The On-TIME User Interface (Demonstration)

T. Catarci*, R. Giuliano, M. Piva, A. Poggi*, F. Terella, E. Tracanna

SAPIENZA Università di Roma

*{catarci,poggi}@dis.uniroma1.it

ABSTRACT

This demonstration concerns the visual user interface of the On-TIME system, a task-centered information management system, whose aim is to actively participate to and support the user tasks. The design of a user friendly interface is one of the key challenges that needs to be addressed for the success of On-TIME. Being On-TIME based on the use of a so-called Personal Ontology to provide a semantic account to user's personal data, the interface has to allow the user to easily browse the ontology. On the other hand, it has to address the management of tasks. This requires to both suggest tasks that the user might be willing to perform, and to support her while executing tasks. We present a typical user scenario in order to illustrate a possible interaction with the On-TIME interface, and discuss some preliminary user evaluation.

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces;

H.3.3 [Information Storage and Retrieval]: Miscellaneous;

H.2.8 [Database Management]: Database Applications

General Terms

Human Factors, Experimentation

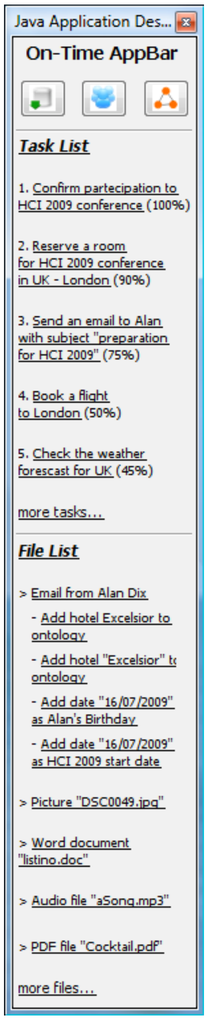
Keywords

Personal Information Management, Task Management, Ontologies

1. INTRODUCTION

Personal Interaction Management System (PIMS) [6] is a new paradigm of system, that allows the users to focus on the tasks they have to perform rather than just managing their personal information. On-TIME is an example of PIMS, in which user tasks, as well as user data, are described in terms of explicit semantics that the user can share, i.e., by means of a so-called Personal Ontology, reflecting the user's view of the world and her personal interests.

The first component allows the user to navigate and edit the Personal Ontology, both at[Emanuele1] the conceptual and the instance level, with the capability of focusing on the instances with highest *level of activation*. These are the instances that are currently of most interest for the user, because of their nature and/or as a consequence of particular events, e.g. the user inspecting the instance, the reception of an email considered of interest. This component is based on coordinated multiple views, which allow the user to navigate the Personal Ontology by possibly switching from one view to the other, maintaining the same focus of attention. In particular, it provides both a tree view, which might be preferred by an expert user, and a graph view, which might be preferred by a less expert user, since it graphically provides an immediate idea of the relationships among the elements of the ontology.


Java Application Des...

On-Time AppBar

Task List

1. Confirm partecipation to HCI 2009 conference (100%)

2. Reserve a room for HCI 2009 conference in UK - London (90%)

3. Send an email to Alan with subject "preparation for HCI 2009" (75%)

4. Book a flight to London (50%)

5. Check the weather forescast for UK (45%)

more tasks...

File List

> Email from Alan Dix
- Add hotel Excelsior to ontology
- Add hotel "Excelsior" to ontology
- Add date "16/07/2009" as Alan's Birthday
- Add date "16/07/2009" as HCI 2009 start date

> Picture "DSC0049.jpg"

> Word document "listino.doc"

> Audio file "aSong.mp3"

> PDF file "Cocktail.pdf"

more files...

Fig. 1 - AppBar

The second component of the On-TIME user interface supports the user in the execution of her tasks. Since[Emanuele2] tasks are[Emanuele3] more of an abstract notion than a computer manageable entity, designing an interface to make them accessible, i.e. executable, is a challenge. Our solution is based on the idea of allowing the user to deal with her tasks like she does with files and traditional applications. For instance, she can execute a task by selecting it from a task list occurring in a dedicated application bar. In this case On-TIME would automatically infer the appropriate input for the task on the basis of the task definition and the instances level of activation. Alternatively, the user can execute a task by accessing an element of the Personal Ontology, and choosing among a list of tasks, that would take as input (or produce as output) the element itself. Note that in both cases, tasks suggested are those that are more likely to be performed. Specifically, these are the tasks that are more closely related to data having highest level of activation.

[Emanuele 4]We point out that, at this stage of the project, the intensional level of the Personal Ontology is not editable, but is set *a priori*: the user can only edit the extensional level, e.g. by adding new instances, modifying or deleting current ones. Also the task list is defined and, actually, contains only illustrative tasks. Editing the intensional level of the Personal Ontology and creating new tasks will be future steps of our work.

Demonstration highlights To illustrate the main aspects of the On-TIME user interface and to show how it can be used to access user's personal data and to support her while executing her tasks, we consider a typical researcher user scenario, assuming the existence of a suitable pre-defined personal ontology.

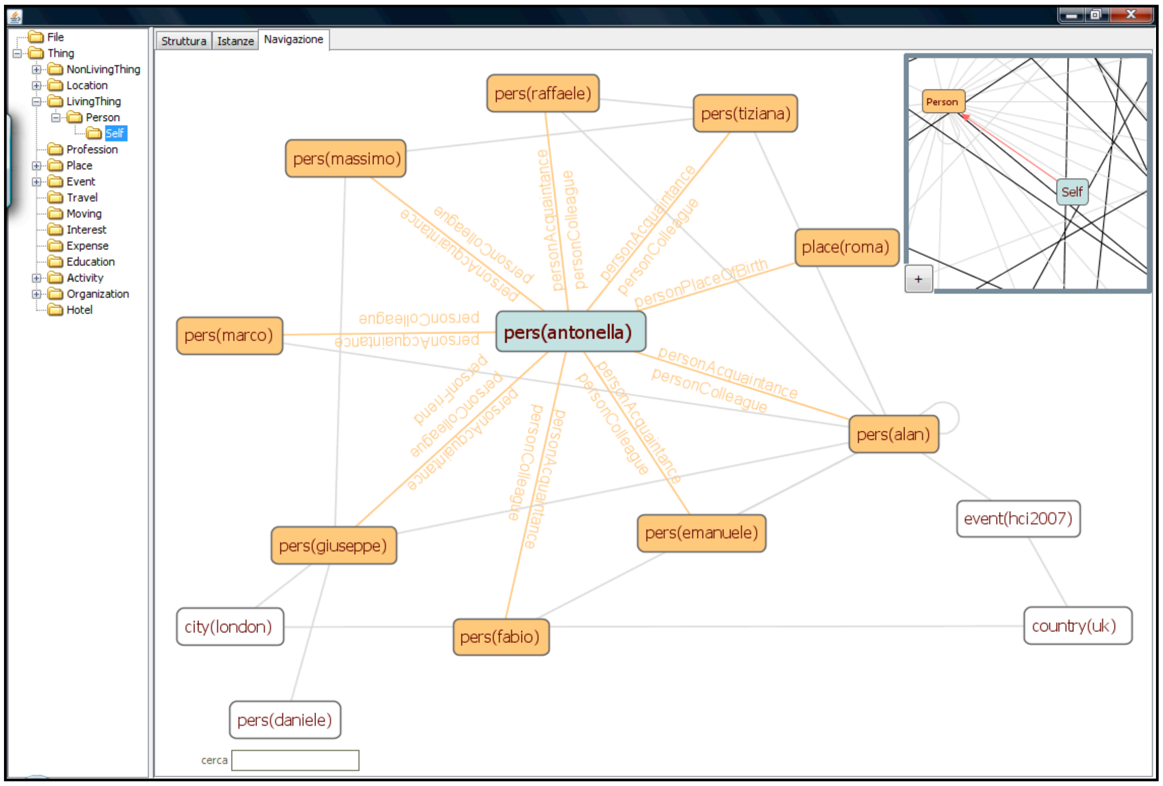


Fig. 2 – Graph view

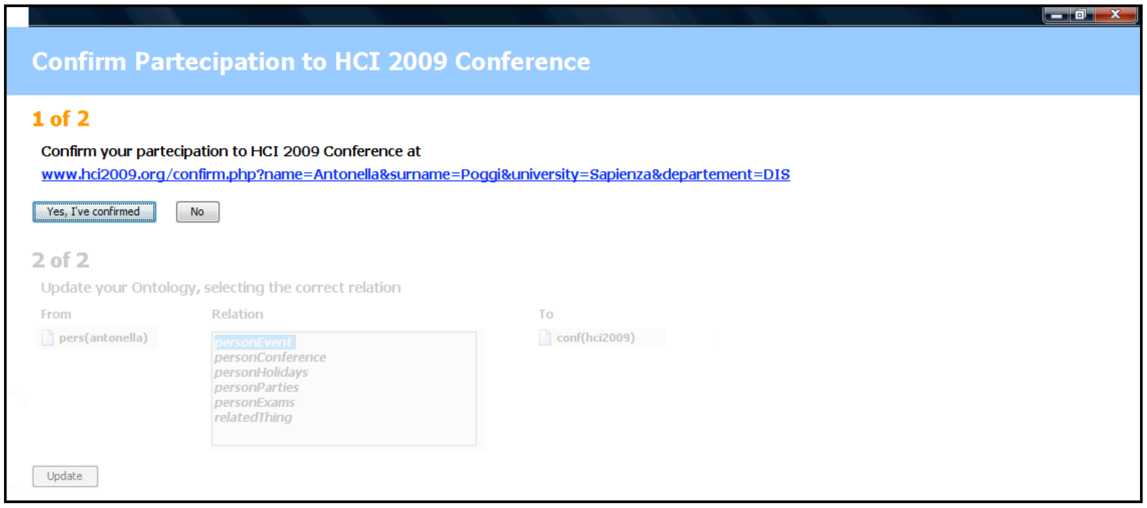
The remaining of this paper is structured as follows. Section 2 illustrates the On-TIME user interface through the researcher user scenario. Section 3 presents the results of the user evaluation carried out over the interface, and then Section 4 discusses related work.

2. A RESEARCHER USER SCENARIO

Suppose that Antonella works in a research lab that has recently installed a new fantastic fax and she needs to update all the documents where she used to indicate the fax number. This can be easily achieved with On-TIME, by simply updating the Personal Ontology. Hence, the starting point of her interaction with On-TIME is the AppBar [Fig. 1] installed on her desktop.

The *AppBar* component is the core of the system interface and the first component that appears to the user. This bar is divided into three sections. The first section contains a link, represented as a button, for each possible view of the *Personal Ontology Interface* (see below). The second section contains a list of tasks that are more likely to be willed to be performed. [Emanuele 5]The third section contains the list of files that the user is currently interacting with: for each of them a list of updates, suggested by the system, is proposed. More precisely, if candidate instances, or instance attributes, that are not currently in the ontology, were found within the newly saved documents, the user would be able to update the Personal Ontology associating to those instances the right semantics, e.g., a date could be her best friend's date of birth or an important event date.

The Personal Ontology Interface is a component providing three coordinated views[8] over the Personal Ontology, having a common section that is an indented list showing the concepts hierarchy tree. While the common section allows the user to select a specific concept quickly, for example to view its instances in details, the three views have each a different purpose, and are coordinated so that when the user switches from one view to the other, she keeps focusing on the same particular instance/concept. Specifically, the *Structure View* allows the user to investigate the details of the ontology structure, the *Instance View* to select an instance of a specific ontology concept to visualize and edit its details, and finally the *Navigation View* to browse the ontology through both its instances and concepts. Let us describe the latter in more details. It shows always two graphs, representing respectively connections among instances and concepts. Let us describe the latter in more details. It shows always two graphs, representing respectively connections among instances and concepts. One of the two graphs is always foreground and the other is visible in a frame positioned in the upper right corner of the screen, with the facility of switching between them. In order to optimize the extensional level browsing, we use the focus plus context technique[3] which allows the user to focus[Emanuele6] her attention on a specific instance or concept, by moving it on the centre of the screen, highlighting neighboring nodes (at a certain distance from the focus) and displaying the names of relations connecting them. This approach allows for maintaining the graph visualization as thinner as possible, succeeding in managing huge graphs[Emanuele7]. Moreover,[Emanuele8] the user can navigate the graph arbitrarily, and then return to the[Emanuele9] focused instance, by clicking on an anchor that is always visible.



Confirm Partecipation to HCI 2009 Conference

1 of 2

Confirm your partecipation to HCI 2009 Conference at
www.hci2009.org/confirm.php?name=Antonella&surname=Poggi&university=Sapienza&departement=DIS

2 of 2

Update your Ontology, selecting the correct relation

From	Relation	To
pers(antonella)	personEvent	conf(hci2009)
	personConference	
	personHolidays	
	personParties	
	personExams	
	relatedThing	

Fig. 3 - Task execution window

Coming back to our scenario, by clicking on the *AppBar* appropriate button, Antonella accesses the Navigation View to browse the Personal Ontology. Doing so, the ontology graph appears foreground, centered on the instance *pers(antonella)* [Fig. 2] denoting herself, (i.e., the system owner). Then, she verifies all relations connecting her with other ontology instances. In particular, she now switches to the *Instance View*, that according to the coordinated multiple views paradigm, is focused on *pers(antonella)*. Thus, Antonella immediately accesses the attribute *personFax* denoting the fax number that she is using, and she updates it. Note that, while saving the changes, the system ensures that the information currently contained in the Personal Ontology is not contradictory [4].

Suppose now that Antonella moves her attention back to the *AppBar*. She would then notice that On-TIME suggests the task "*Confirm participation to HCI2009 conference*". Actually, the Task Inferencer Module (a specific component of the system) proposes such a task, because Antonella just received an email request of confirmation by the HCI2009 organizers, which was detected by the system, and made increase the activation level of the instance denoting the event HCI2009. Moreover, it already happened several times that Antonella checked the correctness of her personal details before confirming her participation to a conference, in order to provide up-to-date contact details.

Antonella takes then advantage of this smart suggestion by the system, and executes the task by clicking on the relative link in the *AppBar* which opens a new window [Fig. 3]. In our scenario, the selected task is then performed in two steps, which will be executed by a wizard. The first step is the reservation confirmation: clicking on the proposed link, a web page is opened where Antonella checks her data (automatically returned by the system) and makes the reservation. Then Antonella returns to the task execution window and clicks on the button "*Yes I've confirmed*", specifying in this way that the external action was completed succesfully. Then, the system completes the task execution by updating the ontology with information about the HCI2009 conference.

3. USER EVALUATION

The On-TIME development is based on a human-centered design, with an iterative-incremental development cycle requiring the involvement of all project stakeholders, including the final users of the application. In particular, the interface development, was made with the implementation of richer and richer prototypes up to the final version presented here. During the interface realization (both at prototype and implementation level) we have considered many technologies and visualization metodologies for ontology and tasks, discussing them in dedicated groups among system developers. Ideas like the use of coordinated multiple views technique, for coordinating the various ontology views, or the focus plus context technique[3] for graph visualization, are the results of these internal meetings, where we have examined many prototypes and assessed all possible improvements up to the actual state. After having traced the system development route, in order to verify the correctness of our choices, we involved also users in many tests. These tests were performed on specific user actions, all involving the system interface. We then asked to our users to give their sensations during the interface use.

To be more precise, we adopted Observational Techniques like *Cooperative Evaluation* [5], which allows for having a direct interaction between tester and developers. The cooperative evaluation tests were pretty positive, and confirmed that our choices were right and exhaustive for the user needs. We also got suggestions for improvements. In particular, the cooperative multiple view was appreciated by all users, since it was judged useful and necessary in order to avoid disorientation and incoherent screens. The *Navigation View* was appreciated too: it was defined the most usable and immediate one, so that also users with no expertise not familiar with ontologies, were able to correctly interact with the system. Furthermore, we tested the *AppBar* component, to check whether the users would like to have it always visible. We also tested the projectual choices, taken during our project meetings, like the partition in three different areas. Test results were positive and confirmed the effectiveness of our choices. We finally executed several tests on the Task Manager Interface [Fig. 3], a crucial feature of On-TIME. The wizard adoption has been approved with favour and encouraged us to continue with this choice.

4. RELATED WORK

To the best of our knowledge, there exists no system in literature that is fully comparable with On-TIME: there are only ontology visualization tools or tasks-oriented systems. An interesting survey on ontology visualization techniques was made by Akrivi Katifori et al. in [7], where many different metodologies (and related systems) were considered, namely, *Indented list*, *Node-link and tree*, *Zoomable*, *Space-Filling*, *Focus + context* or *distortion*, *3D information landscapes*. These were a starting point for our work, which combines many of their positive aspects, such as the use of different colors, the facility of controlling the amount of loaded nodes, the facility of moving within the graph, the nodes search, etc. However, the On-TIME fundamentally differs from all these systems, in that it is directly connected to an inferential engine (i.e. QuOnto [1]), and is not a simple ontology viewer.

Concerning task-oriented systems, the closest to our work is *Activity-Centered Task Assistant* (ACTA) [2], recently implemented as a Microsoft Outlook add-in. In this system, a user's task, named *ACTA activity*, is represented as a pre-structured container, that can be created inside the email folder hierarchy. Focusing on the interface, the only similarity between ACTA and On-TIME is the task list, named TV in ACTA, from which activities can be launched. However, the list of activities in ACTA is managed by the user, who can insert new to-dos or drag email messages into the list. On the contrary, the On-TIME task list is automatically inferred, taking into account the user current interests.

5. REFERENCES

[1] A. Acciarri, D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, M. Palmieri, and R. Rosati. QuOnto: Querying ontologies. In Proc. of the 20th Nat. Conf. on Artificial Intelligence (AAAI 2005), pages 1670-1671, 2005.

[2] V. Bellotti, J. Thornton, A. Chin, D. Schiano and N. Good. TV-ACTA: Embedding an Activity-Centered Interface for Task Management in Email, Fourth Conference on Email and Anti-Spam CEAS 2007 Aug 2-3, 2007, Mountain View, California.

[3] S.K. Card, J.D. Mackinlay, and B. Shneiderman (Eds.). Readings in Information Visualization: Using Vision to Think, pp. 1-34, Morgan Kaufmann Publishers, San Francisco, Califomia, 1999.

[4] G. De Giacomo, M. Lenzerini, A. Poggi, and R. Rosati. On instance-level update and erasure in description logic ontologies. Special issue of the Journal of Logic and Computation (JLC). To Appear.

[5] A. Dix, J. Finlay, G.D. Abowd, and R. Beale. Human-Computer Interaction, Pearson Prentice Hall, third edition, 2004.

[6] A. Dix, A. Katifori, A. Poggi, T. Catarci, Y. Ioannidis, G. Lepouras, and M. Mora. From Information to Interaction: in Pursuit of Task-centred Information Management. In DELOS Conference 2007, 2007.

[7] A. Katifori, C. Halatsis, G. Lepouras, C. Vassilakis, C., and E. Giannopoulou. Ontology visualization methods - A survey. ACM Comput. Surv. 39, 4, October, 2007.

[8] C. North and B. Shneiderman. "A taxonomy of multiple window coordinations", University of Maryland, College Park, Dept of Computer Science Technical Report #CS-TR-3854, 1997.