa.gr

Alan Dix
Computing Dept., Lancaster University, LA1 4YR, UK
alan@hcibook.com

Antonella Poggi, Tiziana Catarci
Dipartimento di Informatica e Sistemistica "A. Ruberti",
Università di Roma "La Sapienza",
Via Salaria 113, 00198, Rome, Italy
{poggi,catarci}@dis.uniroma1.it

ABSTRACT

Ontologies offer a flexible and expressive layer of abstraction, very useful for capturing the semantics of information repositories and facilitating their retrieval either by the user or by the system to support user tasks. This work presents an ontology-based user profiler, in the context of a Personal Interaction Management System (PIMS). The profiler, based on an ontology of the users' domain, enables them to create their personal ontology by initially choosing one of the available template ontologies as a starting point, which they subsequently populate and customize. The profiler employs a web interface which allows users to populate their personal ontology through forms, hiding ontology complexities and peculiarities. Forms are dynamically generated through *ontology views*, which are specified by ontology designers.

Author Keywords
User profile, ontology, web-based profiler.

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

As a direct result of the rapid technological progress of the last few decades, personal computers have become repositories for company information and scientific data, documents, electronic mail as well as personal collections of media, like photographs, video or music. Computer users, however, in order to take advantage of this memory aide offered to them, have to invest more and more time in managing and organizing their collections and repositories because, if they don't, retrieving information from them when necessary will be nearly impossible.

Furthermore, in current computer systems the user interaction paradigm is based on (a) functionally-defined applications (word processing, address management, internet browsing) and (b) on the storage, organisation and retrieval of information in files or databases, the content types and structure of which are determined by the units of operation of the applications. However, real activity, whether for work or leisure, crosses application boundaries, may involve portions of files, and interlinks fragments of both. Users should not have to focus on managing their information but rather on performing the tasks this information is to be used for.

Recent research in the domain of Personal Information Management (PIM) and Task-centered Information Management (TIM) has recognized the need for a paradigm shift towards more task- and activity-oriented systems [16]. Ontologies, as semantic networks with a structure very similar to the one used by the human brain for storing long-term knowledge, may be very useful as the basis of such a system. They offer a flexible and expressive layer of abstraction, very useful for capturing the semantics of information repositories and facilitating their retrieval either by the user or by the system to support user tasks. To this end, if combined with appropriate "intelligent" mechanisms, they may become useful tools to record semantics related to documents and tasks and function as an extension to the user's own memory, available both for the user and the system.

However, the end user of such a system should be able to interact with the ontology easily and effectively in order for an ontology based, task oriented system to be successful. This calls for ontology visualization and presentation methods that allow the user to manipulate the ontology through a natural interface, which will capture the user's personal information and document semantics.

This work presents our effort towards a non-expert user – oriented ontology presentation method, to be integrated in a Personal Information Management System (PIMS) prototype, as well as in any application that wishes to use an ontology as way to model the user's personal

CHI 2008, April 5–10, 2008, Florence, Italy.
Copyright 2008 ACM 978-1-60558-011-1/08/04…$5.00

information. The following section briefly outlines the elements of a Personal Interaction Management System (PIMS) and the use of the proposed ontology profiler in the context of the PIMS. Then, the Personal Ontology is described and the next section outlines the ontology profiler. The last section concludes the paper and outlines future work.

THE ONTOLOGY PROFILER WITHIN THE CONTEXT OF A PIMS

The motivation of our work on personal ontologies has been the vision of more activity-centric computing and the general aim of moving from systems focusing on the management of personal information (i.e. PIM) to systems focusing on the management of personal interaction. We define Personal Interaction Management System (PIMS) to be a system that supports the user in executing tasks in an interactive and efficient way, providing at the same time effective and transparent mechanisms for maintaining the user's personal document collection.

In order for a PIMS to be effective, it should provide mechanisms for user profiling, semantic storage of documents and context inference. Figure 1 shows a sketch view of the main components a PIMS must include to support this functionality. The information sources side (documents, emails etc.) is linked to the computation side (actions) through two main components:

1. A *recogniser* finding suitable fragments of the raw information that are semantically meaningful and that can be used to initiate or feed into actions

2. A *personal ontology* that contains knowledge specific to the user (people, projects, etc.).

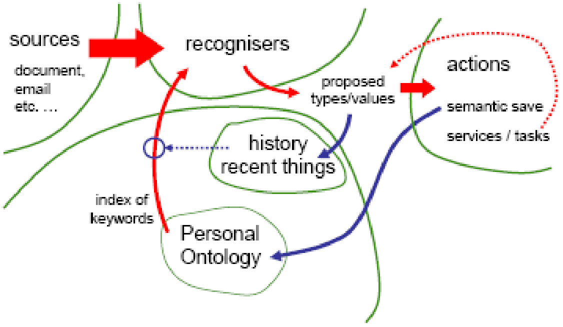


Figure 1. Basic outline of a PIMS

These two feed into one another. The various terms, names, emails, etc, in a personal ontology can yield keywords to be matched against text or semi-structured sources. So an increasingly rich personal ontology will lead to better identification of suitable loci for action. Furthermore, as users perform actions the way in which they use information, the results of their activities can be used to enrich the ontology. For example, if a piece of text is used to search in a gazetteer it suggests that (i) it is a place name - that is we know more about its type and (ii) it is a place name that is important to the user - so it will be suggested to be added into the personal ontology.

Figure 1 also shows a history sub-component related to the personal ontology. We need to record what is done in order to both (a) establish a sense of context and (b) be able to allow the system to gain some understanding of the user's ongoing activities.

This system requires inference mechanisms which sit outside this picture, exploiting both the personal ontology and a representation of recent history to infer information, and then feeding this in to modify the recognition and action selection. For information inference, we use MASTRO[37], an ontology-based data integration system, based on the language DL-Lite$_A$[38], allowing for answering complex queries over the personal ontology, whose extensional level is stored in a relational database. For task inference we aim to use a bottom-up approach the description of which is provided in [16] and prototyped over actions on web forms. For context inference we are using spreading activation over the Personal Ontology. Both task and context inference mechanisms are currently work under progress.

Furthermore, a PIMS requires an appropriate interaction mechanism for allowing the user to have access to the information stored in the Personal Ontology, to use and update it manually if needed. The Personal Ontology itself is one of the keys to the success of our proposed PIMS, as it constitutes its "knowledge" and "memory" of the user's personal information and history.

The PIMS architecture and individual components are discussed in more detail in [16] and [18], whereas this work focuses on the personal ontology and its interaction with the user through the web-based ontology profiler tool.

ONTOLOGIES AND PERSONAL INFORMATION MANAGEMENT

According to [19], an ontology is an explicit specification of a conceptualization. The term "conceptualization" is defined as an abstract, simplified view of the world that needs to be represented for some purpose. It contains the concepts (classes) and their instantiations (instances) that are presumed to exist in some area of interest and their properties and relations that link them (slots). The term "ontology" is borrowed from philosophy, where an ontology is a systematic account of Existence. This section presents the creation of a personal ontology to be the basis of the intelligent context inference mechanism of our PIMS.

Ontologies in PIMS

Using an ontology to model semantics related to the user personal domain has already been proposed for various applications like web search [2], [10]. Most of these approaches use ontologies only as concept hierarchies, like hierarchies of user interests, without particular semantic complexity, as opposed to our approach which incorporates the full range of ontology characteristics.

The value of ontologies for personal information management has also been recognized and there is on-going research on incorporating them in PIM systems like OntoPIM [1], GNOWSIS [21] and the semantic desktop search environment proposed in [22]. However, up to this point, there is no detailed work available on the exact personal ontology to be used for such an application.

In the context of our proposed PIMS the personal ontology has a very important part to play. On one hand, it may constitute a useful repository of information related to many aspects of the user's personal and professional life. There the user will be able to store and access information on contacts (friends, colleagues, etc), activities (like a research project or a hobby), events (like project meetings, conferences, etc), documents (collected books and research papers, etc) and tasks. With the appropriate interface, the ontology may become an easily customizable repository of information that may serve as a memory aide for the user. On the other hand, coupled with intelligent mechanisms, the ontology may become invaluable for context inference in the process of supporting user tasks, through task inference.

It should be noted here that, taking into account the variety of user groups and needs, the Personal Ontology will be a basic core of classes that will be easily extensible to accommodate several user stereotypes. In the context of the PIMS prototype, the use of several *template* Personal Ontologies is proposed, all being extensions of the basic core of classes described in this work, enriched with classes related to specific domains and user stereotypes, like "Researcher", "Professor", "Student", "Architect", etc.

The Personal Ontology constitutes an extended and enriched version of a user profile maintained by most applications as it attempts to group under one structure the user personal information, contacts, interests, important events, etc. The next section provides more details on the ontology and its creation.

Personal Ontology Creation and Characteristics

Creating a personal ontology, either automatically, manually or semi-automatically is not an easy task. In order for such an ontology to be truly personal, it should be able to reflect the user individuality, but, it should do so in the context of a specific general model that will enable exchange of information between users and will be usable by computers. This is the main reason why the personal ontology model we propose encompasses a basic core of general concepts that may be enriched to accommodate several user stereotypes or individual profiles. The addition of new classes may be accomplished both at the ontology designer and the end user level.

In order to create a simple yet comprehensive set of upper level concepts for the personal ontology, profile information models maintained by various applications, like instant messengers [23] and community websites [24] [26], and proposed by researchers, like [2], [9], [10], [11], [27], were examined and general ontologies like the ones presented in [28] were taken into account along with the MIME directory profile vCard [15].

More details on the creation of the personal ontology may be found in [12]. The version of the personal ontology used in this work [17] is an extension of the one in [12], as it has been enriched with more user-related classes and underwent several changes in its structure. The ontology, along with example instances may be found in [17]. Figure 2 presents an overview of the upper levels of the class hierarchy.

The personal ontology attempts to encompass a wide range of user characteristics, including personal information as well as relations to other people, preferences and interests. The ontology may be extended through inheritance and the addition of more classes, as well as class instantiation according to the needs of user stereotypes or individuals.

The personal ontology classes are divided in two main groups, which comprise the two upper levels of the ontology, "Thing" and "Value class" (Figure 2).

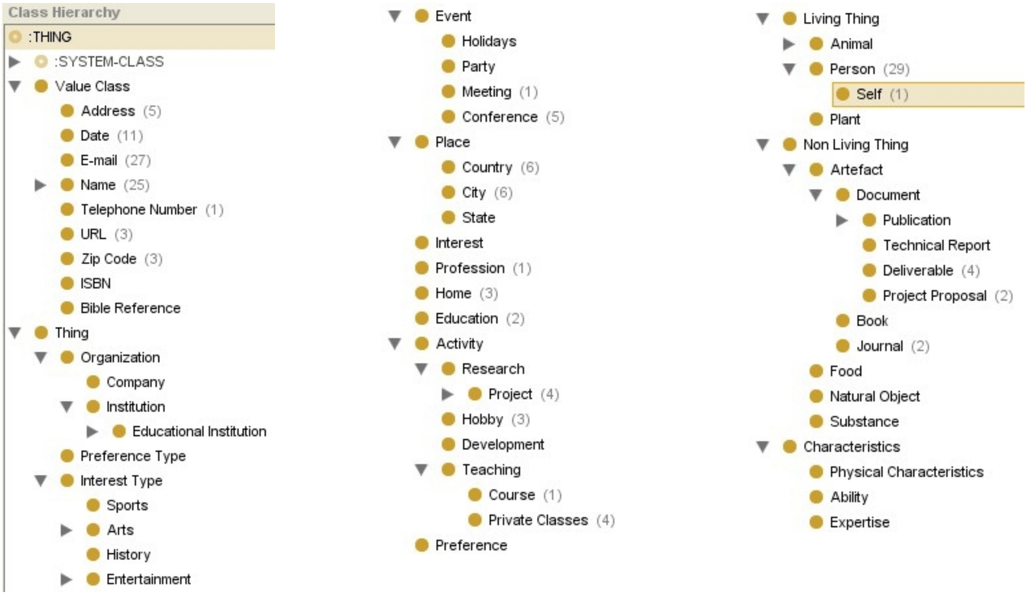
Class "Thing" (Figure 2) denotes the set of all personal domain objects, which may consist of both abstract and tangible things. Example of classes denoting abstract things are "Interest Type" and "Preference Type". These model interest and preference hierarchies as the ones suggested in [2] and [10]. Example of classes denoting tangible things are "Organization" and "Person". Note in particular that "Self" (highlighted in Fig. 2), is a direct subclass of "Person", modeling thus the user.

On the other hand, "Value Class" denotes the set of all possible domain values. Whereas simple data types are implicit sub-classes of "Value Class", the ontology designer has the ability to define more complex data types to be included in the ontology as separate classes. These among others include dates, URLs, telephone numbers, zip codes and names and may be used as types of slot modeling attributes for the concepts of the "Thing" sub-hierarchy. Instances of these classes may serve as information items driving both task and context inference in PIMS.

The purpose for the addition of "Value Class" in the ontology was the need for basic information items, like person names, urls, telephone numbers, etc, to have a separate existence in the ontology. They have been modeled as a distinct class so as to be identified by the recognizers as such, to apply to them special validation/formatting rules and/or provide tailored editing methods (e.g. forms). In addition they may have internal structure such as 'first name', 'surname', 'title' for a name or 'day', 'month', 'year' for a date. If we modelled these Value Classes as types of things (as is the case with RDF), this internal structure of complex data values would be lost inside the string representation and its semantics unavailable in the ontology.

However, care has to be taken with Value Classes in the case when two "Thing-typed" instances point to the same

"Value-class" instance. For example, in the ontology there may exist two distinct instances of the Value Class "Date" with the same value, 11 June 1977. The one will represent the birthday of a "Person" whereas the other, the publication date of a "Journal". It is important that a modification to the publication date is performed by allocating a fresh date instance and not updating the shared one. The fact that the Value Classes sit under a separate branch of the class structure makes it easy to ensure these semantics in the user interface and underlying application logic.



Class Hierarchy

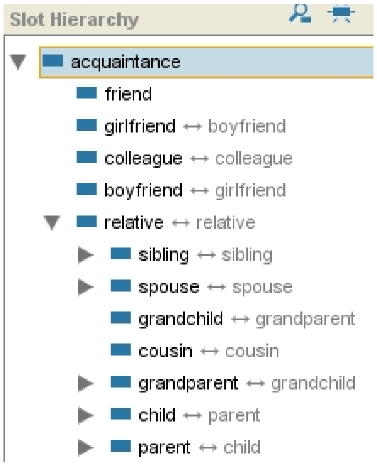
- :THING
  - :SYSTEM-CLASS
  - Value Class
    - Address (5)
    - Date (11)
    - E-mail (27)
    - Name (25)
    - Telephone Number (1)
    - URL (3)
    - Zip Code (3)
    - ISBN
    - Bible Reference
  - Thing
    - Organization
      - Company
      - Institution
        - Educational Institution
    - Preference Type
    - Interest Type
      - Sports
      - Arts
      - History
      - Entertainment

- Event
  - Holidays
  - Party
  - Meeting (1)
  - Conference (5)
- Place
  - Country (6)
  - City (6)
  - State
- Interest
- Profession (1)
- Home (3)
- Education (2)
- Activity
  - Research
    - Project (4)
  - Hobby (3)
  - Development
  - Teaching
    - Course (1)
    - Private Classes (4)
- Preference

- Living Thing
  - Animal
  - Person (29)
    - Self (1)
  - Plant
- Non Living Thing
  - Artefact
    - Document
      - Publication
      - Technical Report
      - Deliverable (4)
      - Project Proposal (2)
    - Book
    - Journal (2)
  - Food
  - Natural Object
  - Substance
- Characteristics
  - Physical Characteristics
  - Ability
  - Expertise

Figure 2 Overview of the Personal Ontology with the upper levels expanded

In a personal ontology, relations to other people -either personal or work-related- play a very important part. We used slot sub-classing to create a set of basic person relations slots, as shown in Figure 3, with the “acquaintance” slot sub-hierarchy. This can be extended or adapted according to the needs of each application and user stereotype.



Slot Hierarchy

- acquaintance
  - friend
  - girlfriend ↔ boyfriend
  - colleague ↔ colleague
  - boyfriend ↔ girlfriend
  - relative ↔ relative
    - sibling ↔ sibling
    - spouse ↔ spouse
    - grandchild ↔ grandparent
    - cousin ↔ cousin
    - grandparent ↔ grandchild
    - child ↔ parent
    - parent ↔ child

Figure 3. The first levels of the slot hierarchy modeling person relations. The inverse slots are represented by Protégé in grey.

The ontology has been modeled using Protégé [13], due to the fact that, apart from being a widely used ontology open source ontology editor, it offers a variety of ontology visualization methods, useful for experimentation to determine the most appropriate one for the end user.

A WEB-BASED ONTOLOGY PROFILER

There exist a number of ontology visualizations that are being used in the context of ontology management tools or as information retrieval aids in applications that use ontologies. Some interesting ontology management tool surveys may be found in the Protégé web pages [13], whereas in [29] a detailed presentation of existing ontology visualization methods may be found. The preliminary results of a survey using questionnaires related to ontology editing tools and ontology visualization are presented in [30].

There are also evaluations of hierarchical visualizations like the ones in [31] and [32], however, up to this point, there is a lack of extensive comparative evaluations concerning the effectiveness of ontology visualization methods in different contexts and with different users.

An Evaluation of Ontology Visualization Methods

In order to identify an appropriate visualization method for users not familiar with the concept of ontologies, we have conducted an evaluation of four ontology visualization techniques [33]. The techniques are Class Browser [13], Jambalaya [36], TGViz [34] and OntoViz [35], representative of the four main ontology visualization approaches defined in [29], indented list, zoomable, context+focus and node-link/tree respectively.

A group of 37 users with various degrees of computer expertise participated in the experiment. The purpose of the experiment was to investigate ontology visualization issues relevant not to expert ontology designers but to ontology users not necessarily familiar with the concept of an ontology. Information retrieval (IR) tasks of different types and varying complexity were posed to users in this evaluation, in order to investigate the strong and weak points of the methods for each IR task class. Special attention was given to queries involving temporal characteristics (e.g. evolution of an entity as well as the users' browsing method in relation with the visualization employed.

The preliminary results of the evaluation may be found in [33]. The complete results are still being processed; however it has already been made evident that ontology visualization alone is not enough for supporting complex queries. A visualization with interactive features should be combined with a powerful search mechanism in order to be really effective.

Furthermore, some users had problems when familiarizing themselves with the ontology model - they found it in some cases overwhelming. The existence of many empty slots was a bit annoying to them, as they felt that they somehow occluded the non-empty ones. In all the visualizations it seemed that the full complexity of an ontology and especially features like multiple inheritance may be difficult for the end user to comprehend.

Taking into account that user's personal ontology contains information relevant to the user and his/her domain of interest, s/he should be able to manipulate and update it easily. As making the ontology simpler would mean diminishing its power for providing complex semantics to our PIMS, we opted for investigating appropriate presentation methods for the interaction of the user with the personal ontology. The following section describes our first steps to that direction in more detail.

The Personal Ontology Profiler

The Personal Ontology Profiler in this section is being presented as a separate application that communicates with the selected ontology storage repository through web services; therefore, this module may be used not only in the context of our PIMS, but also for other applications opting for using a personal ontology as a user profile model.

The profiler is being implemented as a standard web application using HTML and PHP and its communication with the ontology storage module may be accomplished through the exchange of ontology data with the ontology storage module in XML format [39].

For the moment, in order to test the profiler, we have been using a database as ontology storage, which is based on the Protégé database format. Appropriate web service methods were implemented for constructing the XML documents containing the information requested by the profiler [39].

Applications wishing to retrieve the ontology for performing personalization or adaptivity tasks may request it from the ontology storage repository, in our case the Protégé server [14]. The intermediation of the Protégé server, besides offering enhanced security (since applications do not directly access the database), enables the use of alternative storage formats.

It is worth noting here that since the ontology schema is maintained together with the ontology instances and their relevant slot values, the profiler application is made immediately aware of any changes to the ontology schema (e.g. addition of new classes or relationship types). This dynamic nature of the profiler is very important, as it allows changes made to the ontology schema to immediately update the profiler forms. These changes are made by ontology designers, typically using the Protégé Frames [13] editor.

The overall architecture of the proposed system is illustrated in Figure 4. In this figure, the Personal Ontology Database is a repository where the personal ontologies of each user are stored. In the repository a number of personal ontology templates are available, which correspond to

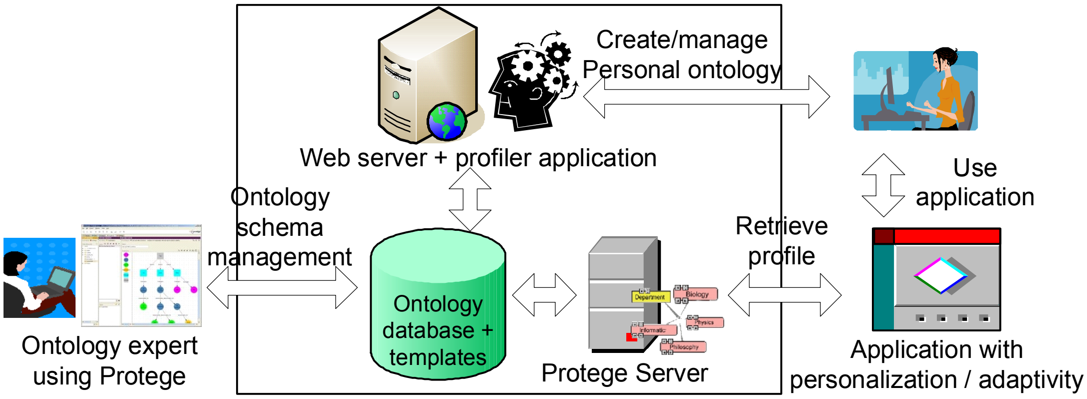


Figure 4. Overall system architecture

various user stereotypes. Ontology designers, using an ontology editor like Protégé, may edit the ontology templates or add new ones.

In general we might expect a user to select more than one starting template. For example Tiziana may select a researcher profile bringing in concepts such as 'Journal' and 'published_in', but also may be interested in horses bringing in concepts such as 'Horse' and 'is ridden by'. So long as these templates are sub-graphs of a single schema then using several stereotype templates, or adding additional ones later, is simply a union of the relevant classes and relations. However, we may want to add facilities at some point to allow users to select roles so that only relevant classes and instances are shown at particular times.

While the users are expected to start with predefined classes, it is also expected that they will develop their ontologies over time, not just by adding instances, but also by adding new classes (usually subclasses of existing ones) and relations. Often these classes may be idiosyncratic representing some concept that is only of meaning to the particular user – for example a subclass of ''Friend' might be added called 'Darts' where the user puts the group of friends she plays darts with.

As well as giving users a start point for their personal ontology, the use of templates means that core concepts are standardized and therefore can more easily linked to external resources and tools. For example, an instance of the idiosyncratic 'Darts' class would be recognized as also being an instance of 'Friend' and 'Person', so that, for example, they could be linked to external directory services.

Applications, like a PIMS, which will make use of a personal ontology, will retrieve it through the server that communicates with the ontology storage. The end users of these applications will be able to access and edit their personal ontology by interacting with it through the ontology profiler module.

Creating, however, a tool to allow the end user to edit his/her personal ontology is not a straightforward task. As already stated, the ontological structure is not very clear, especially to a non-expert user with no familiarity with ontologies. To this end, the profiler has to provide the end-user with an appropriate presentation layer, screening the complexities and peculiarities of the ontological structure. The ontology profiler is responsible for presenting to the user a simplified view of the personal ontology, organized in forms, as defined by the ontology designer in Protégé.

As it was explained in the previous section on the characteristics of the personal ontology, all information that the user has to fill in is in fact the slots of the instance of the "Self" class. As a result, the key issue here is to provide to the user an appropriate, simple and meaningful, visualization of these slots, preferably modeled after a form-based interface, to which users are generally accustomed. These forms, however, could not be statically designed, since any changes in the ontology schema would require corresponding changes to the form layouts and back-end code. To this end, Ontology Views have been introduced. The ontology designer will define for each class in the ontology one or more "views", meaning sets of slots, along with their order and possibly alternative labels that will be presented to the user when adding or editing an instance of the class. These views will correspond to different dynamically generated forms in the profiler,

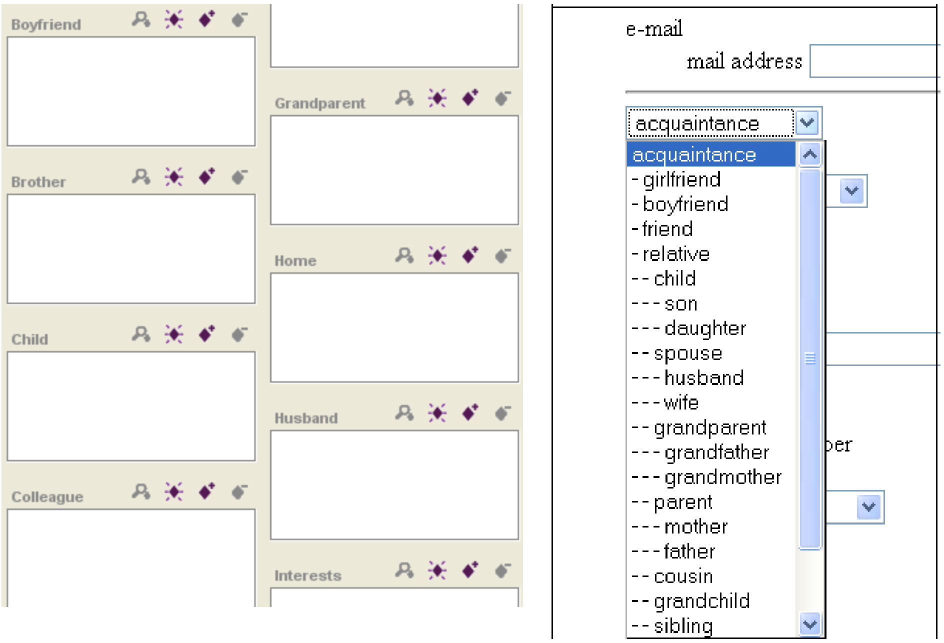


Figure 5. On the left: some of the "Self" class person relation slots as shown in the Protégé tool. On the right: these slots are presented in a more organized, hierarchical way in the Profiler.

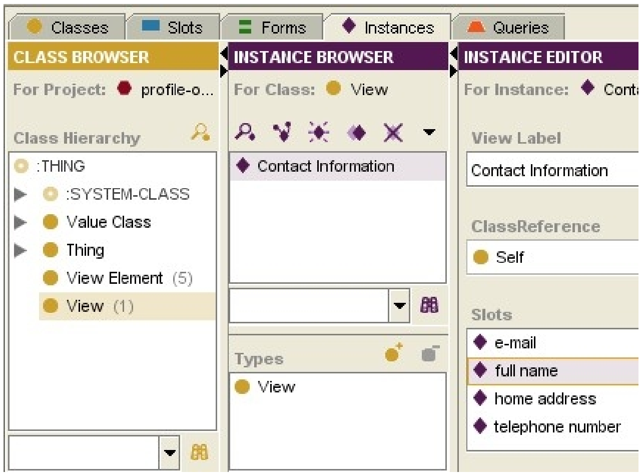
allowing thus the formulation of groupings of information that may be more familiar to the user and easier to handle.

For the initial implementation of the profiler, the views have been included in the ontology itself as sub-classes of the "View" class. This option has the advantage that the ontology designer, when fine-tuning the ontology and defining the profiler views, does not have to use a separate tool for ontology editing and the definition of views. To accommodate view definitions, two ontology classes have been added to the ontology, "View" and "View Element".

The class "View" contains a reference to the class this view refers to, a label of the View as well as a set of instances of the "View Element" class. The "View Element" class contains a reference to the slot, an alternative label and an ordering field, which defines the position of the slot in the view form.

Figure 5 presents the interface for filling in slots values as provided by the Protégé ontology management tool and the proposed profiler. As evident from the figure, Protégé (a tool primarily addressed to ontology designers and experts) presents all the "Self" slots on the same pane and with alphabetical order. Slot inheritance used in the "person relation" slot hierarchy is not explicitly presented. On the other hand, the profiler apart from grouping slots in separate forms also provides a more organized view of slot inheritance. The example presents the "person relations" available in a drop down list in the profiler. When the user wishes to add a new acquaintance, s/he may choose from the list its type.

As an example of a View, Figure 6 presents the "View" instance "Contact Information", used to group the user's full name, address, telephone number and e-mail.



For Project: ● profile-o...

For Class: ● View

For Instance: ◆ Cont

:THING
:SYSTEM-CLASS
Value Class
Thing
View Element (5)
View (1)

◆ Contact Information

● View

View Label: Contact Information

ClassReference: ● Self

Slots:
◆ e-mail
◆ full name
◆ home address
◆ telephone number

Figure 6. The "View Class" instance for the user Contact Information

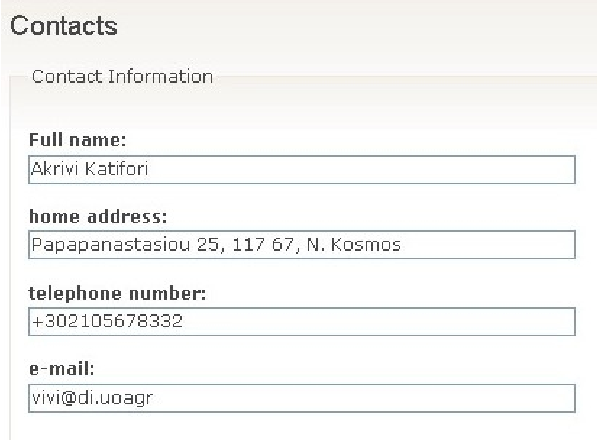
Using this View, the ontology profiler automatically creates the appropriate web form where the user may view and edit this information (Figure 7).

Besides plain value typing and editing, ontology manipulation through views allows instance creation and linking between entities, providing thus full ontology population capabilities to users.

CONCLUSIONS AND FUTURE WORK

This work presents a web profiling application based on an ontology of the user's domain in the context of a PIMS prototype. The main advantage of the profiler is its dynamic characteristics, as the presented information may change according to changes to the underlying ontology schema. The profiler is coupled with a personal ontology that models several user characteristics and may be extended and adapted to accommodate a wide range of applications, as the profiler has been designed to be an independent module and communicate with the other PIMS modules only through web service methods.

The basic core of the Personal Ontology presented in this work has been completed, but it is still under evaluation. Different users are asked to populate it with their own information and make comments on its design and completeness.



Contacts

Contact Information

Full name: Akrivi Katifori

home address: Papapanastasiou 25, 117 67, N. Kosmos

telephone number: +302105678332

e-mail: vivi@di.uoagr

Figure 7. The Contact view as it is created by the profiler

Furthermore, user stereotypes need to be incorporated in the ontology in the form of templates, allowing the user to choose from the beginning sets of classes that would be more relevant to his/her profession or domains of interest and then to further fine tune the personal ontology.

There are also other issues under investigation such as the modeling of more dynamic user characteristics (e.g. his/her current position), provisions for applications targeted to mobile devices, multilinguality, automatic ontology population and privacy issues when sharing information with ontological tagging.

The profiler is at its final implementation stage and an evaluation will be designed in order to test its effectiveness and collect user feedback in order to further improve it. In order to test its effectiveness with a working application, we are currently preparing its integration with the Snip!t tool.

Snip!t [3] [4] is a web bookmarking tool, however unlike most web bookmark utilities, the user can also select a portion (snip) of the page content and then, using a bookmarklet, this selection is sent to the Snip!t web application. The snip of the page is stored along with the page URL, title etc, and the user can classify the snip into categories or share it with others using RSS feeds. In addition, when the selected text contains a recognised type of data such as a date, postal code, person's name, etc., then actions are suggested.

Lastly, there is on-going work on the rest of the PIMS modules, namely task and context inference and the extension of Snip!t to be the core of our PIMS prototype.

REFERENCES

1. Katifori, V., Poggi,. A., Scannapieco, M., Catarci, T., Ioannidis, Y. OntoPIM: how to rely on a personal ontology for Personal Information Management, In Proc. of the 1st Workshop on The Semantic Desktop, (2005).
2. Trajkova, J., Gauch, S. Improving Ontology-based User Profiles, Proc. of RIAO 2004, University of Avignon (Vaucluse), France, April 26-28, (2004), 380-389
3. Snip!t, http://www.snipit.org
4. Dix A., Beale R. and Wood A. Architectures to make Simple Visualisations using Simple Systems. In Proceedings of Advanced Visual Interfaces (AVI2000), (2000) ACM Press. 51-60.
5. Dix A. and Marshall J. At the right time: when to sort web history and bookmarks. In Volume 1 of Proceedings of HCI International 2003. J. Jacko and C. Stephanidis (ed.). 2003. Lawrence Erlbaum Associates., (2003), 758-762
6. Kules, B. User Modeling for Adaptive and Adaptable Software Systems, 2000, Available at http://www.otal.umd.edu/UUGuide/wmk/
7. Rich, E. Users are individuals: individualizing user models, International Journal of Man-machine Studies 18(3), (1983) 199-214.
8. Cornelis, B.: Personalizing search in digital libraries, Master's thesis CS 03-01, University of Maastricht, (2001).
9. Kobsa, A. User Modelling: Recent work, prospects and hazards, in Adaptive User Interfaces: Principles and Practices, edited by Schneider-Hufschmidt, T. Khme and U. Malinowski, North Holland, (1993).
10. Gauch, S., Chaffee, J., Pretschner, A. Ontology-Based User Profiles for Search and Browsing, User Modeling and User-Adapted Interaction: The Journal of Personalization Research, Special Issue on User Modeling for Web and Hypermedia Information Retrieval, (2003).
11. Razmerita,, L., Angehrn, A., Maedche, A. Ontology based user modeling for Knowledge Management Systems, Proceedings of the User Modeling Conference, Pittsburgh, USA, Springer Verlag, pp. 213-217, (2003).
12. Golemati M., Katifori A., Vassilakis C., Lepouras G., Halatsis C. Creating an Ontology for the User Profile: Method and Applications, in Proc. Of the First RCIS Conference, April 23-26, 2007, Ouarzazate, Morocco.
13. Protégé, http://protege.stanford.edu/
14. Protégé multi-user tutorial, http://protege.cim3.net/cgi-bin/wiki.pl?MultiUserTutorial
15. Representing vCard Objects in RDF/XML, 2001, http://www.w3.org/TR/vcard-rdf
16. Catarci, T., Dix, A. Katifori, A., Lepouras G. and Poggi A. Task Centered Information Management, (2007), In Proceedings of the DELOS Conference, February 2007, Springer-Verlag LNCS. To appear.
17. Katifori, A., Vassilakis, C., Dix, A., Daradimos, I., Lepouras, G. Spreading activation user profile ontology, http://oceanis.mm.di.uoa.gr/pened/?category=pub#ontos
18. Lepouras, G., Dix, A., Katifori, A., Catarci, T., Habegger, B., Poggi, A., Ioannidis, Y. OntoPIM: From Personal Information Management to Task Information Management, Personal Information Management, SIGIR 2006 workshop, August 10-11, (2006), Seattle, Washington
19. Gruber, T. R. A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, Special issue: Current issues in knowledge modelling, Vol 5, Issue 2 (1993) 199-220
20. Catarci, T., Habegger, B., Poggi, A., Dix, A., Ioannidis, Y., Katifori, A., and Lepouras, G. Intelligent user task oriented systems. In In Proc. of the Second SIGIR Workshop on Personal Information Management (PIM 2006), 2006
21. Sauermann, L. The Gnowsis Semantic Desktop for Information Integration, Proceedings of the 3rd Conference Professional Knowledge Management (2005)
22. Chirita, P.-A., Gavriloai, R., Ghita, S., Nejdl, W., Paiu, R. Activity Based Metadata for Semantic Desktop Search, Proceedings of the 2nd European Semantic Web Conference (2005)
23. ICQ, http://www.icq.com
24. Facebook, http://www.facebook.com/
25. Representing vCard Objects in RDF/XML, 2001, http://www.w3.org/TR/vcard-rdf
26. MySpace, http://www.myspace.com/
27. Tazari, M. R., Grimm, M., Finke, M. Modeling User Context, Proceedings of the 10th International Conference

on Human-Computer Interaction (HCII2003), Crete, Greece, June 2003

28.Miller, G. A. WORDNET: An On-Line Lexical Database. International Journal of Lexicography 3–4, 1990, pp. 235–312.

29.Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., Giannopoulou , E., Ontology Visualization Methods - A Survey, To appear in ACM Computing Surveys

30.Ernst, N. A., and Storey, M.-A., A Preliminary Analysis of Visualization Requirements in Knowledge Engineering Tools, University of Victoria, Victoria, CHISEL Technical Report August 19, 2003

31.Kobsa, A. 2004. User Experiments with Tree Visualization Systems. In IEEE Symposium on Information Visualization (INFOVIS'04), 9-16

32.Wiss, U., Carr, D., and Jonshon, H. 1998. Evaluating Three – Dimensional Visualization Designs: a Case Study of Three Designs. In Proceedings of the Second International Conference on Information Visualisation (IV'98), p. 137.

33.Katifori, A., Torou, E., Halatsis, C., Lepouras, G., and Vassilakis, C., A Comparative Study of Four Ontology Visualization Techniques in Protégé: Experiment Setup and Preliminary Results, proceedings of the IV 06 Conference.

34.Alani, H., TGVizTab: An Ontology Visualisation Extension for Protégé, Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering, Sanibel Island, Florida, USA, (2003)

35.Sintek, M., Ontoviz tab: Visualizing Protégé ontologies, 2003, http://protege.stanford.edu/plugins/ontoviz/ontoviz.html

36.Storey, M.-A., Mussen, M., Silva, J., Best, C., Ernst, N., Fergerson, R., Noy, N. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé, Workshop on Interactive Tools for Knowledge Capture, K-CAP-2001, Victoria, BC, Canada, (2001), http://www.thechiselgroup.org/jambalaya

37.Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Poggi, A. and Rosati, R. Ontology-based database access. In *Proc. of the 15th Italian Conf. on Database Systems (SEBD 2007)*, 2007.

38.Poggi, A., Lembo, D., Calvanese, D., De Giacomo, G., Lenzerini, M., and Rosati R. Linking data to ontologies. Journal of Data Semantics (JODS). To Appear.

39.Katifori, A., Daradimos, I., Dix, A. Web Methods for Ontology Retrieval, Technical Report, http://oceanis.mm.di.uoa.gr/pened/?c=pub#tr