Proceedings of I-KNOW '08 and I-MEDIA '08
Graz, Austria, September 3-5, 2008

Semantic Task Management Framework

Ernie Ong[1], Uwe V. Riss[2], Olaf Grebner[2], Ying Du[1]
([1]SAP Research, Belfast, U.K.
[2]SAP Research, Karlsruhe, Germany
ernie.ong@sap.com, uwe.riss@sap.com, olaf.grebner@sap.com, ying.du@sap.com)

Abstract: Despite the growing importance of knowledge work in today's organizations, its support by means of ICT tools is still rather limited. Recent trends in semantic technologies provide novel approaches for an effective solution to these challenges in terms of semantic-based task management. However, task management involves the complex interplay of information and work activities. Thus a semantic task management framework is needed which supports an adaptable semantic foundation, to meet the challenges of knowledge work, via a set of task services on the desktop. To this end, we propose the Nepomuk Semantic Task Management Framework (STMF) as platform for a task-oriented ecosystem for desktop applications.

Keywords: knowledge work, task management, social semantic desktop
Categories: : H.5.3, K.8.0, M.5

1 Introduction

Knowledge work (KW) plays a decisive role in the success of knowledge intensive enterprises and beyond. Consequently, the need for effective support in KW grows increasingly urgent. However, KW is quite a recalcitrant domain with respect to ICT support since it is characterised by highly variable activities of highly skilled knowledge workers (KWers) operating both autonomously and collaboratively [DeFillippi, 06]. There are two core aspects in KW: (1) supporting the management of **information artifacts**, and (2) supporting the coordination of **work activities** – or *task management* (TM) in short. Both are closely entwined and require a joint handling to provide thorough benefit to KWers [Riss, 05b].

Recent emerging trends in semantic technologies provide better support for KWers' Personal Information Management on the desktop [Sauermann, 05]. This also affects task management with respect to the complex interplay of information and work activities. To fully integrate and support TM the semantic framework has to provide a set of task services which can be leveraged from within existing desktop applications to meet the KWers' demands. Therefore the Nepomuk Semantic Task Management Framework (STMF) is built on the Social Semantic Desktop (SSD) that provides such a foundation and meets the challenges of KW, which are described in the following issues:

Modelling: provision of uniform and flexible semantic models of information artifacts and work activities in different social layers (personal vs. organizational) and in different modelling layers (application vs. domain). Here, the STMF exploits the extensible model provided by the Task Model Ontology (TMO) [Nepomuk, 06].

Knowledge: capture and reuse of explicit and implicit knowledge to support knowledge work. To this end, STMF provides the infrastructure to handle information and process-oriented knowledge within common productivity applications. Seamless annotation of semantic metadata in existing work processes and tools is crucial.

Infrastructure: support a task-oriented ecosystem for all desktop applications in a networked environment. This originates from our perspective of tasks as a generic concept that is pervasive across applications and user activities on the desktop, and represents a conceptual hub for organizing information and work activities. STMF additionally narrows the gap between semantic technologies and conventional development technologies to foster widespread adoption.

Architecturally, the STMF is designed as a task management component on top of the fundamental semantic layer, the Nepomuk middleware [Groza, 07]. The STMF provides an interface to desktop applications which require a uniform task model and specific task services. In this respect, the STMF does not see task management as an application on the desktop among others but as a fundamental layer for applications, which deal with tasks and thus require task services, to coordinates all task related activities across all desktop applications.

In Section 2 we first describe the Nepomuk approach of the Social Semantic Desktop as the basis for our approach before we come to the description of the STMF in Section 3. Section 4 provides a short glance a related work. In Section 5 we conclude the paper with a discussion of the results.

2 General Approach

In this section we give some motivation for introducing the STMF. We see task management as a bundle of desktop-wide services that are available for *all* desktop applications to support *all* user activities on the desktop. As such they are bundled in an application embracing framework that application developers can apply to integrate task management in a manner that is consistent with the character of their applications.

For the application developer the STMF brings the advantage that they can deal with a stable interface independent of changes to the ontologies in underlying semantic foundation layer and accessible in a uniform way. The set of services that are offered by the STMF is specific for the task management functionality, i.e., application developers are not required to work with the semantic infrastructure consisting of several ontologies but they can access tasks directly as SOAP web services.

For the KWer the integrating framework allows a uniform access to TM functionality all over the desktop, even if this is not a mandatory consequence. Since the STMF only provides an API and acts as a façade to underlying semantic and context services, task user interfaces may be adapted in a contextual way so that the particular needs of KWers in their specific work situation are optimally addressed.

Semantic integration yields clear advantages for TM since tasks are no longer isolated entities that might possess some task-specific metadata and attachments but provide a comprehensive picture of the KWers' work sphere, i.e., they are part of their personal semantic network on the desktop. Often tasks are the natural access point for KWers to find required information, e.g., we can often remember in which context we have last dealt with a specific document but not where it is stored.

Finally, the STMF supports the social aspects of metadata and task management such as the combined transfer of both via email to delegate work including the respective context of task related information objects and metadata. This offers opportunities to implicitly enlarge the delegate's personal semantic network.

The basis for the STMF is the Social Semantic Desktop (SSD) [Groza, 07]. It essentially incorporates the Semantic Web paradigm to conceptualize the KWers' desktop data and their personal mental models. It makes use of a set of ontologies and a set of Web Services that ensure standardized interfaces and capabilities. The NEPOMUK project provides a standardized SSD architecture that is independent of specific operating systems or programming languages.

2 Semantic Task Management Framework

The design and implementation of Nepomuk STMF and its underlying task model called Task Model Ontology (TMO) rely on the Nepomuk semantic foundation layer, i.e., the set of services and ontologies provided by it. The TMO addresses the need for a semantic model for TM comprising a description of information artifacts and work activities. The STMF, on the other hand, provides uniform and pervasive access to task data and services across applications and user activities on the desktop. Beneath the central role of the STMF for the Nepomuk task management, applications on the SSD can access TM data using direct access to the semantic task description in the RDF format allowing the developer to leverage the full expressiveness of the format.

2.1 Task Model Ontology (TMO)

The central challenge of task management is providing effective task-related information support to knowledge workers. To this end, the underlying task representation, the TMO, must be highly expressive and yet extensible to cater for ill-defined and continuously changing knowledge-intensive work situations.

Consequently, the TMO is structured in two layers: (1) A set of classes and resources which describe task-oriented information and work activities, and (2) an underlying set of Nepomuk classes which support the elaboration or concretization of the more generic concepts e.g. the Personal Information Management Ontology (PIMO) [Sauermann, 06]. To increase the flexibility and extensibility of the TMO, we have introduced a distinction between intrinsic and extrinsic TMO properties. Intrinsic properties are those which are fundamental to task modelling whereas extrinsic or incidental properties enrich the core task descriptors. In this way users are not only free to construct and evolve a highly personalised information model but they can further augment the TMO with new concepts.

2.2 STMF Services

One of the main impediments to the widespread adoption of task management is the lack of support for tasks and their respective artifacts as first-class citizens on the desktop. This leads to a dichotomy that separates the management of tasks from the work contexts from which they arise. This artificial separation does not only increase the cognitive load of end-users who need to regularly switch between applications, it also precludes any possibility of defining semantic relations between information artifacts as described above.

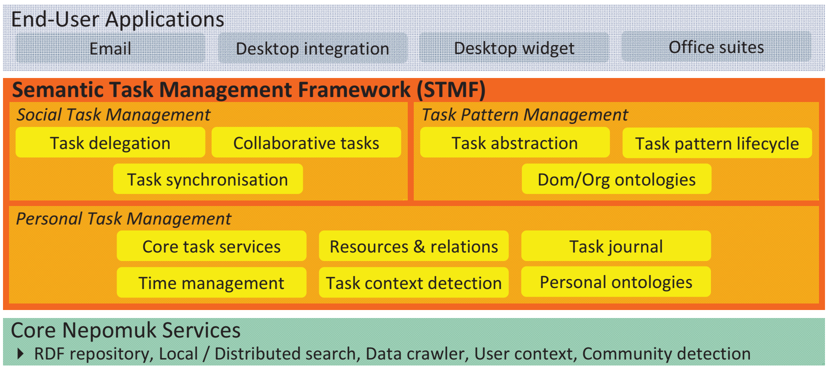

Figure 1: Overview of STMF Components

Effective support for task management on the desktop is predicated on pervasive access to tasks and task services across the desktop, e.g., from different applications, with a consistent data model. While the data model has been described in the last section we now turn to the description of the task services provided by the STMF.

The STMF consists of three categories of task-related services: Personal Task Management, Social Task Management, and Task Pattern Management. **Personal Task Management** (PTM) is the basic layer in the STMF (cf. Fig. 1). It includes *core task services* to create, maintain and delete tasks. Besides, there are other services to enrich tasks. This enrichment includes *resource and relation* handling, e.g., to relate documents or persons to tasks, and *task context detection*, e.g., to relate currently open documents to tasks. Semantic management of task resources is explicitly supported within the TMO but can be further extended via *personal ontologies* and tagging. Moreover, it offers a *task history* to reflect task chronology and *time management* to support users in efficiently organizing task regarding time constraints.

In knowledge work, it often arises that a single person is unable to adequately complete a task due to complexity which requires a multitude of expertise. Such collaboration is supported by **Social Task Management**. In this respect the STMF enables different kinds of cooperation between KWers, e.g., *task delegation*. The respective services enable the exchange of task data including associated information objects. This also encompasses the exchange of metadata that belongs to these information objects including attributes and semantic relations. Exchange of task data therefore goes beyond the conventional use of email and unstructured text as the basis for task descriptions towards a more structured approach via the use of semantics. These services are encapsulated within the STMF and do not impact the user. Beside task delegation there are services to support *collaborative tasks* in which KWers more closely work together, e.g., sharing a common task information space. It supports delegation protocols to control the processes of task delegation and metadata transfer in order to realize *task synchronization*, e.g., for mutual updates of task status information between delegating and delegated tasks.

Finally, we plan to provide services for **Task Pattern Management** that are designed to support experience reuse via task patterns [Riss, 05a]. These task patterns will help KWers to perform new tasks on the basis of recommendations extracted from similar tasks but adapted to new application contexts. The required similarity measures will be based on *task ontologies* that refer to the KWers' business domain. Services will support the categorization of individual tasks via the similarity measures and the extraction of task patterns as *abstractions* of groups of similar tasks, e.g., on the basis of the activities performed within similar tasks as recorded by the *task journal*. Finally, we will provide services that enable the knowledge transfer from personal to collective where the knowledge reuse and organizational learning is possible [Ong, 07]. This will establish a *task pattern lifecycle*.

2.3 STMF Architecture

The overall architecture of the STMF and its environment is depicted in Figure 2.

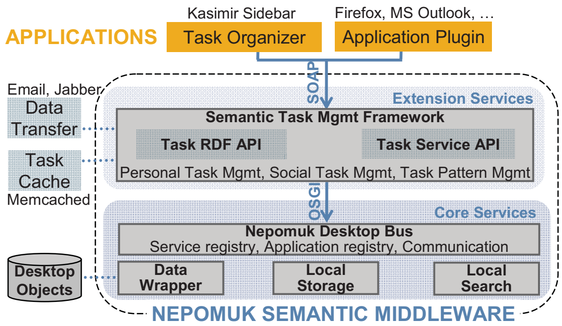

Figure 2: STMF Architecture within the Nepomuk Semantic Middleware

It shows the Nepomuk Middleware that is organized into *Core* and *Extension Services*. Core Services provide the foundational functionality on which the Extension Services are built. All internal communication within the Middleware is based on Java-OSGI whereas applications outside the Middleware rely on platform and language agnostic technologies based on HTTP such as SOAP web services when interacting with a Nepomuk service. The STMF provides Extension Services that use the following Nepomuk Core Services:

- *Nepomuk Desktop Bus* – This acts as a registry and communication layer for semantic Nepomuk services.
- *Data Wrapper* – The Aperture *Data Wrapper* crawls the desktop for *Desktop Objects*, e.g., emails and documents, and adds annotations to the *RDF Store*.
- *Local Storage* – The *RDF Store* based on Sesame2 provides the semantic database for all semantic data.
- *Local Search* – This provides access to semantic data in the *RDF Store* via SERQL and SPARQL queries.

Architecturally, the STMF services are platform and language independent. This is realized by the provision of SOAP web services. From an interface perspective, the STMF services are exposed via two API sets. The first and lower-level API (*task RDF API*) focuses on data access and comprises an RDF interface, the aim of which is to expose task data to semantics-aware applications capable of exploiting the semantic data. This interface provides client applications with direct access to the task data in the RDF Store with both SERQL and SPARQL query support.

Since most conventional applications are not capable of processing semantic data but are object-oriented, the STMF provides a transformation layer which converts RDF data to the object paradigm and thus enables the easy integration of task management within such applications. To this end the STMF also provides a second and higher-level API (*task service API*) that enables both data access and the task management services described above. Internally, the task service API uses the task RDF API to manage the underlying data.

The STMF defines an internal *Data Transfer* layer to manage the transmission of task messages between Nepomuk desktops, e.g., for task delegation and synchronization. The actual implementation can be realized in various ways, e.g., via email or other transport mechanisms such as the XMPP protocol. In the current implementation, the STMF realises the Data Transfer layer by providing interfaces to STMP/POP (email protocol) and to Microsoft Outlook via COM technology. The latter provides full access to the Outlook application model which can be further exploited to access and manipulate Outlook objects including email, address book entries and calendar entries. This enables additional opportunities for bringing email and desktop personal information management tools closer to task management.

Early tests of the STMF in *Task Organizers* such as the Kasimir Task Sidebar [Grebner, 08] have revealed a need for higher performance read operations. Such situations arise regularly when a user routinely navigates between tasks in the Task Sidebar. To this end, the STMF uses a dedicated *Task Cache* to cache frequently read task data. The current STMF uses a desktop-based **memcached** server,[1] a high performance distributed object cache. In preliminary tests, users have noted visible improvements to responsiveness which has had a significant impact on the usability of the Task Sidebar. From a design perspective, caching is realized in a way that is transparent to the core STMF components thereby clearly separating the concerns of provisioning of task management functionality, data caching and data access.

The use of an external cache server, as opposed to one that is tightly integrated within the STMF, has two indirect benefits. Firstly, this architecture enables the STMF to be deployed separately as a Java framework within Java applications. This results in further performance gains due to the binary-level integration with the STMF. In such a deployment, the STMF instance accesses the Nepomuk Core Services via HTTP for data management only. Secondly, the use of an external cache and a write-through caching policy ensures task data consistency across all STMF instances on the desktop in addition to increasing the cache hit rate, and therefore performance, across STMF instances.

The current STMF is designed as a façade that uses service composition to mediate access to the underlying Nepomuk services in addition to orthogonal

[1] www.danga.com/memcached

services, e.g., caching and data transfer. Extensions to the STMF can be realized by delegating stable functionality to the existing STMF (closed for modification). New functionality, on the other hand, can be intercepted and handled separately (open for extension). This adheres to the Open-Closed Principle.[2] From an architectural perspective, an STMF extension can be realized as an additional Extension Service within the Nepomuk Semantic Middleware or as a separate web service. In this way, multiple monotonic variants of the STMF can co-exist on the same desktop.

3 Related Work

An approach rather close to the STMF is the Unified Activity Management (UAM) project at IBM Research [Moran, 05; Cozzi, 06]. The WAX system that results from this approach provides a Web service framework that applies a semantic representation of activities (or tasks) similar to the present approach. In the same way as the STMF it focuses on collaborative task handling, support of unstructured information and a plug-in approach.

The key difference between the STMF and other approaches such as UAM mainly consists in its embedding in the SSD. Thereby we obtain a seamless integration of tasks and other desktop objects that gives KWers universal access to information inside and outside tasks. A further benefit is that extensions to the standard SSD semantic model with personal, domain or organisational ontologies are also well integrated into the STMF. The SSD therefore provides not only services on which to realize the STMF but also a solid modelling foundation on which to enrich task descriptions. One final deviation between the two approaches is the strong foundation for modelling activities within tasks via task journals in the STMF. This forms the basis for enabling task experience reuse.

4 Conclusions and Future Work

The STMF philosophy regards tasks as a natural hub for integrating information, semantics and work process descriptions. As a natural consequence, the goal of the STMF is to realize a task-oriented operating environment for the desktop that provides KWers with more effective work process and information support. This requires semantic integration of KM and TM within productivity tools KWers use to perform their daily work. To this end, the STMF provides a uniform *task model* across all applications and user activities, and realizes a pervasive set of *task services* thereby elevating tasks and task services to first class citizens across the desktop

In the longer-term, we plan to provide more effective support for experience transfer and reuse via *task patterns*. This includes the effective integration of external ontologies and the introduction of multi-faceted context management within the STMF. The former leverages the potential of Nepomuk to integrate the TMO with personal, domain and organisational ontologies to provide a richer means by which multi-faceted task context can be modelled and captured. This in turn provides a

[2] http://www.objectmentor.com/resources/articles/ocp.pdf

means to leverage maturing activities and services for information, semantics and work processes.

Acknowledgements

This work was supported by the European Union IST fund (Grant FP6-027705, project NEPOMUK).

References

[Cozzi, 06] Cozzi, A., Farrell, S., Lau, T., Smith, B., Drews, C., Lin, J., Stachel, B., &Moran, T. P. 2006, *Activity management as a web service*. IBM Systems J., 45(4), 2006, pp. 695-721.

[DeFillippi , 06] DeFillippi, R.J., Arthur, M. B., Lindsay, V. J. 2006, *Knowledge at Work*, Oxford, Blackwell, 2006.

[Grebner, 08] Grebner, O., Ong, E., Riss, U. V. 2008, *KASIMIR – Work process embedded task management leveraging the Semantic Desktop*. In: Martin Bichler et al., ed., Proceedings of 'Multikonferenz Wirtschaftsinformatik' MKWI 2008, Berlin, pp. 715-726.

[Groza, 07] Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R. 2007, *The NEPOMUK Project - On the way to the Social Semantic Desktop*. Proceedings of I-Semantics' 07, pp 201-211.

[Moran, 05] Moran, T.P., Cozzi, A., Farrell, S.P. 2005, *Unified Activity Management: Supporting people in e-business*. Communications of the ACM, 48(12).

[Nepomuk, 06] NEPOMUK Deliverable *D3.1 Task Management Model*. http://nepomuk.semanticdesktop.org/xwiki/bin/download/Main1/D3%2D1/D3.1_v10_NEPOMUK_Task_Management_Model.pdf. Accessed 2007-12-01.

[Ong, 07] Ong, E., Grebner, O., Riss, U.V. 2007, *Pattern-Based Task Management: Pattern Lifecycle and Knowledge Management*. In Proceedings of WM2007. Mar. 2007.

[Riss, 05a] Riss, U. V., Rickayzen, A., Maus, H., van der Aalst, W. M. P. 2005, *Challenges for business process and task management*. Journal of Universal Knowledge Management. J.UKM 0(2) (2005), pp. 77-100.

[Riss, 05b] Riss, U. V. 2005, *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, LNAI, vol. 3782, 598-608.

[Riss, 07] Riss, U. V., Cress, U., Kimmerle, J., Martin, S. 2007, *Knowledge transfer by sharing task templates: two approaches and their psychological requirements*. KMRP 5, 287-296.

[Sauermann, 05] Sauermann, L., Bernardi, A., Dengel, A. 2005, *Overview and outlook on the semantic desktop*. In Decker, S., Park, J., Quan, D., and Sauermann, L., editors, Proceedings of the First Semantic Desktop Workshop at the ISWC Conference 2005.

[Sauermann, 06] Sauermann, L. 2006, *PIMO-a PIM Ontology for the Semantic Desktop* (draft). Technical Report, 2006. http://www.dfki.uni-kl.de/~sauermann/2006/01-pimoreport/pimOntologyLanguageReport.html. Accessed 2007-09-01.