



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *I-SEMANTICS International Conference on Semantic Systems. Austria. September 3-5, 2008.*

Citation for the original published paper:

Grebner, O., Ebner, H. (2008)

Collaborative Tasks using Metadata Conglomerates - The Social Part of the Semantic Desktop.

In: *Proceedings of I-SEMANTICS '08Graz*

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-50104>

Collaborative Tasks using Metadata Conglomerates – The Social Part of the Semantic Desktop

Olaf Grebner

(SAP Research, Karlsruhe, Germany
olaf.grebner@sap.com)

Hannes Ebner

(Royal Institute of Technology, Stockholm, Sweden
hebner@csc.kth.se)

Abstract: This paper¹ presents an application that enables loose and ad-hoc task collaboration for knowledge work and explicitly integrates task metadata into the collaboration process. With the increasing availability of semantic desktop technology, knowledge workers (KWers) can organize their personal information in a formalized way, including tasks. In an organization, KWers need to collaboratively access task-related information and to collaboratively work on it. In such a scenario, today's available collaboration support applications, e.g. enterprise collaboration systems like wikis, either sacrifice end-user experience or semantic richness when dealing with structured knowledge. The presented collaborative task management (TM) application circumvents this trade-off by showing how the Kasimir TM client and the Collaborilla collaboration server interact closely. The TM client supports the KWer's personal TM and incorporates collaborative tasks. It invokes the collaboration server which centrally manages the collaborative tasks. It implements elaborated methods for metadata sharing and collaborative editing of this shared metadata. We present the detailed proposal for an underlying architecture of the application, review related work and conclude this paper by pointing out future work.

Keywords: Task Management, Ad-hoc Collaboration, Metadata Sharing, Collaborative Metadata Editing, Semantic Desktop

Categories: H.1, H.4, M.3, M.4, M.5, M.6

1 Introduction

People performing knowledge-intensive work, i.e., knowledge workers (KWers) like, e.g., managers and researchers, have a highly dynamic working style [Davenport, 05]. Executing knowledge-intensive work tasks is characterized by a highly dynamic work approach as well as the situation that diverse outcomes satisfy the given goal.

In today's business environment, KWers increasingly work in teams to achieve organizational goals. For this, they work together on defined tasks towards a common, task-related goal, a *collaborative task*. Often, collaborative work takes place under

¹ This work is supported by the European Union (EU) IST fund (Grant FP6-027705, project NEPOMUK, <http://nepomuk.semanticdesktop.org>) and the EU eContentplus programme (project Organic.Edunet, <http://www.organic-edunet.eu>).

time pressure and is *distributed* with respect to time and place. This leads to the need for loose and ad-hoc collaboration and corresponding task management (TM).

Key to a collaborative task is that participating KWers *share a common set of information* to enable joint work and to achieve the task goal. Thereby, the KWers need to *commonly access and collaboratively work with the information*. E.g., this comprises of task information like task status, goal and description as well as attached task resources like e.g. files, contacts and emails. Each KWer can add, remove and modify the commonly available information, e.g., attach new documents, add new participants, change the status of the task or create new subtasks. Roles grant a KWer rights on a collaborative task, e.g., the task owner can change all task information.

A recent trend is the *increasing availability of structured, semantic information* in both personal and enterprise environments. Enterprise applications organize their information in a structured way and *organizational knowledge management* initiatives leverage ontologies to create a flexible information model. With the advent of *semantic desktop technologies*, e.g., on the Nepomuk Social Semantic Desktop (SSD) [Groza, 07], *personal* information is available in a structured form by using e.g. a personal information model ontology (PIMO) [Sauermann, 07]. This enables a KWer to keep and retrieve personal knowledge in a structured way.

However, the *problem* is that today's support applications can't deal with structured knowledge in the collaborative task scenario as described above. There is no widely-adopted enterprise solution enabling KWers to share and collaboratively edit structured task information. Existing solutions have severe drawbacks with regard to metadata handling. E.g. one option is to enable task collaboration by serializing task metadata into a human-readable form and putting it onto a wiki page. However, on the wiki the semantic task structure gets lost, even in a semantic wiki many KWers have a hard time applying correct syntax in the editing process. Even if a wiki page is attached to each structured task, the collaboration there doesn't include the available structured task metadata. Another option consists of serializing a task into a machine-readable form, like e.g. RDF and putting it into a shared RDF store. This preserves task semantics, but versioning problems cannot be resolved, e.g., in case of two concurrent modifications, as no handling protocol is in place. As well, every task participant needs write access to the shared RDF store, a hard prerequisite for ad-hoc tasks and in large organizations with extensive administrative procedures.

Our *proposed solution* is to apply a *wiki principle to structured content* for enabling loose and ad-hoc collaboration for TM. We present the combination of the Kasimir personal TM client application [Grebner, 08] based on the Nepomuk SSD, with the Collaborilla collaboration server (CS) [Ebner, 07 & Collaborilla, 08]. The strength of the proposed approach is that the CS enables collaborative tasks and ad-hoc collaboration while preserving the full semantics of the collaborative task information. To ensure a high usability, the KWer interacts with the CS transparently through a TM client that she uses regularly and she is familiar with, like e.g. Kasimir. The preserved semantics enable other KWers to efficiently re-use the structured task information in their semantic desktop applications, e.g., in their favorite TM client.

The *paper structure* is as follows: First, we present the two collaborative TM application parts in detail. Second, we look at the underlying architecture. Third, we review related work. Finally, we conclude with an outlook on how to generalize the proposed infrastructure for metadata sharing among semantic desktops.

2 Collaborative Task Management with Metadata conglomerates

In this section, we first explain the KWer's task client and then, how the collaboration server (CS) will enable collaborative task work in conjunction with the task client.

2.1 Collaborative Tasks embedded into the Personal TM Client

The Kasimir TM application [Grebner, 08] is the KWer's personal TM system using semantic web technologies of the Nepomuk SSD. It's designed to address the KWer's task overflow [Kirsh, 00] by efficiently supporting information management. E.g., by using application plug-ins it doesn't require KWers to copy and paste information from different application contexts to their TM tools, being a major drawback of personal information management tools [Belotti, 02]. Kasimir organizes a KWer's task lists and tasks. A KWer can assign task metadata to each task, like e.g. due date, task description, and the KWer can prioritize tasks. The KWer can attach as well SSD metadata, i.e., personal concepts of the PIMO like e.g., participating persons, projects and topics. Another key feature is the close integration of tasks with information objects on the desktop. Kasimir allows the KWer to add in the TM application references to emails and files, e.g., task-relevant documents.

Collaborative tasks are integrated in the Kasimir personal TM application and thereby into the KWer's TM process. A KWer manages collaborative tasks like personal tasks within the known personal TM application, as a local version of the collaborative task is checked out from the CS. This version shows the most recent state of the collaborative task with all consolidated contributions and offers additional collaboration options compared to a personal task. The KWer can create a new collaborative task, mark an existing task as collaborative or upon synchronization, a new collaborative task triggered by a task collaborator is put into the task inbox, see below.

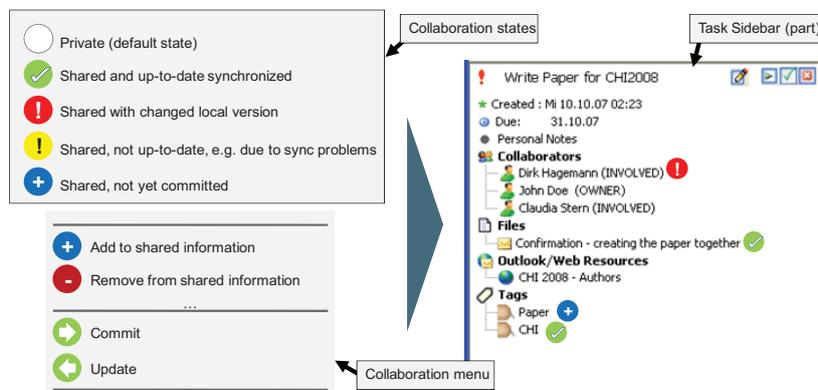


Figure 1: Collaboration status indicators & -menu & Task Sidebar (mock-up).

The KWer can add SSD metadata to the task, e.g., information like notes, emails or documents, and can prioritize this task in the context of the other personal tasks. By default, all information attached to the local version of the collaborative task is private

and it is only shared on request, i.e. by the KWer's explicit opt-in. A visual indicator shows the collaboration state for each task information object, i.e., whether it is shared and if yes along with the state of the shared information, see .

All task information is queued locally until the KWer initiates a synchronization transaction. The KWer initiates a synchronization transaction, i.e., an update of the local version by the current CS version, by selecting "Update" in the collaboration menu. The KWer can contribute to a collaborative task by committing this information to the CS by selecting "Commit", see . This way, a KWer can add or modify task information, i.e., adding a contribution to the collaborative task on the CS.

The KWer can change own contributions and commit them, the KWer's local version is the master. Contributions owned by other task collaborators can be changed as well, however the CS handles this by annotating the changes to the task collaborator's contribution and changes the local version accordingly. This procedure is transparent to the KWer, e.g., when a KWer removes a shared email from a task, the respective information object is annotated as to be deleted from the collaborative task. The KWer's next commit transaction publishes this annotation to the CS.

2.2 Collaboration server hosts collaborative tasks and mediates interaction

The *Collaborilla* CS caters for metadata sharing and collaborative editing of task information and thus addresses mentioned metadata sharing problems. Collaborilla implements elaborated *methods for collaboratively constructing artifacts* and building knowledge conglomerates [Ebner, 06 & Ebner, 07].

It can handle concurrent contributions, i.e., modifications. Following the wiki approach, each KWer can comment on the task metadata as well as on the task contributions of other KWers. A KWer can *modify the task without changing the original task*, including other KWer's contributions to it, as the modification is attached itself as contribution to the task. The client merges all contributions by participating KWers to present the KWer with an up-to-date task version. Thus, the KWer can modify this task conglomerate without write access to the KWer's original task contribution. Furthermore, it enables *ad-hoc collaboration* as there's *no need to grant write access to each participating KWer*. The KWer doesn't need to know in advance who will work with this task, because the contributions can be made by all participating KWers and the client filters the contributions based on the invited participants. For the KWer all these functions are available transparently in the TM client.

3 Architecture

In this section we show information management details of the Kasimir TM client based on the Nepomuk SSD, of the Collaborilla CS and the corresponding interaction between these components. See Figure 2 for an architecture overview.

3.1 Kasimir task management client

The Nepomuk SSD establishes a semantic layer on the desktop. Its core, the personal information model, is formalized using the Personal Information Model Ontology (PIMO) [Sauermann, 07] and represents concepts like e.g. tasks, persons, projects and

possible relations among them. The model instances represent the KWer's personal information. For example, the KWer can model that the person "Claudia Stern" is related to the project "NEPOMUK" using the "works for" relation. As well, this personal information model is grounded with the desktop resources like e.g. emails or files. So-called data wrappers crawl desktop resources to create corresponding metadata items which can be referenced in the semantic layer. E.g., the Aperture data wrapper [Aperture, 05] crawls files which can be related to a task.

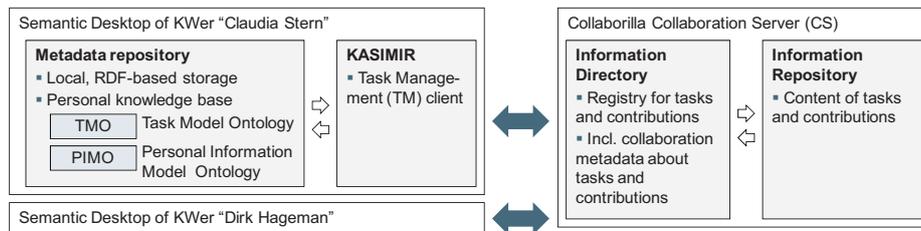


Figure 2: Architecture of Kasimir TM client and Collaborilla CS.

The Kasimir TM client handles both personal and collaborative tasks (local version) using the same task model. The representation of tasks is supported by the Nepomuk task model and the corresponding task model ontology (TMO) [Nepomuk, 06], an extension of the PIMO task concept [Sauer mann, 07]. It serves as the ontology for tasks in the presented TM prototype and uses PIMO concepts to express the relations to other concepts and information objects on the desktop.

3.2 Collaborilla collaboration server

The *Collaborilla CS* has two components. A central registry acts as *information directory* and keeps track of artifacts, i.e., a collaborative task, and its decentralized contributions by KWers. It provides a list of metadata elements for each task and a complementary description of each contribution in order to allow end-users to make informed decisions about what to include in their task views. Each contribution can include multiple task information elements. The information directory needs to be updated whenever someone publishes a task or a contribution to a task. The actual task metadata and task information elements are not stored in the information directory, as it only acts as resolving and referring entity to the information repository.

The central *information repository* hosts the content of the task metadata and contributions, i.e., stored in RDF files. It only contains task metadata, e.g., in case of an attached task document it contains a document reference to a shared information space. It is implemented by a remotely accessible RDF store, which can be accessed using WebDAV, and uses Subversion for storing and versioning the RDF files. Using Subversion allows as well for version control on collaborative tasks, i.e., KWers can revert to older collaborative task states in the TM client. Tasks are expressed using RDF and multiple contributions can be merged into a single contribution or task by merging their corresponding RDF graphs. This requires a conflict-free ontology (e.g. no cardinality issues). Otherwise KWers need to resolve such conflicts manually.

We focus on Collaborilla's *information directory* containing two *metadata* types:

- Registry for tasks and contributions – Correlation between the task URI and a contribution and the place where the contribution can be loaded from.
- Metadata about a published task or contribution – It helps the KWer making a decision whether to take a task contribution into account or not, e.g., for a task contribution the contributing person and time is relevant metadata.

For the *information directory*, all involved collaborative task elements have a *globally unique identifier*, a URI, which is assigned on the initial contribution of the task to the CS. This allows the task to occur in several tasks, i.e. a contribution to a task could include elements from other tasks. Contributions have no own identity separate from the task identity, as a separate identity would hinder the merging on the level of RDF graphs. A contribution can be indirectly identified by the identifier of the task and which entity, i.e. the RDF file, it is expressed in.

Some contributions have a specific status, e.g., the initial task contribution by the creating KWer will be considered as the *original* contribution distinct from all *optional* contributions. Such collaborative task dependencies trigger different TM client behavior. If a dependency is referred to as original, the referring task cannot be loaded without it, since it contains essential information, i.e., basic task information. An optional dependency is a real contribution, i.e., the task can be loaded without it. It is up to the KWer whether or not to load optional dependencies. Contributions are *not dependent on earlier contributions*, i.e., all contributions are (in principle) expressed independently of each other. However, a contribution that provides additional metadata on a task element will not be visible unless the task element itself is visible.

3.3 Interaction between task management client and collaboration server

Collaborilla exposes its information directory resources via REST-based web services, which the Kasimir TM client invokes. We demonstrate the proposed interaction at the example of making a task available for collaboration, i.e., creating a collaborative task from the TM client on the CS, publishing it and thus making it available for contributions. This requires a set of steps, see below. Committing and updating collaborative tasks from the TM client follow a similar scheme.

- The KWer marked a task as shared and starts the commit transaction.
- The TM client invokes the CS to publish the task information to the *information repository*. This eventually returns the URL where task information can be requested from later on.
- This location is sent together with the task's URI to the *information directory*'s resolver service. It then can resolve an identifier into a real location.
- The task URI is sent together with the dependencies to the *information directory*'s referrer service, which then keeps track of the original task and eventually existing contributions to it.

4 Related Work

For *collaborative task support*, i.e., task information sharing and collaborative work, there are state-of-the-art applications both in research as well as commercial products. These applications approach the problem from different perspectives.

Knowledge management applications and *collaboration support* applications provide collaborative information spaces that enable KWers to share information, like e.g. in wikis or blogs. But, this spans only unstructured information, like e.g. putting meeting notes into a wiki page that is associated to a collaborative task. *Document management* applications as part of collaboration support applications provide document sharing. Here, documents are made available to all authorized task participants, but again no collaborative work on metadata is possible. *Business Process Management (BPM)* applications have coordination and collaboration functions enabling KWers to work together on tasks in a defined sequence as described by the BPM application. However, only rudimentary functions to collaboratively edit task metadata are provided, especially in conjunction with personal information.

For *collaborative metadata management*, there are alternative approaches for CSs supporting loose collaboration. *Annotea* [Annotea, 03] is a W3C RDF standard for collaboration around web pages, using shared metadata and annotation servers. It uses constructs similar to XPointers for locating the information to be annotated. However, the Collaborilla CS works on a higher level, i.e., it does not go into the structure of a document (e.g. a task) and instead uses plain URIs. E.g., individual task contributions change the overall task only by merging the corresponding RDF graphs. This leaves a contribution valid even if the structure of the original task changes. *Semantic indices* like *Sindice* [Tummarello, 07] provide another way of keeping track of collaborative tasks and their contributions. After announcing all task-related RDF data to such an index, a search for a task's URI would return a list of RDF files mentioning it. The task could then be loaded including all contributions. However, the lack of metadata describing the contributions makes it difficult for the KWer to decide about its relevance, e.g. it's not clear e.g. in which sequence the contributions were added. Another issue is the need of indexing the RDF data before querying the index which might result in unwanted delays before updating a task. These approaches' drawbacks do not occur with a registry supporting ad-hoc collaboration like Collaborilla.

5 Discussion & Future Work

We presented a collaborative task management application that enables loose and ad-hoc task collaboration among KWers and that explicitly integrates metadata into this collaboration process. It brings the information sharing process in collaborative task work to the next level. KWers can collaboratively work on task information in a familiar environment while preserving the semantic structure of the task information. This enables for the collaborating KWers a better information management using their semantic desktop applications. The personal information on the semantic desktop, including tasks, provides the metadata fundament. Using the Kasimir TM client, the KWer can now share this structured knowledge with co-working KWers. The Collaborilla CS enables KWers to collaboratively work on this information. The here proposed combination of both components, i.e., leveraging the CS approach for collaborative TM, uniquely offers the KWer both recognized usability and information management efficiency. This solution's implementation is work in progress, but both contributing components Kasimir [Grebner, 08] and Collaborilla (for collaborative Context-Maps [Ebner, 07], supporting conceptual modeling and distributed discourse management) are evaluated as working well today.

In future work, the presented application and its infrastructure can be leveraged to support metadata sharing and collaborative editing between semantic desktops beyond task information. In such an electronic portfolio system, like for example already realized with Confolio [Confolio, 07], ad-hoc and non-invasive collaboration is possible for all metadata like e.g. the whole KWer's personal information on the SSD.

In the current CS version, KWers may contribute independently of each other with separate contributions to the same task. However, just as anyone is allowed to link to any page on the web (if we ignore juridical concerns), anyone will be allowed to contribute to tasks that have been published. This is a security problem for closed work groups who need to have full control over the published task conglomerates. This will be addressed with the next, planned version of the Collaborilla CS.

References

- [Annotea, 03] Annotea, 2003, <http://www.w3.org/2001/Annotea/>.
- [Aperture, 05] Aperture, 2005, <http://aperture.sourceforge.net>.
- [Belotti, 02] Bellotti, V.; Ducheneaut, N.; Howard, M.; Smith, I.: Taskmaster: recasting email as task management. Workshop: "Redesigning Email for the 21st Century", CSCW. 2002.
- [Collaborilla, 08] Collaborilla, 2008, <http://collaborilla.conzilla.org>.
- [Confolio, 07] Confolio, 2007, <http://www.confolio.org>.
- [Davenport, 05] Davenport, T.H.: Thinking for a Living. Harvard Business School Press, Boston, MA, 2005.
- [Ebner, 06] Ebner, H.: Collaborilla - An enhancement to the Conzilla concept browser for enabling collaboration. Master's Thesis at the Department of Computer and Systems Sciences, Royal Institute of Technology (KTH), Stockholm, Sweden, 2006.
- [Ebner, 07] Ebner, H., Palmér, M., Naeve, A.: Collaborative Construction of Artifacts, Proceedings of 4th Conference on Professional Knowledge Management, Workshop Collaborative Knowledge Management (CoKM2007), Potsdam, Germany, 28-30 March 2007.
- [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, GITO-Verlag, 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.: The nepomuk project - on the way to the social semantic desktop. In Proceedings of I-Semantics' 07, JUCS, (2007) 201–211.
- [Kirsh, 00] Kirsh, D.: A few thoughts on cognitive overload. *Intellectica*, 30, 19-51, 2000.
- [Nepomuk, 06] Nepomuk Project Deliverable D3.1, Task Management model, 2006, <http://nepomuk.semanticdesktop.org/xwiki/bin/view/Main1/D3-1>.
- [Sauermann, 07] Sauermann, L., van Elst, L., Dengel, A.: Pimo - a framework for representing personal information models, In Proceedings of I-Semantics' 07, JUCS (2007), 270–277.
- [Tummarello, 07] Tummarello, G., Delbru, R., Oren, E.: Sindice.com: Weaving the Open Linked Data, ISCW, 2007.