Semantic Task Management Framework: Bridging Information and Work

Ernie Ong, Uwe V. Riss, Olaf Grebner, and Ying Du

Abstract. Despite the growing importance of knowledge work in todays 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.

1 Introduction

In a world of rapid change, knowledge work (KW) plays a decisive role of growing importance in the success of knowledge intensive enterprises. The reality of

Ernie Ong
SAP Research, TEIC Building, University of Ulster, Shore Road; Newtownabbey, BT37 0QB, UK. Current address: The British Library, 96 Euston Road, London NW1 2DB, UK
e-mail: ernest.ong@bl.uk

Uwe V. Riss
SAP Research, Vincenz-Priessnitz-Str. 1, Germany
e-mail: uwe.riss@sap.com

Olaf Grebner
SAP Research, Vincenz-Priessnitz-Str. 1, Germany
e-mail: olaf.grebner@sap.com

Ying Du
SAP Research, TEIC Building, University of Ulster, Shore Road; Newtownabbey, BT37 0QB, UK
e-mail: ying.du@sap.com

S. Schaffert et al. (Eds.): Networked Knowledge - Networked Media, SCI 221, pp. 25–43.
springerlink.com

globalization of networked enterprises and economies places additional emphasis on this frontier. 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 [7]. This condition brings about two core aspects (1) supporting the management of **knowledge artifacts**, and (2) supporting the coordination of **work activities** or task management (TM) in short.

So far the support for KW by ICT tools is still rather limited. The most frequently applied tool in this respect is email although it shows a large number of drawbacks [25]. For example, it lacks appropriate support for information delivery and tracking possibilities, as well as for work organization. These observations suggest an apparent potential for efficient collaborative task management.

In the past several attempts have been undertaken to provide such support on the basis of process-aware information systems [8]. However, so far these approaches show significant shortcomings in terms of flexibility as required for KW. This results in lacking acceptance among KWers [20, 13]. We can put this down to the fact that workflow-like process structures are too rigid and their integration with information management systems is problematic due to variety of possible work situations. Often these rather resemble search activities than well defined processes.

Recently emerging trends in semantic technologies make new approaches for an effective solution to the challenges possible, to better support KW [22]. However, task management involves the complex interplay of information and work activities [17]. Consequently, support for TM within existing work processes and tools is just as crucial [10]. To this end, an effective task management framework is needed which is based on (and supports) a rich and adaptable semantic foundation, to meet the ill-defined challenges KWers face, via a set of task services which can be leveraged from within existing desktop applications.

This is the motivation for the Nepomuk Semantic Task Management Framework (STMF). To meet the challenges of KW, the successful framework must address the following challenges:

1. **Modelling:** support 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 needs an expressive and extensible model of all KW artifacts from desktop information objects and Internet resources to enterprise directories. This is the aim of the Task Model Ontology (TMO). In particular, the TMO must provide efficient access to task information and activity description. This, of course, is the subject studied by knowledge organization (KO) [24] and suggests that the TMO must support the modelling of optimized access paths to such task information.
2. **Knowledge:** capture and reuse of explicit and implicit knowledge to support knowledge work. To this end, STMF should provide opportunities for managing informational and process-oriented knowledge within common productivity applications. Seamless annotation of semantic metadata in existing work processes and tools is crucial.

3. **Infrastructure:** support a task-oriented ecosystem for all desktop applications in a networked environment. This stems 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 should additionally narrow the gap between semantic technologies and conventional development technologies to foster widespread adoption.

Addressing these challenges the STMF is designed as a task management component on top of the fundamental semantic layer provided by the Nepomuk middleware [12]. The STMF provides an interface to desktop applications which require a better integrated task model and specific task services. Moreover, we do not see task management as an application on the desktop among others but as another fundamental layer for applications that provides task services for desktop applications and coordinates all task related activities across all desktop applications.

In the following we first describe the Nepomuk approach and its integration in the Nepomuk Social Semantic Desktop (SSD) as the basis for our approach before we come to the description of the STMF.

2 General Approach

In this section we will explain the motivation for the introduction of the STMF. In particular we will explain the reasons for a specific task management layer between the desktop applications and the fundamental semantic layer. In short, the rationale is that the task management (TM) and the semantic infrastructure, as it has been developed in Nepomuk, supplement each other in central aspects. They represent complementary views of the knowledge artifacts KWers work with. In the following we describe the synergies that result from such integration. In order to find the synergies, however, we first have to look at the limitation of today's task management systems as well as semantic technologies [10].

To start with the analysis of TM we can refer to a study of Bellotti et al. [2] who have investigated the tools which KWers use to record and organize their to-do items and to track task execution. A central result of this and other studies is that most tasks are contained in emails or compiled on paper or print-outs. Only a minority of users applied dedicated TM systems. Another key finding of the study was that the effort of formally managing tasks is usually too high compared to the benefits that the KWer can expect in return. Therefore even writing tasks down on paper is considered as preferable compared to using TM systems. One reason for this is the missing integration of TM systems with email clients and other applications and the support in relating tasks to desktop knowledge artifacts. The deficiencies have mainly prevented an extensive usage of TM tools so far.

If we look at semantic technologies, the situation is similar. Although it is a widespread opinion that semantic technologies possess a high potential for improving KW, there are still considerable obstacles that prevent widespread adoption of such technologies. For example, Colucci and co-workers [5] have asserted that the computational complexity is often challenging. This even holds for rather simple operations. The interaction with semantic-based systems is largely tedious and users often do not possess the required skills. Finally KWers often do not realize the benefit that they could obtain from the additional effort of annotation since suitable applications that make use of semantic capabilities are still missing.

Another problem occurs in information retrieval. KWers spend a considerable amount of time looking for knowledge artifacts. By this term we mean all digital objects that are suited to increase the knowledge of KWers. However, these knowledge artifacts usually appear in one or more work contexts so that often a unique location of the respective artifact is not possible. On the other hand, KWers can often remember in which work activity they have last dealt with a specific artifact so that the work activity appears as an excellent knowledge hub. This requires a task model as formalization of work activity, which can be defined by a task ontology. This ontology can be seamlessly embedded in the ontologies used to describe knowledge artifacts (cf. section 3.1). The STMF uses this integration to translate task related user activities into metadata [10]. In this way the work processes provide the glue between knowledge artifacts and the work context which helps to interpret these. Here we use the term work context to describe all knowledge artifacts, persons, topics, sub-activities etc. that have been involved in this activity. Semantic technologies such as those developed in Nepomuk provide the basis for this integration. For example, the integration allows KWers to navigate through the entire semantic network starting from a suitable task.

The STMF approach aims at overcoming the deficiencies of both sides by employing the mutual strengths. To leverage the synergies we have to work out a way how semantic technologies can support TM and vice versa. The essential improvements of TM and semantic framework encompass the following issues:

1. **Leveraging Task Management**
    a. Semantic network providing support in handling knowledge artifacts
    b. Establishment a desktop-wide task management layer
    c. Providing a platform for application developers to include TM services
2. **Leveraging Semantic Technologies**
    a. Automatic annotation of knowledge artifacts
    b. Ensuring a consistent usage of ontologies
    c. Social aspects of TM and exchange of metadata

In the following subsections we will further focus on these opportunities of TM and Nepomuk Semantic Web Services and show how they are addressed by the STMF.

2.1 Leverage Semantic Information for Task Management

Semantic technologies can realize the information integration required for TM. The basis of this integration consists in the fact that almost all tasks are related to knowledge artifacts which can be stored at various places, e.g., on the desktop. It is often tedious for the KWer to bring all required knowledge artifacts together even if they have worked with them frequently. The reason is that the places where the objects are stored are often selected according to criteria that are not task related, e.g., if all presentations might be stored in one folder. Therefore it is often difficult for KWers to remember the place where they have stored specific information.

One of the central aspects of the STMF is the enrichment of task data by assigning resources that are used in the task. This assignment provides users with easier access to the data that they need for task execution. This is exactly the information that is transferred to the semantic network by the STMF.

This means that KWers can later see in which tasks a person or a document was involved and this information helps them to better understand the roles of these objects. The KWers can directly navigate to the respective task and might find other KWers and knowledge artifacts involved. They can replace a person or the content of a document in new work activities since the role of such objects in the context of the task is clear.

The Nepomuk system provides a compilation of topics - personal semantic concepts defined by the user - that can be extended by topics resulting form task execution. These are simply added to the existing topics and can be used in the same way and exchanged with task co-participants. In the same way new persons that are added to a task are automatically incorporated in the KWer's contact list.

Moreover, task management provides the Nepomuk system with information with whom and when a KWer collaborated and in which order specific activities took place. It also gives information about used resources. This is information that generic Nepomuk metadata annotations cannot generally provide. In this respect we can make use of the Nepomuk context management which provides low level event information, e.g., when a specific document was opened, but cannot reliably assign these events to tasks. Here additional information from the STMF is required.

TM provides a more activity-oriented view since often the mere contents of a document, for example, does not make clear which purpose it was used for. The TM logs can inform about the utilization of a resource since it provides information about what, when and how the resource was applied in the task. A service that supports such a temporal description is the Task Journal Service [19].

Since the STMF is seamlessly integrated in the Social Semantic Desktop (SSD) it can make use of SSD services that help KWers to find required resources and assign them to tasks. This is supported by the integration of the Task Management Ontology (TMO) in the ontologies that describe a conceptualization of the KWers desktop data and their personal mental models.

2.2 Establish a Desktop-Wide Task Management Layer

According to Boardman [3] we can distinguish production and support activities. While the first describe those activities that directly contribute to the KWers work goals, e.g., development environment and text editors, the former are required to organize work so that it can be performed more efficiently, e.g., employing time or task management. Usually both aspects are clearly separated, i.e., we have applications for support activities and applications for production work. However, this separation requires KWers to switch between support and production applications additionally the other frequently occurring interrupts in their work process. Such interrupts lead to a decrease in the KWers productivity and is one of their motivations to avoid TM applications, since a piece of paper can be generally used in parallel to other desktop activities.

The STMF approach consists in the provision of a desktop-wide TM framework offering a set of task services that enable a tight connection of the STMF to desktop applications. These services allow for the incorporation of TM functionalities in the production applications. Prototypically such services have been implemented in applications such as the Mozilla Firefox browser and the Microsoft Outlook client. The services are called from these applications via specific application plug-ins. These plug-ins enable KWers to directly assign websites or emails to tasks or to create new tasks that are assigned to these objects [10].

Figure 1 shows how these applications plug-ins work together with the STMF. On the right hand side of Figure 1 there is the Kasimir TM sidebar, a TM prototype that has been developed in the Nepomuk project [10]. It mainly provides a to-do list with the existing tasks of a user, showing task-subtask relations. For the selected task various views of varying detail are offered, e.g., a detailed resource view, a context view, a Task Journal view, and a Task Pattern view. Kasimir represents a traditional task management application that is enriched by information from the Nepomuk RDF repository.

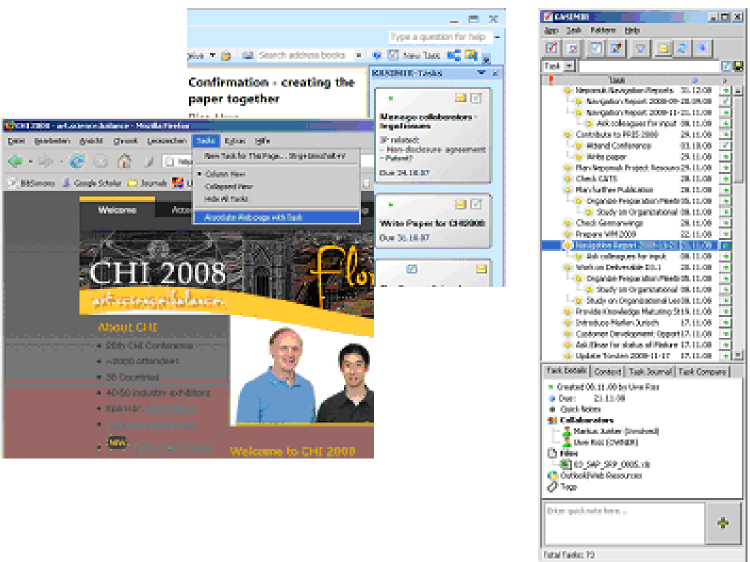

Fig. 1 Application Interactions based on the STMF

On the left hand side of Figure 1 we see the Mozilla Firefox browser and Microsoft Outlook with their TM extensions which are unobtrusively integrated. Thus KWers obtain the opportunity to efficient task handling directly within the context of productive applications. For example, while browsing the internet or intranet or scanning emails the KWers can work with tasks. In this way the KWer can immediately continue the browsing (production activity). The same holds for the email client.

2.3 Platform for Application Developers

A tight integration with desktop applications, however, requires that desktop application developers can efficiently develop plug-ins, which are based on STMF services. One advantage that application developers take of the STMF is the stable interface that it provides. Furthermore, the STMF does so in a manner that is insulated from changes to the underlying ontologies. In particular application developers are not required to directly work with the semantic infrastructure consisting of several ontologies but they can access tasks directly via SOAP web services, for example. Moreover, the STMF performs additional consistency checks that are generic for the TM, e.g., they check that a requested task state transition is valid for the current task state. In this way, the STMF layers task-specific semantics on top of the basic Nepomuk services and, in so doing, the STMF ensures that the task data stored in the RDF Store and the operations on the task data make sense. The services provided by the STMF focus on the TM requirements and disburden the application developer from explicitly dealing with the generic semantic infrastructure.

For the user the integrating framework suggests a uniform access to task management functionality all over the desktop, even if this is not a mandatory consequence. A common framework might even advise application developers to follow common user interaction principles which make it easier for users to deal with TM functions. Since the STMF only provides an API, task user interfaces may be adapted in a contextual way so that the particular needs of a KWer in the specific work environment can be addressed optimally. For example, we might know that in a bibliographic environment a user mainly attaches to be read tasks to found documents so that specific functionality for such tasks can be offered and transformed into metadata. This reduces the definition effort for the KWer.

2.4 Enriching the Semantic Desktop by Task Management

The Nepomuk infrastructure provides a large number of services that help KWers to deal with the semantic data such as desktop crawlers, personal information model support [23], local and distributed search and others. For example, the crawlers support the KWer in including resources in the semantic network. However, such automatic services can only provide rather low-level semantic annotations, e.g., relations of an email to its sender and recipients. Higher level semantic information must directly come from the KWer but it is not required that the KWer manually define them.

It is particularly the TM that can provide such high-level semantic information resulting from the KWers work activities. For example, if a KWer assigns a document or a person to a task by adding it to the respective task context this implicitly means the task and the knowledge artifacts are related in a way that a crawler cannot provide. Transforming such operations into relations means that TM activities enhance the semantic network. This provides the tangible benefit of reducing the effort for users to manually annotate resource. Moreover, the relation between knowledge artifacts, such as documents and tasks, augments the knowledge artifacts with context information. This helps users to better understand the contents of these knowledge artifacts and their work. This is often a problem when using search engines which only provide rather limited access to the context of an information object.

The relationships that results from TM activities such as the assignment of a document or person to a task are immediately reflected in the semantic repository. This means that such information is immediately accessible, e.g., when the KWer browses through this network.

2.5 Ensuring Consistency of Metadata

Ontology engineering is far from being trivial and this particularly holds for KWers. The particularities of handling metadata are often not obvious. For example, the KWer might not be able to clearly decide whether a specific document is related to a task or whether it is a topic of this task. Even if the difference is clearly defined somewhere it might not be obvious to the KWer. This can lead to inconsistent usage of metadata that spoils consistent reasoning and also makes the semantic navigation more complex.

The STMF translates the handling of the ontologies into the handling of tasks. Since thereby a specific application domain is given the meaning of attributes and relations can be determined more precisely. This means that user activities, e.g., assigning a document or person to a task, can be unambiguously related to metadata which are automatically created based on the execution of these activities. The user is not required to directly deal with the ontology and the definition of metadata.

Consequently metadata are defined uniformly and inconsistencies are avoided due to the STMF that interprets all task operations performed by the KWer in the same way. Actually the KWers do not even realize that they are working with a semantic framework. Nevertheless they profit from the benefits that the semantic representation provides.

2.6 Social Aspects of Task Management

Finally, the STMF supports the social aspects of semantic information and task management. In this respect it makes specific use of the email client. For example, the STMF supports the usage of email for task handling and transfer of metadata. Today emails are extensively used for task delegation and tracking but TM functionality that is adapted to this is mainly missing. At the same time the STMF manifests the relation to the email in the TM system and augments the sent task information by metadata. In this way the delegation of tasks implicitly results in an enlargement of the delegates personal semantic network in which the received resources will be semantically included. Of course, privacy issues have to be considered but the resulting opportunities are nevertheless auspicious, in particular, since they support the networking between KWers. To deal with the resulting demands the STMF will incorporate security aspects to support social TM scenarios.

The Nepomuk SSDs are organized in a peer-to-peer network. The communication between these peers is realized by a Network Communication layer that provides a basis for collaborative TM. This particularly supports the collaboration within organizations whereas the external communication uses email as the medium for metadata exchange. In this way richer task information can be exchanged which increases the value of the TM system.

3 Semantic Task Management Framework

The design and implementation of Nepomuk STMF and its underlying task model [11] 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 of TM comprising a description of information artifacts and work activities. In this way the STMF provides uniform and pervasive access to task data and services across applications and user activities on the desktop built on the task unit as a conceptual hub. Despite the central role of STMF for the Nepomuk task management it is to be remarked that access to TM data is not restricted to the STMF. Direct access to the semantic task description in RDF format is also available for applications and an user interface for such direct access has been provided [4].

3.1 Task Model Ontology (TMO)

The central idea of the STMF as described in Section 2 is a seamless integration of TM in the semantic infrastructure. To this end it was necessary to describe the task structure by means of a proper Task Model Ontology (TMO) as part of the existing Nepomuk ontologies and particularly as task specific supplement of the Personal Information Model Ontology (PIMO) [23].

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 more generic concepts in PIMO. The embedding in the Nepomuk ontologies guarantees that task information can be employed throughout the entire SSD.

3.2 STMF Services

The STMF offers TM functionality in the form of a set of services, so-called Task Management Services, that can be used by all desktop applications via respective plug-ins. These services can be grouped by their functionality in terms of the provide functionality. Figure 3 shows the task services and their classification.

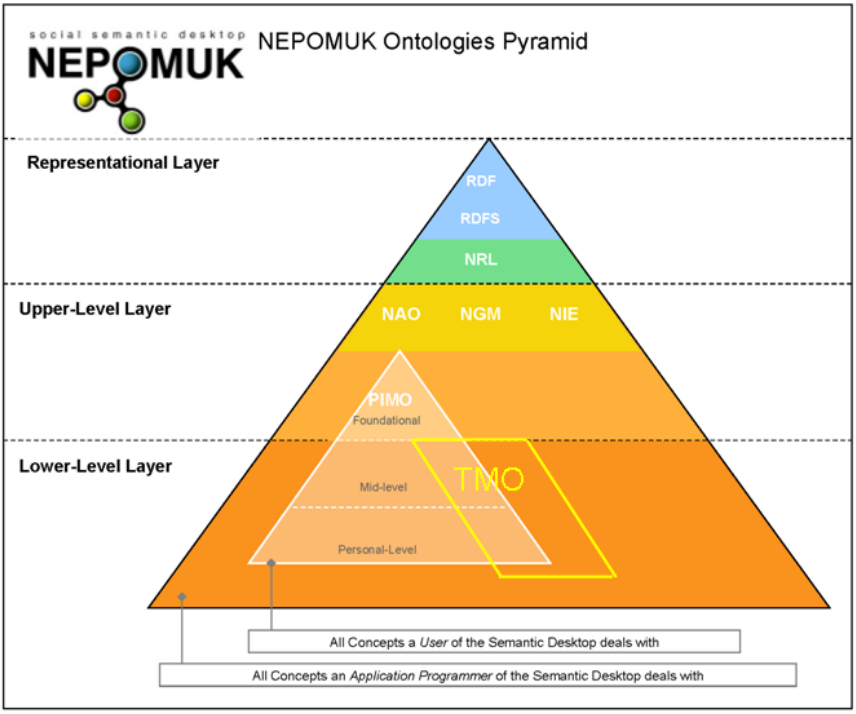

Fig. 2 TMO in the Nepomuk Ontologies Pyramid

The STMF offers services for Core Task Management, Task Experience Management, and Social Task Management as described in Figure 3. The Core Task Management includes Personal Task Management and Task Information Management. The former category includes the main services to organize the personal to-do items while the latter support the task context by enabling the KWer to attach various kinds of information objects to a task. The Social Task Management consists of the categories of (proper) Social Task Management and Social Network Management. The former provides collaborative task services in a wider sense whereas the latter services offers functionality to manage persons and their relations. Finally, the Task Experience Management offers two categories of services, Task Contingency and Task Structure Services. The former category includes services that help KWers to understand individual event that occurs in a task while the latter categories provide services that support the reuse of repetitive task structures. In the following sections, we present these 6 sub-categories in more detail and describe the functionality offered by the services of each of them.

The Personal Task Management offers services for handling tasks and task lists that support KWers in maintaining their task. These services provide methods for **basic task handling** such as task create, read, update and delete. It includes the handling of specific task attributes such as priority, due date, task state. The **task**

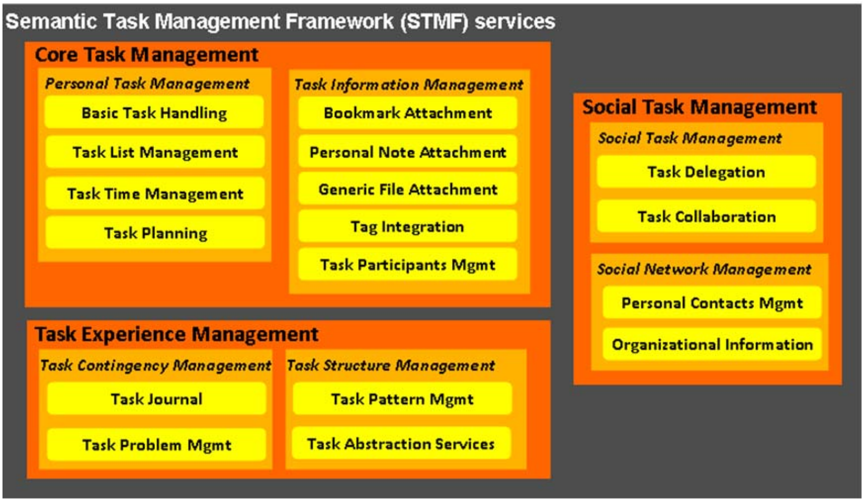

Fig. 3 Overview of STMF Services

list management offers functionality to organize task lists according to different criteria. There are filter and sorting mechanisms. Filter criteria are for example involved persons, involved documents, assigned tags or due dates. It also supports task-subtask relations. The **task time management** enables KWers to monitor and plan the time spent on specific tasks. There are methods to analyze existing sets of tasks as well as for planning future tasks. There are also methods for keeping detailed track records of spent. Similar for planning, task service methods allow for defining and retrieving the target effort. The **task planning** methods allow to (re-)structure tasks. For example, this allows reorganize the task hierarchy.

Task Information Management enables KWers to describe the context of tasks in terms of relevant information. To this end they can attach selected information objects to the task. There information objects include **bookmarks**, **personal notes**, all kind of **desktop files**, **tags**, and **persons** involved in tasks. For the selection of these objects the services can make use of the semantic network provided by the Nepomuk infrastructure. This allows auto-completion and recommendation on the UI level. A particular role plays the **task participants management**. It does not only allow KWers to assign persons to a task but it also enable them to give them specific roles in the task as for example task owner or involved. These roles are also used to indicate to which person a subtask has been delegated or who is the delegatee.

Social Task Management supports the collaboration between KWers in terms of TM. Thus, the STMF enables different kinds of social task interaction as for example **task delegation**. The service also encompasses the exchange of metadata that belongs to these information objects including attributes and relations. Beside task delegation there are services to support **task collaboration** in which KWers share a common task information space. In this respect the distinction of private and public information is supported. Delegation protocols help to control the processes of delegation and metadata transfer in order to realize task synchronization. **Personal contacts management** enables KWers to exchange personal contacts and related information. Finally it is possible to make use of **organizational information** retrieved from organizational repositories.

The Task Contingency Management makes information collected during task execution available. In this way it works with information that is specific for individual tasks. This concerns **task journal** in which the STMF register task events such as the point of time when a particular person was involved in a task or a subtask was delegated. It also includes **task problem handling** of specific situations that occurred during the task execution for later reuse. Aspects of Task Contingency Management have been described in [19].

In contrast the Task Structure Management does not concern singular events but repetitive task structures. This concerns **task patterns** that describe this reoccurring task feature that provide guidance how to perform new tasks on the basis of completed tasks as well as services that recommend information objects to be used in task execution based on previous tasks. Services that help KWers to find suitable information objects on the basis of completed tasks are called **abstraction services**. These services enable the knowledge transfer from personal to collective where the knowledge reuse and organizational learning is possible [16].

3.3 Core STMF Architecture

The core STMF architecture and its environment is depicted in Figure 4. Security related aspects of the STMF architecture will be described in the next section. 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. Conceptually, Extension Services could be used to provide domain- and application-specific support for domain- and application-specific ontologies within the Nepomuk semantic middleware such as that which the STMF provides for the TMO. All internal communication within the Middleware is based on Java-OSGI whereas applications external to 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 service and application registry for semantic Nepomuk services. It also enables the communication between the STMF and the Core Services.
- **Data Wrapper:** The Aperture Data Wrapper crawls the desktop for Desktop Objects such as emails, documents and spreadsheets, and adds their semantic data to the RDF Store.
- **Local Storage:** The RDF Store based on Sesame2 provides the semantic data base for all semantic data ranging from tasks and other concepts like persons to the Nepomuk ontologies such as TMO and PIMO.

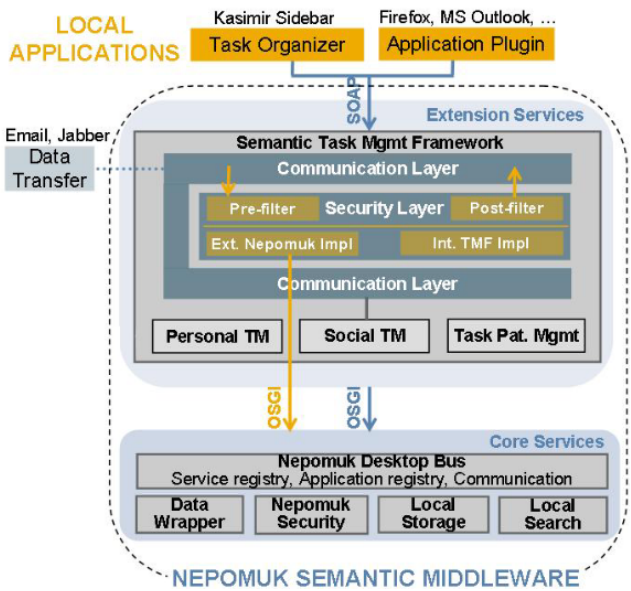

Fig. 4 STMF Architecture within the Nepomuk Semantic Middleware

- **Local Search:** This provides access to semantic data in the RDF Store via SeRQL and SPARQL queries.

From an architectural perspective, 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 that aims at exposing 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 and do not use semantic technologies such as RDF and SPARQL but are based on more conventional object-oriented technologies, the STMF provides a transformation (adapter) layer which converts RDF data to the object paradigm thus enabling 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 provides task management specific services on top of the Nepomuk semantic middleware, thus enabling both data access and the task management services described above. Internally, the task service API uses the task RDF API to realize the data access.

The STMF defines a Communication Layer and accompanying Data Transfer adaptors to manage the transmission of task-related messages between Nepomuk desktops, e.g., for task delegation and synchronization of semantic data and information objects. The actual implementation of the Data Transfer adaptors can be realized in various ways, e.g., via email or other transport mechanisms such as Jabber/XMPP. In the current implementation, the STMF provides adaptors via the standard email protocols (STMP/POP) and Microsoft Outlook using COM technology. In addition to data transfer, the Outlook adaptor also provides full access to the Outlook application model. This can be exploited to access and manipulate Outlook objects including email, address book entries and calendar entries from within the STMF thus enabling bi-directional synchronisation between semantic and Outlook application repositories. The end result is a much closer integration between popular desktop productivity and information management tools and semantic task management.

Communication scenarios, which deal with data and information object exchange between task participants, do however pose one key challenge: security. For example, the initiation of task communication such as task delegation requests presents vulnerabilities where information may be exposed to other users. The Security Layer in the STMF manages the task-specific security requirements in such scenarios.

The STMF implements security at two levels corresponding to the exchange of data and information objects described above. At the metadata level, the STMF limits access to task data in two ways. First, private data is automatically omitted or post-filtered so that external users are neither aware of nor able to access the data which the owner considers private. This, in turn, is supported directly in the TMO which provides privacy attributes for tasks, task attachments and task journal entries. And second, the STMF enables the user to exchange non-private task data with task participants. This ensures that the user is responsible for explicitly selecting the data to be supplied to the addressee. In short, these mechanisms realise the principle of least privilege in regard to task data in communication scenarios.

The Security Layer is also responsible for mediating access to information objects referenced in tasks. Once again, it adheres to the principle of least privilege by pre-filtering and rejecting unauthorised requests for information objects. Examples of these include

1. **Requester-integrity verification:** The requester must be a task participant.
2. **Private exclusion:** Only information objects still marked public are allowed.
3. **Explicit authorisation:** The owner must explicitly authorise access to the requested information object.
4. **Trusted return address:** The owner, and not the requester, is responsible for determining the return address for the information objects.

From a design perspective, the STMF applies aspect-oriented principles to separate security concerns from the rest of the STMF. For example, pre-filters are designed to prevent the completion of a STMF operation, i.e., a document request, in the event a precondition is not satisfied, e.g., the person requesting access to a document is not a task participant. This is realised in a manner that is transparent to core STMF functionality by encapsulating the Security Layer within the Communication Layer. This provides a clear separation of concerns between main STMF operation, communication and security within the STMF.

The Internal (Security) TMF Implementation component addresses only task-specific security requirements. On the other hand, the External Nepomuk (Security) Implementation component cooperates with the Nepomuk Security component to address more generic security aspects which are common across the Nepomuk infrastructure, e.g., role-based access control, encryption and digital signatures. This provides a clear separation between task-specific and generic security aspects.

3.4 Layered Refinements on the TMO and STMF

The STMF is an implementation of a set of domain-specific services to support task management within the Nepomuk semantic framework. To this end, the STMF has been designed with extensibility to its underlying data model and its functionality to support orthogonal domain- and new application-specific concepts and services. Orthogonal concepts can be accommodated by composing the TMO while the new services can be introduced by extending the STMF.

The TMO task data model can be extended with orthogonal organizational, domain and application specific ontologies which describe the modelling needs in specific situations. Such ontologies can be aligned with the TMO via ontology composition which aims to harmonise and align two or more ontologies.

From a technical perspective, since the STMF uses data access objects (DAO) to mediate access to the task data in the RDF Store, extensions to the task data model can be supported by introducing new DAOs to encapsulate access to the extended data model. Due to the availability of different RDF data access frameworks, abstract DAO factories are used to instantiate concrete DAO classes and the respective mappings between the RDF data model and the corresponding Transfer or Value Object (VO) in the object realm.

At present, the STMF uses the RDF Reactor framework in its DAO layer to provide object-oriented Java proxies to the underlying RDF data. Consequently, changes to the underlying RDF framework or the use of a different data management technology can be implemented without any impact to the non-storage classes in the STMF, e.g., by adding new concrete DAO classes. This design aims at providing configurability within the STMF while maintaining stability both within and without, e.g., for applications using the STMF. This is crucial since the STMF goes beyond supporting task data but furthermore mediates access to other data in the RDF Store and other Nepomuk services. It is to be remarked that the use of DAO, DAO factories and VO provide a flexible framework for integrating business objects from enterprise systems.

Whereas extensions to the STMF data model are realized via ontology composition, refinements and extensions to the STMF functionality are realized via service composition. This may be used to provide new domain- and application-specific services based on the STMF. The current STMF is designed as a façade (design pattern) that uses service composition to mediate access to the underlying Nepomuk services. Extensions to the STMF can also be realized in a similar fashion where core STMF functionality is delegated to the existing base STMF. New functionality, on the other hand, can be intercepted and handled separately. 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. Very importantly, this adheres to the Open-Closed Principle of object-oriented software construction.

The STMF together with the TMO therefore provide strong technical and semantic foundation, respectively, on which to build and customise task management services according to the needs of different application and organisational situations.

4 Related Work

One of the core insights of the present paper is the strong relation between Personal Information Management and Task Management. In fact, every task execution requires and produces information (which is primarily on the personal level). In this respect the current approach is similar to the one that is realized in the OntoPIM approach [14]. In a similar way as the Nepomuk project, OntoPIM fosters the idea of a Personal Ontology reflecting the users perspective of their work domain. To obtain the Personal Ontology, OntoPIM extracts information from the information objects the users are working with, e.g., emails, via an inference engine. In this way OntoPIM supports users in performing tasks. For example it proposes new tasks as successors of current tasks on the basis of existing task logs, e.g., suggesting a FindFlight task after a FindHotel task.

The STMF is aiming at the same goal. However, the central idea of STMF is to go beyond automatic inference and derive relevant information directly from the users' work activities, i.e., not from task logs. Moreover, it aims at providing relevant task information based on social experience. Here the STMF focuses on social interaction, i.e., exchange of metadata between KWers, and the idea of abstract task patterns [18].

One of the first TM products that has followed the idea of information integration was Caramba [9], supporting TM for virtual teams by enabling links to information objects, tasks, and resources. However, it does so on a non-semantic basis.

Another related approach is the Haystack system [1] that goes already back to 1999. It is also rooted in a semantic network technologies based on an RDF infrastructure and includes tasks as a central concept as well. In this respect it is more comparable to the Nepomuk infrastructure. Haystack also shares in the insight that task handling determines a significant percentage of the users working time. To support tasks the Haystack system provides a task pane that gives access to task relevant objects. On the other hand, Haystack follows the traditional view of regarding task management as an application among others and not as a service layer that can be accessed by various applications. Thus the only closer integration of tasks is realized for email. Nevertheless, from a general perspective Haystack follows a similar cross-domain approach as Nepomuk.

The approach that is closest to the STMF is the Unified Activity Management (UAM) project at IBM Research [15]. The WAX system that results from this approach provides a Web service framework that applies a semantic representation of activities (or tasks) in a similar way as the present approach [6]. In the same way as the STMF it focuses on collaborative task handling, support of unstructured information and a plug-in approach. Moreover, we share with UAM the belief that the formalization of tasks opens a wide range of opportunities for better support of KW. The main difference between UAM and STMF is the integration in the SSD.

The key difference between the STMF and other approaches as UAM mainly consists in the fact that the STMF is essentially embedded in the SSD and extensively utilizes this integration. In this way the STMF does not only provide information to the SSD but also supports the task management by making information objects from the SSD available to the task management and the knowledge artefacts related to tasks are not only available for TM applications but to all desktop applications that are connected to the STMF. A further benefit here 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.

5 Conclusions and Future Work

Despite its growing importance, the support for task management for knowledge workers by desktop ICT tools is still limited. Thus, task support tools are clearly separated from the tools KWers mainly use in their daily work activities. This results in additional cognitive and administrative overhead. The goal of the STMF is to realize a task-oriented operating environment for the desktop that provides KWers with more effective support in a manner that can be fully integrated with tools they already use. To this end, it addresses two key challenges, namely providing a uniform task model across all applications and user activities, and realizing a pervasive set of task services thereby elevating tasks and task services to first class citizens across the desktop.

However, the STMF initiative is far from complete. In the short-term, we plan to realize a task-oriented messaging bus to support multi-directional events between the STMF and any STMF-aware services and applications. This leads us closer to the ideal of a universal task-oriented operating environment on the desktop by providing a communication layer for supporting complex interactions between desktop applications, events, and enterprise systems from which such events may arise.

In order to incorporate security aspects, the STMF will provide security-based filtering, e.g., of private information, so that external users are neither aware of nor able to access the information that the owner considers as private. The security concept will also include the access to semantic relations between resources, i.e., indirectly related resources.

A central aspect of our further development is to use the STMF to provide more effective support for experience management and reuse via task patterns [19]. Experience management in the field of knowledge work requires a tight integration of process and knowledge management. The STMF provides an ideal platform to bring both aspects together. Moreover, the web service approach of the SSD offers interesting integration opportunities for business process management and the STMF [21]. These advantages can even be increased by the integration of external ontologies and the corresponding metadata 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. This provides a richer means by which multi-faceted task context can be described. The multi-faceted nature of task context is necessary to provide an effective basis for understanding the different aspects of the information and work process needs embedded within tasks. This in turn forms the foundation for task pattern abstraction based on the needs of the KWer. The realization of the STMF within the Nepomuk SSD is therefore highly valuable, not just from a technical perspective but also as a means to gain clearer insights into the needs and preferences of the KWer.

Acknowledgements. This work has been partially funded by the European Commission as part of the Nepomuk IP (grant no. Grant 027705) within the 6th Framework Programme of IST.

References

1. Adar, E., Karger, D.R., Stein, L.A.: Haystack: Per-user information environments. In: CIKM, pp. 413–422. ACM, New York (1999)
2. 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: Dykstra-Erickson, E., Tscheligi, M. (eds.) CHI, pp. 735–742. ACM, New York (2004)
3. Boardman, R.: Improving tool support for personal information management. Ph.D. thesis, Department of Electrical and Electronic Engineering Imperial College, University of London (2004)
4. Brunzel, M., Mueller, R.M.: Handling of task hierarchies on the nepomuk social semantic desktop. In: Sebillo, M., Vitiello, G., Schaefer, G. (eds.) VISUAL 2008. LNCS, vol. 5188, pp. 315–318. Springer, Heidelberg (2008)
5. Colucci, S., Noia, T.D., Sciascio, E.D., Donini, F.M., Ragone, A., Rizzi, R.: A semantic-based fully visual application for matchmaking and query refinement in b2c e-marketplaces. In: Fox, M.S., Spencer, B. (eds.) ICEC. ACM International Conference Proceeding Series, vol. 156, pp. 174–184. ACM, New York (2006)
6. Cozzi, A., Farrell, S., Lau, T., Smith, B.A., Drews, C., Lin, J., Stachel, B., Moran, T.P.: Activity management as a web service. IBM Systems Journal 45(4), 695–712 (2006)
7. DeFillippi, R., Arthur, M.B., Lindsay, V.J.: Knowledge at Work. Blackwell, Oxford (2006)
8. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H.: Process-aware information systems: bridging people and software through process technology. John Wiley & Sons, Inc, New York (2005)
9. Dustdar, S.: Caramba - a process-aware collaboration system supporting ad hoc and collaborative processes in virtual teams. Distributed and Parallel Databases 15(1), 45–66 (2004)
10. Grebner, O., Ong, E., Riss, U.V.: Kasimir - work process embedded task management leveraging the semantic desktop. In: Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P. (eds.) Multikonferenz Wirtschaftsinformatik, GITO-Verlag, Berlin (2008)

11. Grebner, O., Ong, E., Riss, U.V., Brunzel, M., Bernardi, A., Roth-Berghofer, T.: Task management model - Nepomuk deliverable D3.1. Tech. rep. (2006), `http://nepomuk.semanticdesktop.org/xwiki/bin/download/Main1/D%3%2D1/`
12. Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R.: The nepomuk project - on the way to the social semantic desktop. In: Pellegrini, T., Schaffert, S. (eds.) Proceedings of I-Semantics 2007, pp. 201–211. JUCS (2007)
13. Holz, H., Maus, H., Bernardi, A., Rostanin, O.: From lightweight, proactive information delivery to business process-oriented knowledge management. Journal of Universal Knowledge Management 2, 101–127 (2005)
14. Lepouras, G., Dix, A., Katifori, A., Catarci, T., Habegger, B., Poggi, A., Ioannidis, Y.: Ontopim: From personal information management to task information management. In: Personal Information Management, SIGIR 2006 workshop, Seattle, Washington, August 10-11 (2006)
15. Moran, T.P., Cozzi, A., Farrell, S.P.: Unified activity management: Supporting people in e-business. Communications of the ACM 48(12), 67–70 (2005)
16. Ong, E., Grebner, O., Riss, U.V.: Pattern-based task management: Pattern lifecycle and knowledge management - benefits and open issues. In: Gronau, N. (ed.) 4th Conference on Professional Knowledge Management - Experiences and Visions, Potsdam, Germany, March 28 - 30, GITO, Berlin (2007)
17. Riss, U.V.: Knowledge, action, and context: Impact on knowledge management. In: Althoff, K.-D., Dengel, A.R., Bergmann, R., Nick, M., Roth-Berghofer, T.R. (eds.) WM 2005. LNCS (LNAI), vol. 3782, pp. 598–608. Springer, Heidelberg (2005)
18. Riss, U.V., Cress, U., Kimmerle, J., Martin, S.: Knowledge transfer by sharing task templates: two approaches and their psychological requirements. Knowledge Management Research and Practice 5, 287–296 (2007)
19. Riss, U.V., Grebner, O., Du, Y.: Task journals as means to describe temporal task aspects for reuse in task patterns. In: ECKM 2008 Proceedings of the 9th European Conference on Knowledge Management, pp. 721–729 (2008)
20. Riss, U.V., Rickayzen, A., Maus, H., van der Aalst, W.: Challenges for business process and task management. Journal of Universal Knowledge Management 2, 77–100 (2005)
21. Riss, U.V., Weber, I., Grebner, O.: Business process modeling, task management, and the semantic link. In: Hinkelmann, K. (ed.) AAAI Spring Symposium on AI Meets Business Rules and Process Management, Stanford Univ., pp. 99–104. American Association for Artificial Intelligence, Menlo Park (2008)
22. Sauermann, L., Bernardi, A., Dengel, A.: Overview and Outlook on the Semantic Desktop. In: Decker, S., Park, J., Quan, D., Sauermann, L. (eds.) Proceedings of the 1st Workshop on The Semantic Desktop at the ISWC 2005 Conference, CEUR Workshop Proceedings, vol. 175, pp. 1–19. CEUR-WS (2005)
23. Sauermann, L., Elst, L., Dengel, A.: Pimo - a framework for representing personal information models. In: Pellegrini, T., Schaffert, S. (eds.) Proceedings of Iqand I-fInternational Conferenceson New Media Technology and Semantic Systems as part of TRIPLE-I 2007, pp. 270–277. J.UCS, Know-Center, Austria (2007)
24. Sigel, A.: Towards knowledge organization with Topic Maps (2000)
25. Whittaker, S., Sidner, C.L.: Email overload: Exploring personal information management of email. In: CHI, pp. 276–283 (1996)