

Mindmap-Inspired Semantic Personal Information Management

Jenny Rompa Christos Tryfonopoulos Costas Vassilakis George Lepouras
 Dept. of Informatics and Telecommunications,
 University of Peloponnese, Karaiskaki Str, GR22100 Tripoli, Greece
 {jr, trifon, costas, gl}@uop.gr

ABSTRACT

Users nowadays need to manage large amounts of information, including documents, e-mails, contacts, and multimedia content. To facilitate the tasks of organisation, maintenance, and retrieval of personal information, a number of semantics-based methods have emerged; these methods employ (personal) ontologies as an underlying infrastructure for organising and querying the personal information space. In this paper we present OntoFM, a novel personal information management tool that offers a mindmap-inspired interface to facilitate user interactions with the information base. Besides serving as an information retrieval aid, OntoFM allows the user to specify and update the semantic links between information items, constituting thus a complete personal information management tool.

Keywords

personal information management, mindmaps, file manager, ontology, visualisation

1. INTRODUCTION

The advances in information technology have enabled the production and exchange of vast amounts of information, in the form of documents, e-mails, web pages, pictures, and videos. The advances however in techniques for managing, organising, and retrieving this information have not kept up with this production pace, making it extremely hard for users to handle and organise their personal information. *Personal information management* (PIM) aims to facilitate these tasks by offering specialised and effective tools and techniques [3,12,13]. One of the most promising approaches employed by many proposals is the use of *personal ontologies*, which are used to link user-level concepts (e.g., persons, places, events, etc.) to files and directories, and to facilitate the *semantic management* and retrieval of the latter. The establishment of these links though is not sufficient, since users need tools that would exploit the potential offered by the ontology to help them perform their information management and retrieval tasks. *Mindmaps* are diagrams used to visually outline information and may offer valuable assistance to users in this context, by (i) focusing on concepts related to the keywords entered by the user and (ii) allowing the user to *navigate* through the semantic links between concepts and information items. In this demonstration,

we present a novel personal ontology-based *file manager* that offers a *mindmap-oriented visualisation* to support user navigation within the personal information base. Browsing aids are complemented with search mechanisms, while the user is also offered with the potential to establish links (either manually or in a semi-automatic way) between ontology concepts and files/directories, constituting thus a *fully functional* PIM environment. The presented work differs significantly from previous approaches that exploit ontologies to retrieve information [1,2,8,12] since it (i) introduces a mindmap-oriented visualisation, (ii) utilises a full-text search/indexing/result scoring engine, (iii) integrates semi-automatic update capabilities for the personal ontology, and (iv) emphasises file management.

2. PROTOTYPE OVERVIEW

In this section, we present the high-level architectural design of the OntoFM file manager and describe its key functionality based on user tasks. The OntoFM Protégé plug-in, a short video demonstration, and more screenshots of the OntoFM file manager are available under the GNU Lesser General Public License in the OntoFM project page at <http://hci-vr.dit.uop.gr/ontofm>.

2.1 Architecture

OntoFM has been built according to a layered architecture, as illustrated in Figure 1.

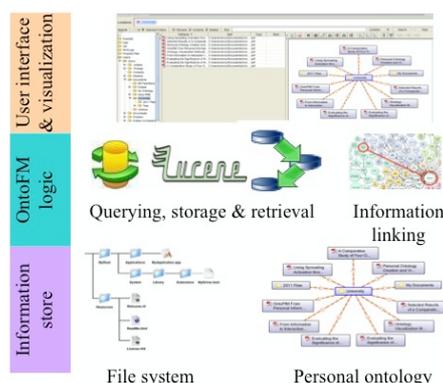


Figure 1. OntoFM architecture.

The bottom layer is the information store, comprising of the file system (as offered by the underlying operating system) and the personal ontology, which hosts the concepts that are important to the user, as well as the associations between these concepts and the file system elements (files and directories). The personal ontology used in this version of OntoFM is an updated version of the one presented in [11], to allow the support of the ontology population and maintenance tasks. Accessing and retrieving

(c) 2014, Copyright is with the authors. Published in Proceeding of the 17th International Conference on Extending Database Technology (EDBT 2014) on OpenProceedings.org. Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0

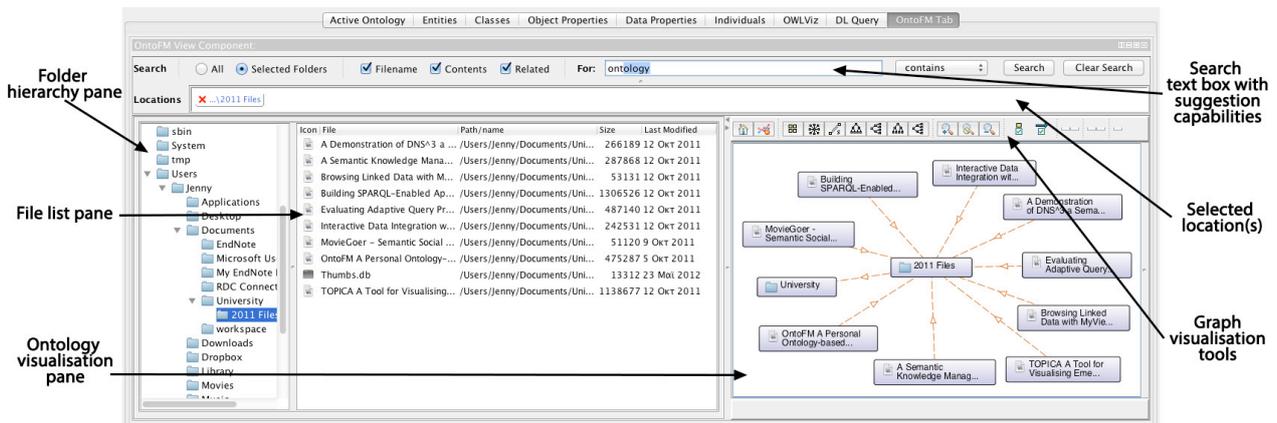


Figure 2. OntoFM user interface components when browsing through folders to locate a file.

ontology elements is implemented using the Protégé¹ open source ontology editor and knowledge-base framework, thus exploiting both the functionality and the visualisation features offered.

On top of the information store lays the OntoFM logic layer, which implements: (i) information querying, i.e., finding file system items associated with given concepts, (ii) information linkage and ontology population, i.e., allowing the user to associate (manually or semi-automatically) selected items or item hierarchies with concepts, and (iii) information update and synchronisation with the information store. As accessing the personal ontology through the Protégé-OWL API was inefficient, the full-text search engine Lucene² is used for the indexing and fast retrieval of ontology elements, file/folder names, and associated metadata. Lucene is initialised at the OntoFM bootstrapping phase and is used to retrieve and index all personal ontology data to enable fast access during OntoFM usage.

Finally, the top layer (user interface and visualisation) collects user input and displays results to the user. The ontology visualisation is based on the Ontograf³ plug-in, which has been extended to support the additional functionality offered by the file manager and tailored to display the ontology according to the mindmap paradigm. Mindmap-inspired ontology visualisation and interaction allows users to manipulate the ontology without requiring detailed technical knowledge in ontology building. In this way, we are able to offer semantic expressiveness by exploiting the inherent mindmapping connections to enrich and present the user's information space.

2.2 Functionality

Two highly important functions (typically implemented by common file managers) are browsing and searching through files and directories. Depending on what information is known to the user with respect to the search target, the file location task can be categorised in two major categories: (a) *perfect exploration*, where the user knows the requested file's name and location, and (b) *limited/incomplete/partial exploration*, where the user has incomplete information about the file she wants to locate. When the user performs a partial exploration, she might have varying information available to assist her in locating the file, as she might

(i) know the file location but not the filename, (ii) know the filename but not the location, and (iii) know some metadata about the file (e.g. date, keywords from content, author, etc.) but neither the filename nor the location. In the first case, the user will simply navigate to the location and browse to find the file in question, while in the second case she needs to resort to a search mechanism to locate the file. Additional problems for partial exploration arise in the third case, where traditional file managers present the "feast of famine" problem: they may return an excessive number of results (since too many files will typically match the search criteria) or no results at all (since the criteria are wrong or metadata for the file is not available). The introduction of semantic searching through the use of a personal ontology as a semantic information repository may alleviate this problem and prove a valuable PIM tool.

Browsing task. To maintain the level of user familiarity at the highest level possible, OntoFM offers the ability to browse folders and files within the physical file system hierarchy through its user interface as presented in Figure 2. When the file location is known to the user, the folder hierarchy (left-side pane) may be used for browsing and locating the requested file. As shown in this figure, only files of the selected folder (and not of its subfolders) are displayed in the file list (middle pane). Moreover, the ontology visualisation (right-side pane) shows the corresponding folder instance surrounded by file instances and concepts/instances directly related to it, following a mindmap-inspired presentation. The user may expand or collapse a node by double clicking on it; upon expansion the related concepts of the node appear in the ontology visualisation pane. Relations between nodes are represented as arrows, indicating the direction of relationships. Icon (resp. colour) diversification is also used to assist the user distinguish the type of file (resp. relationship) presented in the pane. Additionally, the user may control the appearance of the ontology graph by centering on, or creating a tree under/over a concept and moving concepts/instances to new positions. Display refinement is also available by introducing type constraints on concepts and relations of the personal ontology. The user may also exploit the multiple-select functionality (offered by the user interface) to browse the contents and associated relations of more than one folders simultaneously.

Search task. If the file name/location is not known to the user, she may utilise the search text box which features auto-suggestion of candidate search terms using ontology concepts/instances to recover the file in question. Note that since the personal ontology contains the File (resp. Folder) concept, searching is also

¹ <http://protege.stanford.edu>

² <http://lucene.apache.org/>

³ <http://protegewiki.stanford.edu/wiki/OntoGraf>

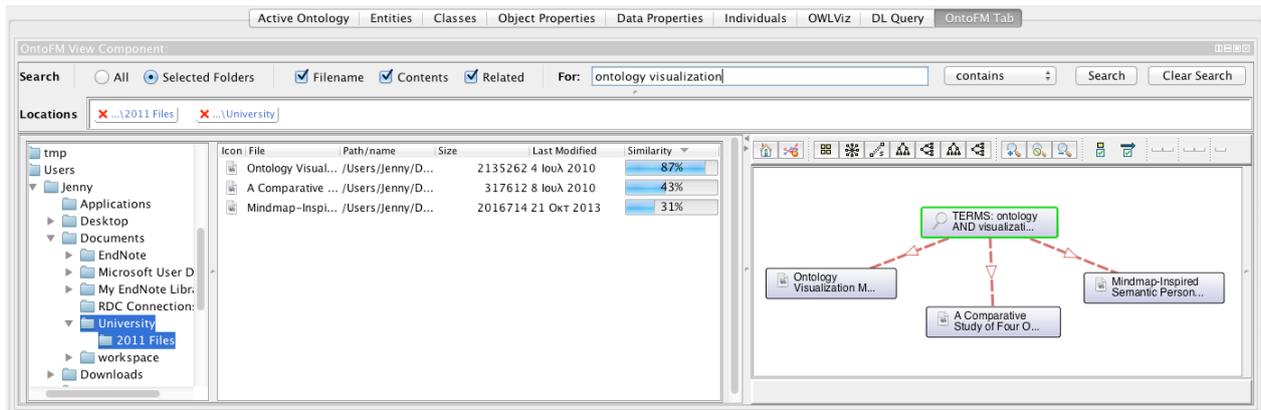


Figure 3. Searching for files related to keywords “ontology” and “visualization” by limiting the semantic search to two selected folders. The entered terms appear in the ontology visualisation pane along with directly connected concept instances.

performed within the file (resp. folder) names and their metadata (e.g. author, date of creation, keywords, etc.). The file list pane displays all file instances that are related to the search terms and are contained in the ontology. Sorting of search results may be performed by filename, path name, size, last modification date, and similarity score (scoring of files is implemented using the Vector Space Model by matching document metadata against query terms). Result refinement is also possible by introducing metadata constraints on concepts of the ontology (e.g., creation/modification date on the Date concept, or file type on the Type concept) to narrow down the result space. Figure 3 shows a user search example that returns three PDF files.

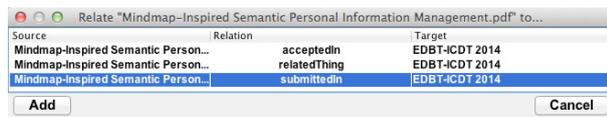


Figure 4. Relating a file to an instance by drag and drop.

Ontology maintenance task. In order for the personal ontology to be constantly up-to-date and meet the current needs of each user, ontology editing mechanisms have been developed to assist the user in maintaining the personal ontology in a semi-automatic way. OntoFM provides basic ontology editing functionalities, using familiar computer actions, through the addition of new concepts and the linking of files/directories to existing concepts, allowing the user to maintain her personal ontology without the need of knowing the ontology structure and restrictions. Relating a file to a concept/instance of the personal ontology may be implemented in two ways: (i) by dragging and dropping a file on an ontology node shown in the ontology visualisation pane and (ii) in a menu-based fashion, by selecting the “Relate to...” option from the context menu of a file. As soon as the user selects a concept or relationship to be created, OntoFM creates automatically all the needed instances and relations in the personal ontology, even those who are indirectly related.

In the drag and drop action, when a file is dropped on an instance, a popup window (an example of which is shown in Figure 4) showing a list with the possible relations between the two instances (as determined by the ontology schema) is shown. Contrary, if the file is dropped on a concept, the file manager updates the ontology using the *is-a* relation between the two elements (since this is the only possible relation) and notifies the user for this action.

When the “Relate to...” option from context menu is used, a dialog window (an example of which is shown in Figure 5) is opened. Then, the user selects the desired relation from a *candidate relations* combobox and the desired concept/instance from the (automatically updated) list of concepts/instances that are valid for the selected relation. Note that when the *is-a* relation is selected, the list of objects in the *candidate targets* list contains only concepts, while in the rest of the cases contains instances. Browsing and deletion of connections for a file/folder is possible through the *existing relations* list shown in Figure 5 [notice that system relations are read-only (indicated by displaying them as greyed out) and thus cannot be deleted]. Finally note that basic ontology updating tasks based on file/folder (meta-)data may also be performed in a semi-automatic way with minimal user intervention (e.g., email or photo metadata) through appropriate plugins already implemented.

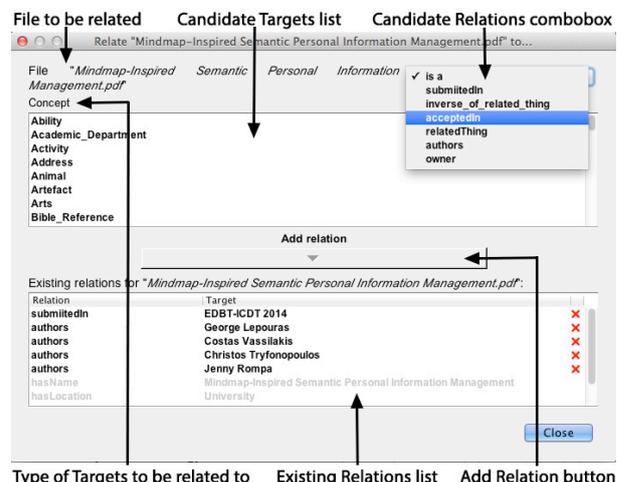


Figure 5. Relating a file to a concept/instance by “Relate to...”.

2.3 Demonstration Scenario

In our demonstration, we will present a full-fledged usage scenario for the OntoFM prototype and invite users for a hands-on experience with OntoFM. Attendees will be assisted to install the tool to their laptop (together with a prototype personal ontology) and will be invited to participate in our long-term user study.

During the demonstration users will browse through the file system both in a conventional and in a multiple-selection way. During the exploration users will issue searches on the personal ontology and related files, and will be able to retrieve the matching content. During search, the autocomplete feature will assist users to issue or refine the query of interest by adding/removing search terms. Upon retrieval of the requested data, users will be able to sort them in a number of ways (e.g., by relevance score) and view the mindmap representation of the result. Users will also be able to populate and maintain the personal ontology by adding, editing, or deleting semantic links among files/folders and concepts/instances through drag-and-drop actions in the graphical pane. Finally, users will navigate in the personal ontology using different node/relationship filters and visualisations, including the mindmap representation. All actions will be facilitated through a graphical user interface.

3. RELATED WORK

There are only a handful of ontology-based PIM tools [1,2,8] available for the end user, with the alpha version of CLUUG⁴ launched by Gnowsis.com being one of the most prominent ones. CLUUG offers a desktop client with item tagging capabilities within Microsoft Outlook, Microsoft Office, Internet Explorer, and the Windows file-system. The product is based on results of the European Project NEPOMUK⁵, aiming to develop an extension of the personal desktop that will support personal information management along with sharing and exchange of information across social and organisational relations. Apart from the CLUUG system, the work in [8] also employs semantic web techniques to enhance the searching functionalities of the user's desktop, while Haystack [1] proposes the use of a semantic network (constantly enriched through user observation) to facilitate search and retrieval. Along the same lines, SEMEX [2] treats user storage as a network consisting of instances and associations between these instances. Although these approaches employ semantic information to perform PIM, they can only be used as an additional tool to help the user locate a file/folder. Contrary our approach gives a complete PIM solution that does not need additional tools such as file browsers/managers. Moreover, the mindmap-oriented visualisation assists users in understanding and correlating the retrieved information with the associated concepts, while the result scoring based on state-of-the-art Information Retrieval metrics allows for better result ranking.

In a different strand of work the GoNTogle [6] system offers a semantic annotation and search mechanism targeted specifically to document annotations, while PerK [14] aims at ranked keyword retrieval based on user preferences. Moreover in [7,9] an ontology depicting the user's personal information space and tools for managing and visualising this personal ontology was presented.

Finally, ideas for applying ontologies to the search and annotation of documents also appear in web-based [5] and multimedia information [4] management. In [5] the interaction between user profiles and conceptual search in documents (indexed under both keywords and concepts) is investigated, while in [4] novel semantic image and video annotation tools are introduced.

⁴ CLUUG was later renamed to Refinder and transformed to an online social tool for collecting and sharing information in online applications.

⁵ <http://nepomuk.semanticdesktop.org/>

4. OUTLOOK

We have presented a prototype personal information management tool based on personal ontologies and mindmap-inspired information management/visualisation. We are currently extending OntoFM functionality to support (i) the automatic import of documents directly from online libraries, (ii) the weighting of concepts based on recent/frequent user activity by using spreading activation [10], (iii) the sharing of the user personal ontology among different personal devices by employing cloud-based storage/maintenance, (iv) a clutter-free mindmap visualisation and (v) extensive user studies.

5. REFERENCES

- [1] Adar, E., Karger, D., and Stein, L. Haystack: per-user information environments. In *CIKM*, 1999.
- [2] Cai, Y., Dong, X.L., Halevy, A., Liu, J.M., and Madhavan, J. Personal information management with SEMEX. In *ACM SIGMOD*, 2005.
- [3] Chirita, A., Gavriiloaie, R., Ghita, S., Nejdil, W., and Paiu, R. Activity based metadata for semantic desktop search. In *ESWC*, 2005.
- [4] Dasiopoulou, S., Giannakidou, E., Litos, G., Malasioti, P., and Kompatsiaris, Y. A survey of semantic image and video annotation tools. In *Knowledge-driven Multimedia Information Extraction and Ontology Evolution*, 2011.
- [5] Gauch, S., Chaffee, J., and Pretschner, A. Ontology-based personalized search and browsing. In *Web Intelligence and Agent Systems*, 2003.
- [6] Giannopoulos, G., Bikakis, N., Dalamagas, T., and Sellis, T. GoNTogle: a tool for semantic annotation and search. In *ESWC*, 2010.
- [7] Golemati, M., Katifori, A., Vassilakis, C., Lepouras, G., Halatsis, C., Creating an Ontology for the User Profile: Method and Applications. In *IEEE RCIS*, 2007.
- [8] Katifori, A., Halatsis, C., Lepouras, G., Vassilakis, C., and Giannopoulos, E. Ontology visualization methods - A survey. In *ACM Computing Surveys*, 2007.
- [9] Katifori, A., Vassilakis, C., Daradimos, E., Lepouras, G., Ioannidis, Y., Dix, A., Poggi, A., Catarci, T. Personal Ontology Creation and Visualization for a Personal Interaction Management System. In *PIM (CHI Workshops)*, 2008.
- [10] Katifori, A., Vassilakis, C., and Dix, A. Using Spreading Activation through Ontologies to Support Personal Information Management. In *CSKGOI*, 2008.
- [11] Lepouras, G., Dix, A., Katifori, V., Catarci, T., Habegger, B., Poggi, A., and Ioannidis, Y. OntoPIM: From Personal Information Management to Task Information Management. In *PIM (ACM SIGIR Workshop)*, 2006.
- [12] Rompa J., Lepouras G., Vassilakis C., and Tryfonopoulos C. OntoFM: A Personal Ontology-based File Manager for the Desktop. In *ISWC*, 2011.
- [13] Sauermann, L. and Heim, D. Evaluating long-term use of the Gnowsis Semantic Desktop for PIM. In *ISWC*, 2008.
- [14] Stefanidis, K., Drosou, M., and Pitoura, E. PerK: personalized keyword search in relational databases through preferences. In *EDBT*, 2010.