

KASIMIR – Work process embedded task management leveraging the Semantic Desktop

Olaf Grebner¹, Ernie Ong², Uwe Riss¹

SAP Research, Karlsruhe & Belfast

¹ Vincenz-Priessnitz-Str. 1, 76131 Karlsruhe

² TEIC Building - University of Ulster, BT37 0QB Newtownabbey, U.K.
firstname.lastname@sap.com

Abstract: The effectiveness of semantic applications can be increased significantly if the system does not urge the users to take care of explicit semantic annotation creation. Today, semantic annotation creation is a well known bottleneck in semantic applications for enterprises. On the other hand semantic technologies have already proven to facilitate, e.g., a more efficient information management especially for knowledge workers (KWs). We pursue in this paper the central idea of using task management as a means for extracting semantic annotations out of the daily work activities in a way that is associated with minimal effort for the KW. To ensure the success of such an approach, a broad acceptance of the task management application by the KWs is a precondition. For this, we present the KASIMIR prototype that embeds task management into the daily work activities of a KW. It facilitates the creation of semantic annotations in form of attributes as well as relations between tasks, users and information objects such as emails, websites, documents, etc. in a manner that is seamlessly integrated into the KW's work activities. To this end we integrate task management plug-ins into desktop applications, e.g., emails clients and web browsers. These plug-ins interact with a sidebar that creates a consolidated overview of existing tasks and their relations. The underlying infrastructure of the Nepomuk Semantic Desktop¹ leverages semantic web technologies to ensure the efficient management of semantic data.

1 Introduction

In this paper we show how a task management system can leverage semantic web technologies to support Knowledge Workers (KWs) more efficiently. Knowledge work essentially consists of the organization of information objects (IOs) on the desktop but of course also in the actual work with these objects [Py05]. According to [Bo04] we call these supporting and production activities, respectively.

Supporting activities are largely related to Personal Information Management (PIM) [Be04], which plays a decisive role in knowledge work. Task management (TM) as

¹ The presented work is part of the Nepomuk Integrated Project funded by the E.C. (<http://nepomuk.semanticdesktop.org>).

supporting activity is a central part of PIM since tasks represent a central means to structure the personal work. In the PIM domain, tasks are mainly organized by emails. Consequently, in the existing literature task management is most often discussed in conjunction with email usage as part of PIM.

Nevertheless, we face the basic challenge that current approaches to task management do not provide an effective integration with the work context of the KWs and the tools they employ in their daily work. In particular, existing TM software does not efficiently support information management, e.g., it requires KWs to copy and paste information from different application contexts to their TM tools [Be02]. This results in duplication of information and work. It increases the cognitive load and introduces additional administrative overhead for the KW and reduces its relevance and value to the KW and thus their motivation to use it. Compared to the required effort these tools do not provide sufficient benefits to knowledge workers.

Semantic web technologies represent an emerging domain in information systems engineering that hold a large potential for the Knowledge Workers and their needs since they provide a framework to better manage, integrate and find information and allow for inferences based on semantic annotations². However, when looking at the creation of semantic annotations in an enterprise scenario, today's solutions suffer from the problem that creating semantic annotations is costly since it requires the KW or a dedicated knowledge manager to invest significant effort. Whereas ontologies can be rather efficiently built today, the creation of instances requires a permanent effort. Generally knowledge workers avoid such activities since they do not directly contribute to their work. The extra maintenance effort – and especially the *perceived* extra work [Cr05] – leads to weak acceptance among the workforce already being under constant time pressure. Another practiced alternative consists of 'mining' semantic annotations using data and text mining methods (see [APS06] and references there). However, this alternative still has to prove its applicability in the mainstream enterprise domain.

In the current approach we intend to realize a TM system with semantic web technologies. Driven by the application domain of building a TM system, the guiding principle that we want to apply here is the idea that human action (as production activity) and knowledge (including semantics) are closely interwoven [Ri06]. This leads us to the approach of annotations exploiting the production activities of a knowledge worker as the primary means to derive semantic annotations. This addresses the above mentioned central challenge of today's semantic web technologies. Integrating semantics into a TM system goes way beyond the usual coupling of email tools and TM systems as part of PIM as reported in e.g. [Be04].

In the following, we focus on the aspect of *seamless* semantic annotations creation out of the work activities of the knowledge worker. The remaining paper is structured as follows: First, we describe the close relation of TM with information objects on the desktop in regard to supporting knowledge work. Second, we present our KASIMIR prototype and approach. Third, we show how semantic annotations results from working with the TM system. Finally, we conclude with some ideas for further work.

² By the term *semantic annotation* we denote ontology-based metadata [Ha05].

2 Tasks and information objects dominate the KWs' daily work

The KW's daily work process shows a highly dynamic working style [Da05], so that professional task handling becomes obligatory in particular regarding existing task overflow [Ki00].

We see tasks as the central hub for personal information management and knowledge work organization. This is supported by a study in the PIM domain [Bo04] that showed that "short-term production activity" (task) is the most used organizational dimension. Beneath tasks, concepts such as topics, events, persons and locations represent important *structuring information objects* used by KWs to organize their personal information.

As well, the KWs' daily work process is dominated by sending and processing emails, web browsing, working on documents and doing calculations in spreadsheets [CHW04]. Thus emails, websites, documents, spreadsheets, etc. represent central, *document-like information objects* on the KW's desktop. The importance of them can be seen by the fact that KWs are highly proficient in handling them today. E.g., many KWs are familiar with email and calendars as their central PIM tools. The same holds for the web browser.

From a TM point-of-view, Figure 1 depicts the concepts of structuring and document-like information objects on the desktop that are related to the task concept.

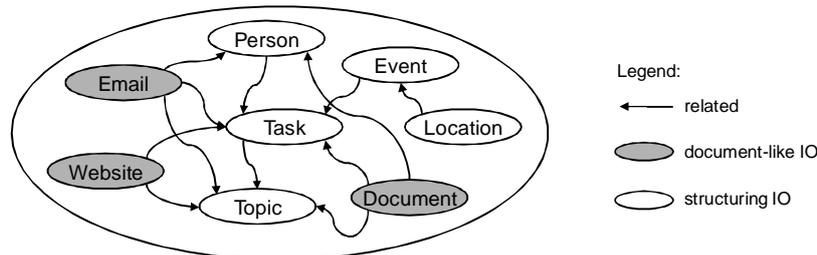


Figure 1: Task-related concepts of structuring and document-like objects on the desktop.

These relations often result from the fact that working with IOs trigger tasks. Our assumption is that a task usually does not stand for itself without having a cause or trigger that is the starting point for the intended activity. Such causes can be telephone calls, in which work issues are discussed, or information objects that require further processing. On a desktop, there are numerous document-like information objects that can represent such a cause. For example, a KW wants to follow up on an email and considers this a personal task. The same holds true for interesting parts of a website and calendar items or even parts of a text. In an enterprise application scenario, a business object can be such a cause. For example, a purchase order that lacks an approved supplier might lead to a task for the responsible buyer.

Conversely, a task can also enrich other document-like information objects as it is a source for further task context information. For example, when stumbling upon a document, a KW finds out that a task is attached to it. The KW can access this task and thus obtains further information related to the document-like information object.

Leveraging these relations, we propose *requirements* for a workplace-embedded TM:

- *Consolidate task-related information into tasks* – Information and metadata extracted from IOs must be transferred respectively referenced to the triggered (i.e., relevant) tasks. This enables a consolidated view on tasks bringing together task information from different applications.
- *Seamless integration of tasks into applications* – TM functionality must be integrated into those applications that handle task-relevant IOs. Thus, the KW can perform TM activities directly from these applications. This reduces interruptions of a KW's daily work process.

This integration brings the tasks to the desktop – as opposed to an incidental and administrative overhead – as a principle work paradigm which allows the access and manipulation of tasks from numerous application contexts.

3 KASIMIR – Integrate task management in a KW's daily work

To address these requirements, we propose a combination of a set of *plug-ins for desktop applications* and a structured TM application in the form of a *sidebar*.

The desktop *application plug-ins* shall enable the handling of tasks in the context of already existing information objects so that KWs can perform TM using their well-known environment of existing applications.

First, to enable systematic TM across application boundaries, the desktop application plug-ins transfer as much information as possible from information objects to tasks. The transferred information requires the KW to supplement minimal additional information. Ideally the task creation should work without any additional effort by the KW. For example, the email plug-in extracts some email information and fills task properties accordingly, for example the subject of the email results in the task name and selected email content results in a task description. The KW can additionally provide other information, e.g., a due date, but is not required to do so for creating a task.

Second, to reduce interruptions, task information is displayed within the context of the respective PIM application and the information objects therein. The KW can also interact with the displayed task information. For example, a KW is able to see in the email inbox whether a task is attached to an email and what information it carries.

In combination with the plug-ins, the TM *sidebar* enables efficient handling of existing tasks and provides detailed task information. It shows tasks that emanate from different information objects, i.e. it provides a consolidated overview on all available tasks on the desktop. This is needed since it does not make sense to overload the plug-ins with TM functionality that the KWs would not need in their particular work situation. The TM sidebar closely interacts with the plug-ins, i.e., the plug-ins can control the TM sidebar remotely. For example, when a KW reads an email related to a task, the TM sidebar shows the details of the particular task.

Our preliminary user study showed that *email, calendar and internet browser* are those applications for information objects with which KWs spend most of their working time. In the following we present the implementation of plug-ins for these applications. Afterwards we present details of the TM sidebar.

3.1 Email client & calendar plug-in – Link tasks with emails & calendar items

To integrate TM with the KW's daily work process in the email client, we developed a Microsoft Outlook plug-in whose functions address the requirements of section 3.

First, emails are linked to tasks by enabling the KW to *create tasks from an email*, i.e., triggering tasks from information objects. The email information is used to pre-populate the task with information. For example, the *email sender* is added per default as a task observer, i.e., as a person interested in the task results, and the *subject* of the email leads to the task name. This automatic pre-population with email information opens emails to systematic TM enabling integrated information management via tasks.

Second, if a task is associated to an email this is indicated with the email. In the email inbox such an association is indicated by an icon at the respective email. Further information on the attached tasks is presented with the email, see Figure 2. When selecting one of these tasks, the KW sees task details on the task sidebar as explained in section 3.3. This helps the KW to understand the context of the email.

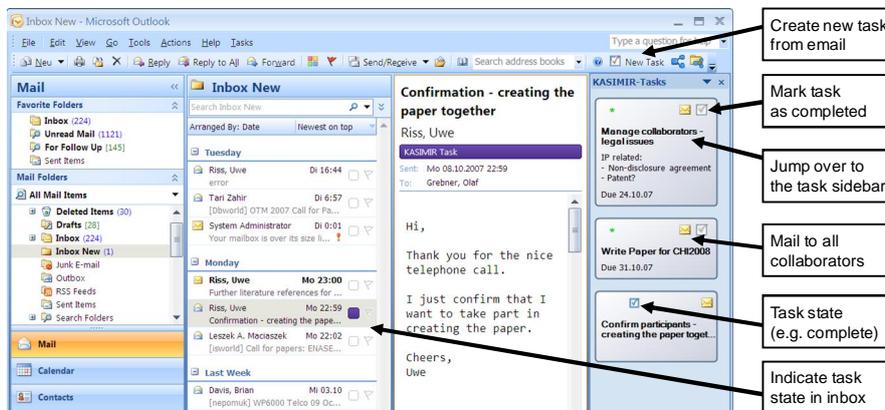


Figure 2: Task integration with email.

The plug-in for the calendar application of Microsoft Outlook works similarly³. Calendar items can also be related to tasks with due dates and duration. Information of a calendar item can be transferred to the task sidebar by creating a task from the calendar item. Conversely, existing task information is indicated at the calendar item for calendar items that have attached tasks.

³ For Microsoft Outlook, the implementation of the plug-in is similar. However, the calendar plug-in is a separate plug-in and can be implemented separately for other calendaring applications.

3.2 Internet browser plug-in – Link tasks with websites

Internet browsers are an almost ubiquitous tool for KWs today. Several tasks can arise when browsing through websites, e.g. the departmental website⁴. For example, reading further resources on a topic or to find interesting information to share with a colleague in the context of a common task. Our internet browser plug-in for Mozilla Firefox has two main functionalities.

First, the plug-in can help to *create tasks from a website or a selection of it* or to add website information to already existing tasks, see Figure 3. For example, when KWs find some information on a website that they want to further consider, they can create a task at this point. The website information, or the selection, is used to pre-populate the task. For example, the selected information is added to the task description and the link to the website is added for further reference. In this way, the website triggering a task is integrated with the TM and can be displayed in the task sidebar.

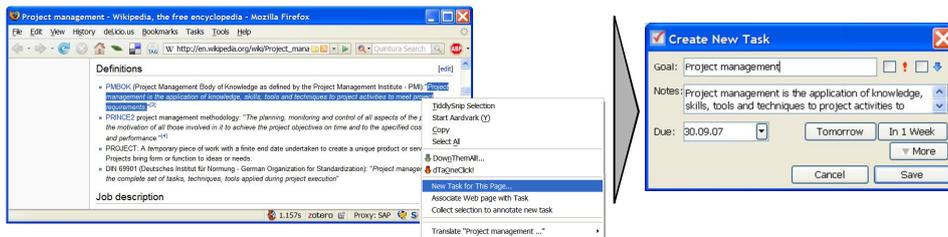


Figure 3: Task creation from a website.

Second, *tasks associated with websites are shown when the website is displayed in the browser*. As well, tasks can be associated with selected information snippets within the website. When browsing a website, the KW is able to see a list of tasks that comprise the reference to the website. This integrates tasks into the working environment of the KW. The KW can choose to follow up any pending tasks directly by clicking the task information. The task sidebar can show further details of the task.

3.3 Task management sidebar – Present a consolidated view on tasks

The *TM sidebar* serves as the primary front-end of the TM system, see Figure 4. It supplements the application plug-ins and enables efficient handling of existing tasks and displays detailed task information. The KWs use the task list to gather an overview of their tasks. The task list can display tasks and subtasks in several levels of detail. Beneath the task list, the task details view shows a single task's properties. Basic properties of a task, such as task goal, due date and importance level are shown as well as the task details, which is a listing of associated information. This listing also displays the content of the associated information objects. For example, calendar items open in their default application in order to change them there. Information objects such as

⁴ The presented browser plug-in works in the same manner for e.g. enterprise intranets and document management systems.

websites, documents and emails are also displayed using the default applications. There are several interaction possibilities for task detail items so that KWs can delete items or see more details.

To better integrate the application plug-ins and the TM sidebar, the sidebar can be controlled from the application plug-ins. The sidebar complements the functionality of the above described application plug-ins. It allows them to remotely trigger the same actions that a user could invoke on the screen. For example, the “add task” dialogue of the sidebar helps different plug-ins to present this dialogue. This takes place in a unified way across their applications.

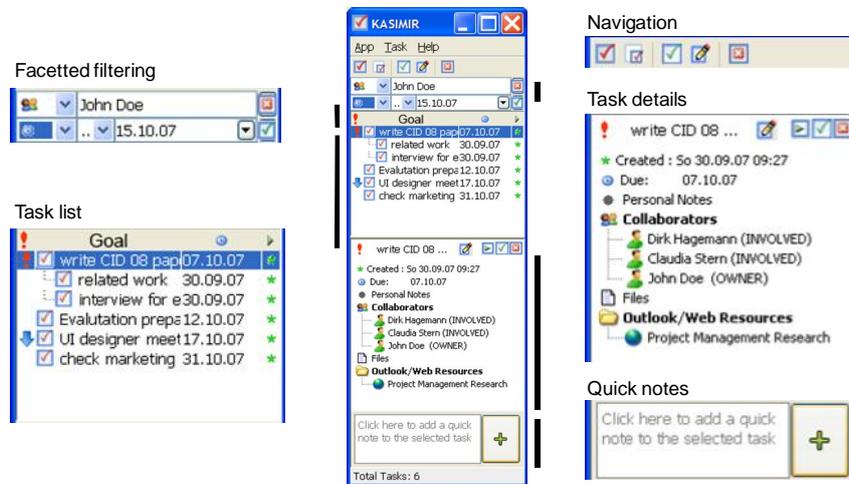


Figure 4: Task side bar: Task list, task details and filtering, notes & navigation.

4 Creating semantic annotations within the daily work process

The KASIMIR prototype is part of the Nepomuk Semantic Desktop. Doing so, KASIMIR leverages the Nepomuk semantic infrastructure. Section 4.1 describes the conceptual framework for how tasks and desktop information objects are semantically represented. Most importantly, section 4.2 outlines how KASIMIR contributes to the framework. KASIMIR creates semantic annotations by seamlessly deriving TM-relevant metadata from within the TM activities of the knowledge worker. Finally, we illustrate in section 4.3 how this is realized within the Nepomuk Semantic Desktop.

4.1 Task and related concepts in personal information ontologies

The Nepomuk Semantic Desktop project establishes a semantic network throughout the desktop. Core of this semantic network, which is useable across desktop applications, is a personal information model that represents the personal knowledge of the KW. It represents concepts of the structuring IOs like e.g. tasks, persons and projects and their relations among them. As well, it links these concepts with the document-like IOs on the desktop. Document-like IOs, being stored physically on the desktop, like for example email messages, files and calendar appointments, are referenced in the personal information model by using their respective metadata concept.

Within Nepomuk, this *personal information model* is represented formally by the Personal Information Model Ontology (PIMO) [Sa06]. It defines a personal knowledge ontology and links document-like IOs to the concepts of structuring IOs. Concepts of structuring IOs can be related to concepts of document-like IOs by ‘grounding’ them with defining a *hasOccurrence* relation to the document-like object [Sa06]. For example, a concept person ‘Claudia Stern’ can be grounded by a Microsoft Outlook contact or a concept topic ‘knowledge management’ by a website URL. As well, the *concepts* of the personal information model can be related formally within the semantic network to represent their relationship. For example, a task can be related to a person because the person works as a collaborator on the task.

The task concept and its relations to IOs on the desktop as expressed in the PIMO are especially relevant for the support of TM. The Nepomuk task model [Ne06a] extends the PIMO task concept [Sa06] and defines particular relations of a task to related concepts of structuring objects and respective properties in a TM model. The task model ontology (TMO) [Ne06a] is the formalized representation thereof. It serves as the ontology for tasks in the presented TM prototype and uses the PIMO for the other related concepts of structuring IOs and document-like IOs.

4.2 Creating semantic annotations using KASIMIR

The presented KASIMIR prototype is fully semantically enabled to support the creation of attributes and relations from the KW’s TM activities. However, it does not present itself to the end-user like traditional semantic applications. Instead, KASIMIR is integrated into TM activities via plug-ins and the sidebar thus enabling KW to relate emails, appointments and websites and their respective concepts of structuring IOs to tasks, all without needing to explicitly manage the underlying semantics. This activity results in semantic annotations according to TMO and PIMO. Furthermore, since the resulting metadata is derived from explicit KW actions, they have a high degree of relevance to the KW. It supports three basic TM activities that result from the work process. In the following we demonstrate the principle of creating semantic annotations for each of them.

Creating a task from an email in Microsoft Outlook is the first TM activity, as shown in Figure 5. It can arise e.g. within the work process, when the KW reacts on an incoming email, reads an email or browses through the email inbox. Creating a task out of the

email, the KW views a pre-populated “add task” dialog and decides about the actual email properties that are added to the task. This leads to the following semantic relations:

- The email represents a document-like information object and is related to the task as attachment.
- The email body defines a topic that describes the task, e.g. using a task tag.
- Involved persons in the email (obtained from the To, CC, BCC properties) represent structuring information objects and are recommended to the KW as potential task collaborators. When approved, these persons are related to the task. When a person does not already exist, the corresponding concept is created in the PIMO and added as collaborator to the task.

Second, the creation of an appointment in Microsoft Outlook works accordingly, except that the related resource is an appointment instead of an email and that the meeting location can be related to the KASIMIR task.

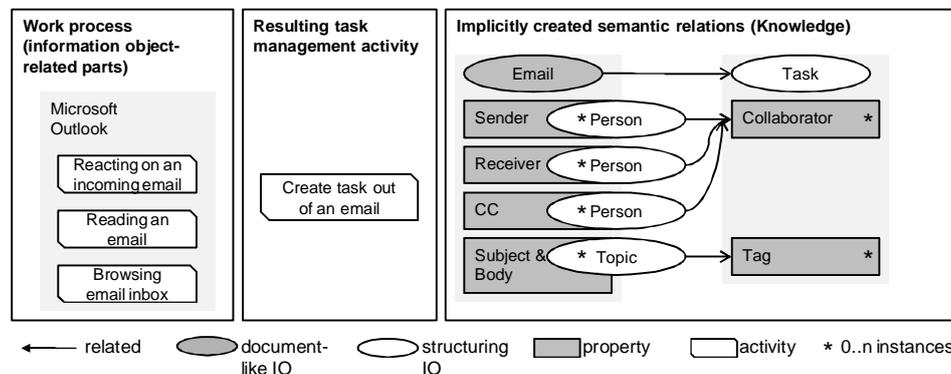


Figure 5: From work process to semantic relations: creating a task from an email.

The third task activity completes the KASIMIR functionality by creating a task from a website in Mozilla Firefox. This creates the following semantic relations:

- Use the URL of the Website and add it as website concept to the task.
- Select website content and assign it as topic concept to the task.
- Select web-site content and assign it as location concept to the task to represent a task’s location.

The Nepomuk semantic middleware supports the presented creation of metadata. In the next section we explain how this is done.

4.3 Nepomuk as the semantic infrastructure for KASIMIR

The KASIMIR prototype (including task sidebar and plug-ins for Microsoft Outlook and Mozilla Firefox) builds upon the Nepomuk Semantic Middleware [Ne06b]. This way we ensure an efficient management of resources and personal information model together with their formal representations - see Figure 6 for a KASIMIR architecture overview.

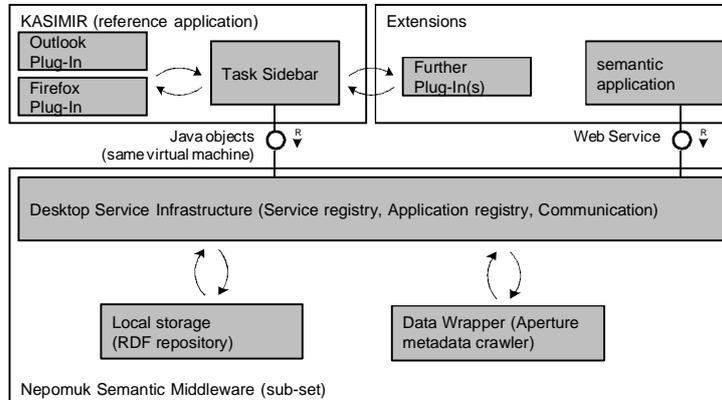


Figure 6: KASIMIR Architecture: Nepomuk Semantic Middleware, KASIMIR & extensions.

For the KASIMIR prototype, the following components of the Nepomuk Semantic Middleware have been used realize the rooting in semantic web technologies:

- *Local Storage* – The RDF repository based on Sesame2⁵ provides the semantic database for all semantic annotations ranging from tasks as well as structuring and documents-like IOs to the Nepomuk ontologies such as PIMO and TMO.
- *Data Wrapper* – The Aperture data wrapper [Ap05] crawls the desktop for document-like information objects such as emails, documents and spreadsheets, and adds their metadata representation to the RDF repository.
- *Desktop Service Infrastructure* – It acts as service and application registry for semantically enabled Nepomuk services and enables communication between them. In particular, it enables the communication between the Task Sidebar and the Local Storage.

Furthermore, the KASIMIR prototype has been designed for *extensibility*. On the one hand, *further plug-ins* can be developed to cover additional TM scenarios where tasks can be integrated into further desktop applications. For example this enables creating tasks from files within the file manager “Windows Explorer”. On the other hand, other *semantic applications* can use the created semantic annotations to perform their domain-specific operations on it by accessing the RDF repository through the Desktop Service Infrastructure.

5 Discussion

We have presented the KASIMIR prototype that integrates TM activities into the desktop applications of knowledge workers. Thus it allows for the implicit creation of semantic annotations that help building a personal semantic network. In particular this provides a consolidated overview of tasks across application boundaries. Due to its application integration KASIMIR demands only minimal effort from KWs to maintain

⁵ <http://www.openrdf.org/>

tasks. Due to its rooting in semantic web technologies, it considerably reduces the major bottleneck of existing semantic applications in an enterprise environment, namely the costs of metadata creation.

On the other hand, the integration of task management into the personal semantic network significantly increases its value for the KW. The task serves as a central information hub that brings all information objects into the work context. In this way it enriches the task information and makes the task a central point of access to work related concepts and resources and thus increases the value of tasks.

If we look for similar user-centric approaches we see the closest relations to other research in the UAM approach [MCF05]. However, compared to this, Nepomuk provides a cutting-edge social semantic technology basis, which is an essential basis for the efficient use of object relation on which the task management particularly depends. Looking into another direction, we also find a parallel to the approach of Adali et al. [APS06], who also extract semantic annotations from task data. However, they do not extract them directly from the operating TM system but by mining. This introduces an additional mediation process that reduces the metadata quality.

The design goal for the presented prototype was to provide a work-process embedded application that supports the KW in performing task management. Consequently, the presented user interface is task-centered and each prototype function has a defined purpose allowing the KW to perform task management. It intentionally doesn't look like a semantic application that e.g. enables generic browsing of semantic information.

The final evaluation of the presented KASIMIR prototype is planned and will be reported on in a forthcoming publication. However, the prototype design was driven by early user-involvement, e.g. by conducting interviews and deriving scenarios. The current prototype has already undergone a preliminary evaluation based on paper prototypes. Eight participants of an evaluation workshop were questioned to solve TM problems with the KASIMIR prototype sketched on paper. These first usability tests lead to early design improvements and have been implemented. Overall, the participants liked the plug-in approach relieving them from getting acquainted to a completely new and isolated TM application.

Being based on the Nepomuk Semantic Middleware, the KASIMIR prototype stands to further benefit from enhancements of this middleware platform. Especially social components, like distributed storage in a Peer-2-Peer network and a community miner for conducting social network analysis, will further enhance TM functionality as well as the efficient creation of semantic annotations.

Further plug-ins will bring tasks even closer to the KW. For example, a plug-in for a file manager, e.g. Windows Explorer enables KWs to relate files to a task. In an enterprise oriented scenario, the integration of plug-ins into enterprise portals and content management systems will definitely create added value.

Introducing further domain ontologies, e.g. about employees and organizational structure, to the existing Nepomuk ontologies will leverage existing enterprise knowledge.

Finally we plan the introduction of task patterns [Ri05] to evolve the KASIMIR prototype from the personal information management level to the organizational knowledge management level.

References

- [Ap05] Aperture. <http://aperture.sourceforge.net>. Accessed 1-9-2007.
- [APS06] Adali, S.; Pearce, S.; Sapino, M-L.: On Querying and Exploring Activities on a User's Desktop. 22nd International Conference on Data Engineering Workshops (ICDEW'06) 2006, p. x103.
- [Be02] Bellotti, V.; Ducheneaut, N.; Howard, M.; Smith, I.: Taskmaster: recasting email as task management. Workshop: "Redesigning Email for the 21st Century", CSCW. 2002.
- [Be04] Bellotti, V.; Dalal, B.; Good, N.; Flynn, P.; Bobrow, D.G.; Ducheneaut, N.: What a to-do: Studies of task management towards the design of a personal task list manager. In Proceedings of CHI 2004. ACM Press, 2004.
- [Bo04] Boardman, R.: Improving Tool Support for Personal Information Management. London, 2004.
- [CHW04] Czerwinski, M.; Horvitz, E.; Wilhite, S.: A diary study of task switching and interruptions. Proceedings of the SIGCHI conference on Human factors in computing systems, pp.175–182. Vienna, Austria. ACM Press, 2004.
- [Cr05] Cress, U.; Barquero, B.; Buder, J.; Hesse, FW: Social dilemmas in knowledge communication via shared databases. In Barriers and Biases in Computer-Mediated Knowledge Communication – and How they may be Overcome (Bromme, R, Hesse, FW. and Spada, H, Eds), pp 143–167, Springer, New York, 2005.
- [Da05] Davenport, T.H.: Thinking for a Living. Harvard Business School Press, Boston, MA, 2005.
- [Ha05] Handschuh, S.: Creating Ontology-based Metadata by Annotation for the Semantic Web. 2005.
- [Ki00] Kirsh, D.: A few thoughts on cognitive overload. *Intellectica*, 30, 19-51, 2000.
- [MCF05] Moran, T.P.; Cozzi, A.; Farrell, S.P.: Unified Activity Management: Supporting people in e-business. *Communications of the ACM*, 48(12), 2005.
- [Ne06a] Nepomuk Deliverable D3.1. Task Management model. 2005: <http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D3-1>. Accessed 1-9-2007.
- [Ne06b] NEPOMUK Deliverable D6.1. First version of the Backbone and Connector Infrastructure. 2006. <http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D6-1>. Accessed 1-9-2007.
- [Py05] Pyöriä, P.: The concept of knowledge work revisited. *Journal of Knowledge Management* 9(3), 116-127, 2005.
- [Ri05] Riss, U. V.; Rickayzen, A.; Maus, H.; van der Aalst, W.P.M.: Challenges for Business Process and Task Management. *J.UKM* 0(2) (2005).
- [Ri06] Riss, U. V.: Knowledge, Action, and Context: Impact on Knowledge Management. In: K.-D. Althoff, A. Dengel, R. Bergmann, M. Nick, T. Roth-Berghofer (Eds.): *Professional Knowledge Management, Lecture Notes in Artificial Intelligence*, vol. 3782, 598-608, 2006.
- [Sa06] Sauermaun, L.: PIMO-a PIM Ontology for the Semantic Desktop (draft). Technical Report, 2006. <http://www.dfki.uni-kl.de/~sauermaun/2006/01-pimo-report/pimOntologyLanguageReport.html>. Accessed 1-9-2007.