CONTASK - Context-Sensitive Task Assistance in the Semantic Desktop

Heiko Maus[1], Sven Schwarz[1], Jan Haas[2], and Andreas Dengel[1 2]

[1] German Research Center for AI (DFKI GmbH), Kaiserslautern, Germany
[2] Computer Science Department, University of Kaiserslautern, Germany
`<heiko.maus,sven.schwarz,andreas.dengel>@dfki.de,jan.haas@senacor.com`

Abstract. In knowledge work, users are confronted with difficulties in remembering, retrieving, and accessing relevant information and knowledge for their task at hand. In addition, knowledge-intensive, task-oriented work is highly fragmented and, therefore, requires knowledge workers to effectively handle and recover from interruptions.
The Semantic Desktop approach provides an environment to represent, maintain, and work with a user's personal knowledge space. We present an approach to support knowledge-intensive tasks with a context-sensitive task management system that is integrated in the Nepomuk Semantic Desktop. The context-sensitive assistance is based on the combination of user observation, agile task modelling, automatic task prediction, as well as elicitation and proactive delivery of relevant information items from the knowledge worker's personal knowledge space.

Key words: Task management, proactive information delivery, personal knowledge space, user observation, agile task modelling, Semantic Desktop, personal information management

1 INTRODUCTION

Corporate work routines have changed vastly in recent years, and today's knowledge workers are constantly presented with the challenge of negotiating multiple tasks and projects simultaneously [1]. Each of these projects and tasks stereotypically requires collaboration with different teams, and dealing with a varied plethora of data, resources, and technologies. In contrast to traditional, static business processes, the task-oriented work of clerks today is often highly fragmented. Each interruption requires knowledge workers to mentally re-orientate themselves, and permanent task switches and disruptions are associated with a significant overhead cost [2]. Especially for complex, knowledge-intensive tasks requiring significant quantities of related documents and resources, reorientation entails a substantial cognitive overhead.

While an efficient task execution requires the processing of particularly task relevant information at hand, today's knowledge workers have to face the known information overload. An increasing amount of data is available, however, spread over various information sources such as email client, address book, local and remote file systems, web browser, wikis, and organizational structures. The efficient and successful processing of a task depends on the quality of finding and selecting the most relevant information for the task at hand. This represents a source of errors in daily knowledge work. Important information is not found, connections are overseen or relevant experts are not identified. The result consists of suboptimal problem solutions, unnecessary repetitions of already accomplished tasks or wrong decisions.

As a solution to the outlined problems of the knowledge worker, this paper proposes a context-sensitive task management system named *ConTask*. It focuses on the areas of knowledge capturing, knowledge reuse, and interruption recovery. By tracking the user's actions, the system provides automatic means to intelligently elicit task-specific relevant information items and, thus, capture a task's *context*. This is used for proactively delivering such context-specific, task-relevant knowledge to a user to ensure a reuse of valuable task know-how. Thereby, ConTask aims at the following goals:

– Automatically capture created/consulted information objects and assign these to tasks to capture task know-how and to structure the personal knowledge space in a task-centric way.
– Increase potential productivity for knowledge workers and reduce resource allocation costs by proactively providing relevant, task-related information and resources.
– Enable and ensure task-specific know-how reuse.
– Facilitate reorientation back into an interrupted task by reducing the cognitive and administrative task switching overhead.
– Improve task-specific assistance by learning from knowledge workers' feedback.

The paper is structured as follows: the next sections introduce the scientific background and give an overview on related work. Section 4 explains the main ingredients and concepts of ConTask. A summary and outlook on future work conclude the paper.

2 BACKGROUND

Various approaches identify business processes as a means for structuring a company's knowledge [3, 4]. As business processes form the core operational sequences of every company, their efficiency is critical for a company's success. Knowledge workers are integrated into crucial parts of the business processes, and the quality of their (procedural) know-how decides between success and failure (for the company). They are embedded in business processes where we are mainly interested in supporting knowledge-intensive tasks for know-how capture, provision, and reuse as well as assistance in multitasking for supporting the knowledge worker in his daily work directly.

Knowledge-Intensive Tasks Especially *knowledge-intensive* tasks entail the challenge of retrieving, structuring and processing information, e.g., for judging a case or for making crucial decisions. Knowledge-intensive tasks notably rely on processing large quantities of relevant information and coevally producing valuable knowledge to be reused in similar situations (later on). Aiming at preserving this valuable knowledge, tasks should be utilized to structure a knowledge worker's personal knowledge space consisting of various resources like documents, emails, and contact addresses, as well as real-life concepts such as persons, projects, or topics.

Typically, knowledge-intensive activities are explorative and not completely known a priori [5]. As many parts of their execution might not be predetermined, they can not be completely modelled in advance. On the basis of this, we introduce the concept of *weakly-structured workflows* consisting of knowledge-intensive activities with specific design decisions for applications supporting these workflows. They mainly incorporate the two aspects *lazy and late modelling* and the *strong coupling of modelling and execution of process-models*. Lazy modelling refers to an on demand refinement of process models initially only partially specified. This pays off for weakly-structured workflows as details of the execution of agile knowledge-intensive tasks are not known in advance. This aspect is strongly related to the coupling of modelling and execution of process-models. Starting with a partial model, weakly-structured workflows allow for dynamical refinement of the process model during its execution.

Our work resulted in the TaskNavigator [6, 7], a browser-based workflow system. It supports weakly-structured workflows through agile task management for teams, proactive information delivery (PID) based on the task context (mainly consisting of task name, description, and attached documents, i.e. text-based), process know-how capture and re-use. Evaluations have shown that a main drawback was the effort for users to maintain their task, i.e., by uploading documents to the task represented in the browser. The work presented here, is a continuation of the overall goal by integrating this into user's personal knowledge space and allowing a much easier management of their tasks embedded on their personal desktops.

Multitasking Nowadays knowledge workers are engaged in multiple tasks and projects in parallel (e.g., [8]). Several studies have shown that task-oriented knowledge work is highly fragmented [9]. Typically, knowledge workers spend only little time on a certain task before switching to another. Task switches and disruptions cause significant overhead costs (e.g., [2, 10, 11]). After an interruption, knowledge workers must reconstruct their task-specific mental state. This encompasses, in amongst other detail, memories around the task, including next steps to take, required resources, critical factors and deadlines.

In addition to the cognitive overhead, restructuring of the desktop and physical work environment is also often required. Resources such as documents, websites, emails, and contact addresses must be relocated and utilized. The retrieval of these various task-specific resources represents a challenging and time consuming problem. Furthermore, the cognitive challenge to remember all task-specific relevant information can result in significant difficulty for the successful completion of a given task. Short term memory loss concerning critical resources or other task-related information can have major ramifications.

The frequent interruption of knowledge-intensive tasks is often a contributing factor to workplace stress and frustration. Studies show that constant interruptions in these situations lead to changes in work rhythm, mental states and work strategies [2]. Often this results in attempts to compensate lost time through an accelerated and therefore even more stressful work pace. The higher workloads and additional effort associated with frequently switching tasks increase both pressure and frustration for today's knowledge workers.

3 RELATED WORK

Many different approaches focus on assisting knowledge workers with knowledge-intensive tasks and in multitasking work scenarios.

The TaskTracer system [12] consists of an user observation framework collecting events from various office applications. The system utilizes the observed user actions and applies machine learning algorithms for automatically predicting the user's current task and for associating accessed information items like files or web pages with the elicited task [13]. This enables a task-specific provision of these information items.

The APOSDLE Project [14] represents a similar approach. The project integrates task management, e-learning, knowledge management and communication systems. APOSDLE utilizes user observation to support the task-centric provision of suitable learning material and to associate knowledge artefacts with corresponding tasks. The user observation framework is realized by software hooks on the operation system level. Observed user actions are reported to the task predictor component, a machine learning component that serves for task prediction and task switch detection.

The OntoPIM [15] project suggests an architecture for a task information system including a monitoring system for user observation. Observed user actions are interpreted by an inference engine to elicit information relevant to the user's tasks and to proactively provide these items. This task-specific PID is integrated into a selected target application (e.g., into a web browser). Alternatively, the windows context menu is enriched with task-related information. Instead of integrating the task-specific assistance into several applications, our approach aims at integrating the PID into a task management system. This reduces the amount of applications having to be adjusted and enables the agile task modelling to directly use the captured and elicited task-specific knowledge.

A task management system embedded in the personal knowledge space along the vision we presented in [4] is KASIMIR [16]. It focuses on capturing, evolving, and providing process patterns from an organisational repository. The tasks within the processes are enriched with information items from the Nepomuk Semantic Desktop (see next section) by user interaction similar to ConTask. Similar work on identifying process patterns but coming from the process management side and embracing task management is the Collaborative Task Management (CTM) approach [17]. The main difference of both approaches from ConTask is the lack of user observation and PID our system applies and their steps further to apply evolving process patterns in process management. We will investigate the combination of these approaches in the joint project ADiWa[1] started in 2009.

4 INGREDIENTS OF ConTask

ConTask is based on the following base components to achieve a context-oriented personal task management: a task management system, the Semantic Desktop, and a framework for retrieving user context. Based on these components, ConTask supports with

[1] http://www.adiwa.net

proactive information delivery and agile task modelling. It enables observing task management, providing a task prediction, and allowing relevance feedback and learning. These components and concepts will be detailed in the remainder of this section.

Task Management As mentioned in Section 2, in order to support knowledge workers in working with knowledge-intensive tasks directly on their desktops, a task management system called *TaskPad* was developed. TaskPad provides the possibility to work on personal tasks and to synchronize tasks from different sources such as TaskNavigator, where the tasks were part of some agile workflows. TaskPad allows to access, attach, and upload documents to TaskNavigator. As both TaskNavigator and TaskPad are fully RDF/S based, they rely on the ontology for weakly-structured workflows developed in [5]. Besides this, TaskPad provides the usual task management functionality such as maintaining tasks, taking notes, attaching URLs, notes, or documents, and filtering the task list[2]. Fig. 2 shows the task list as Task Diary and the Task Editor.

Semantic Desktop To represent and maintain the user's personal knowledge space we use and integrate with the Nepomuk[3] Semantic Desktop which transfers the idea of the Semantic Web to the user's local desktop (for a recent overview, see [18]). It serves to capture and represent the knowledge worker's personal mental models [19]. This personal knowledge space consists of real world concepts such as persons, places, projects or topics, as well as, the connections and relations between them (see Fig. 1). Documents contain information about these concepts and represent the knowledge worker's individual background, tasks or personal interests. In this context, the Personal Information Model Ontology (PIMO) serves to formalize and structure the personal knowledge space (a PIMO excerpt can be seen on the upper left side with PIMO classes).[4] It is the core of the Semantic Desktop and provides the possibility to associate real world concepts with resources, such as documents, emails or contact details, for personal information management.

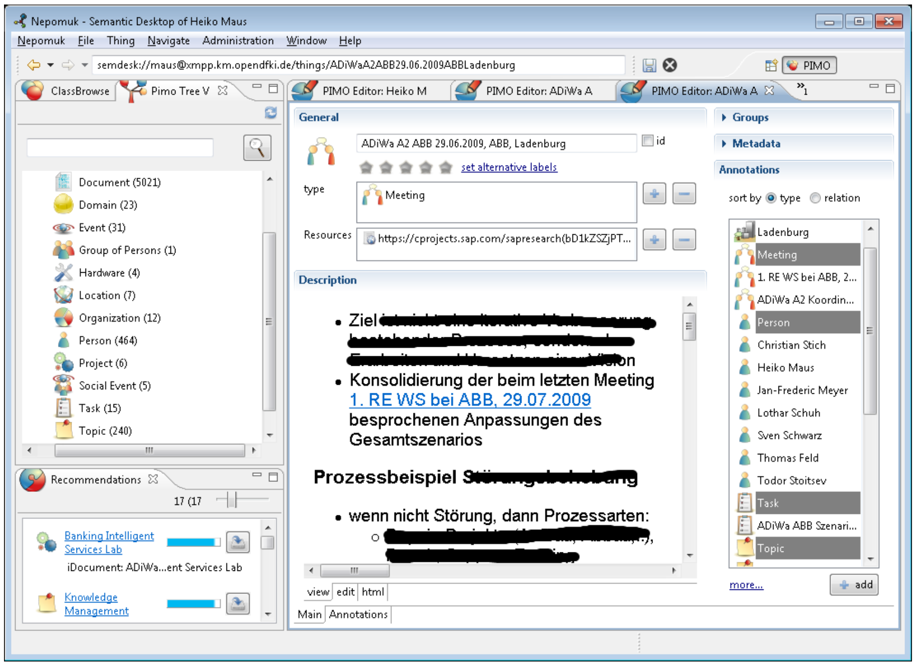
For example, Fig. 1 shows personal notes taken during a meeting using a Semantic Wiki embedded in the Nepomuk Semantic Desktop. Here, the meeting itself is an instance of the PIMO class `Meeting` together with linked concepts such as attendees, previous meetings or resources such as the calendar entry in MS Outlook. The wiki text shown is mixed with concepts, such as the project it belongs to, topics mentioned in the meeting, etc. Using a concept within the text (e.g., by auto-completion) allows to browse to the concept within the wiki text as well as adds relations to the meeting instance automatically. Furthermore, an ontology-based information extraction (iDocument, see [20]) analyzes text and provides proposals of concept which might fit for the current thing (lower left tab). Thus, during everyday usage, the PIMO evolves with relevant concepts of the user's work. As the meeting is then linked

User Context The Context Service elicits the user's work context, which is a snapshot consisting of contextual elements with relevance to the user's present goal or task [21,

[2] the filters are based on SPARQL queries

[3] `http://nepomuk.semanticdesktop.org/`

[4] The PIMO is represented in RDF/S – the basic language of the Semantic Web – and is able to include different vocabularies resp. ontologies, i.e., it can be adapted to different domains.



semdesk://maus@xmpp.km.opendfki.de/things/ADiWaA2ABB29.06.2009ABBLadenburg

ADiWa A2 ABB 29.06.2009, ABB, Ladenburg

type: Meeting

Resources: https://cprojects.sap.com/sapresearch(bD1kZSZjPT...

- Document (5021)
- Domain (23)
- Event (31)
- Group of Persons (1)
- Hardware (4)
- Location (7)
- Organization (12)
- Person (464)
- Project (6)
- Social Event (5)
- Task (15)
- Topic (240)

- Ziel [redacted]
- Konsolidierung der beim letzten Meeting 1. RE WS bei ABB, 29.07.2009 besprochenen Anpassungen des Gesamtszenarios

Prozessbeispiel [redacted]

- wenn nicht Störung, dann Prozessarten:
  - [redacted]

Sort by: [✓ type] relation

- Ladenburg
- Meeting
- 1. RE WS bei ABB, 2...
- ADiWa A2 Koordin...
- Person
- Christian Stich
- Heiko Maus
- Jan-Frederic Meyer
- Lothar Schuh
- Sven Schwarz
- Thomas Feld
- Todor Stoitsev
- Task
- ADiWa ABB Szenari...
- Topic

Recommendations 17 (17

- Banking Intelligent Services Lab
  iDocument: ADiWa...ent Services Lab
- Knowledge Management

Fig. 1: Nepomuk Semantic Desktop: semantically-enriched meeting notes (some text obscured) with PIMO relations (among them the task accessible in ConTask as shown in Fig. 2)

22]. Each contextual element corresponds to an entity from the user's PIMO. Contextual elements also contain a value describing the *certainty* for actually belonging to the current context. As knowledge workers regularly switch their tasks and, hence, their work context, the Context Service maintains several so called *context threads*. Each context thread represents a context snapshot associated with a certain task and contains all information items from the user's PIMO that are relevant to that task. To enable context-sensitive desktop applications, the Context Service provides an API, with the capabilities to switch to a different context thread, to query all relevant information items of a certain thread and to receive information about context thread switches. This API is used to inform the Context Service whenever the user is switching to a different context thread and allows to correctly maintain context snapshots.

In this scenario, the User Observation Hub[5] (UOH) serves as a technical means to automatically observe the user's actions (see also Fig. 5). This is used for gathering evidence about relevant concepts belonging to the user's current context. The UOH includes an extensive user action ontology formalizing all types of so called *native operations* (*NOPs*) that are observable during daily knowledge work. The ontology comprises operations such as browsing a website, adding a bookmark, reading an email or accessing a file in the file system. To gather these user actions, the user observation framework provides a set of installable observers, which report the corresponding user actions to

[5] `http://usercontext.opendfki.de/wiki/UserObservationHub`

the UOH. The observer framework includes plugins for Mozilla Firefox and Thunderbird and a Windows file system observer. Observed user actions are gathered in the UOH and distributed to registered listeners. The Context Service is such a designated user observation listener receiving notifications about observed user actions.
As outlined, the allocation and retrieval of relevant information items represents a time consuming challenge for today's knowledge workers. Thus, ConTask aims at increasing potential productivity for knowledge workers and reducing allocation costs by proactively providing relevant, task-related information and resources.

Proactive Information Delivery The user's PIMO represents a range of concepts and resources the knowledge worker deals with during daily work. Therefore, elements in the worker's PIMO are taken to serve as items for the proactive information delivery (PID). Aiming at a task-centric work support, the information items are provided in a task-centric manner. Thus, the PID structures the personal knowledge space in a task-oriented way.

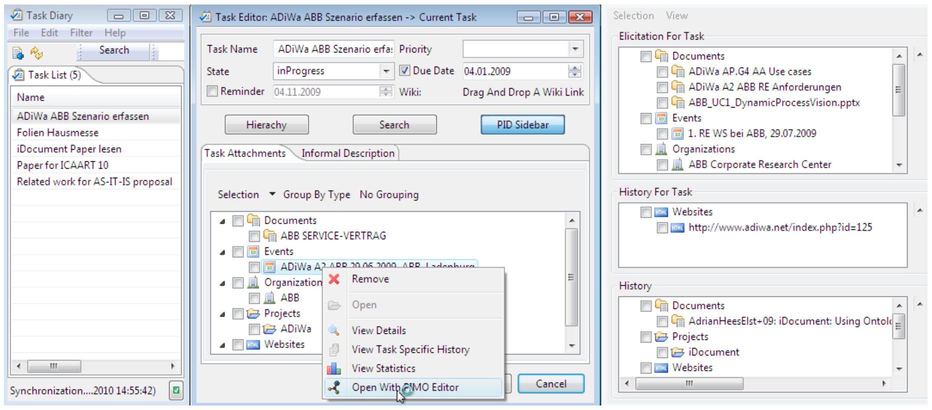
To automatically elicit task-specific relevant information items from the PIMO, ConTask utilizes the Context Service. Based on the user's interactions with desktop applications, such as browsing a website or writing an email, the Context Service elicits task-specific relevant information items using techniques of machine learning, entity recognition, and document similarity (for details see [23]). In addition, a *History Service* provides task-specific records of all accessed information items. Being registered as a UOH listener, the History Service maintains a detailed task-specific access history of PIMO concepts and resources (based on the observed user actions). Using the History and the Context Service, ConTask provides both directly accessed and automatically reasoned items.

Fig. 2 depicts a screenshot of ConTask, where the so called *PID Sidebar* proactively provides task-centric access to relevant information items. The PID Sidebar is part of the *Task Editor* which allows for easy consultation and modification of a task's properties. Below the task's name, status, and time constraints, the so called *task attachments* represent the information items *explicitly associated* with the task. PID Sidebar and Task Editor are located on the right-hand side of the screenshot. ConTask provides further interaction possibilities on both PID items and task attachments, such as viewing an item specific access history, opening resources with the associated application or viewing resp. editing PIMO elements with the PIMO Editor. The left-hand frame of the screenshot shows the user's task list (so-called *Task Diary*), which can be used to easily navigate or switch between tasks.

While the Task Editor contains manually attached information items, the PID Sidebar shows additional, potentially relevant information items. The History Service is used to deliver directly accessed items, and the Context Service is used to propose automatically reasoned items.

PID Categories The PID Sidebar provides information items in the following three categories:

– *Elicitation For Task*: This category contains the most relevant information items from the user's PIMO with respect to the task. It comprises both directly observed and also elicited items, that have not been directly accessed by the user.



Task List (5)

Name
ADiWa ABB Szenario erfassen
Folien Hausmesse
iDocument Paper lesen
Paper for ICAART 10
Related work for AS-IT-IS proposal

Synchronization....2010 14:55:42)

Task Name: ADiWa ABB Szenario erfa:
Priority:
State: inProgress
Due Date: 04.01.2009
Reminder: 04.11.2009
Wiki: Drag And Drop A Wiki Link

Documents
ABB SERVICE-VERTRAG
Events
ADiWa A2 ABB 29.06.2009, ABB, Ladenburg
Organization
ABB
Projects
ADiWa
Websites

Elicitation For Task
Documents
ADiWa AP.G4 AA Use cases
ADiWa A2 ABB RE Anforderungen
ABB_UC1_DynamicProcessVision.pptx
Events
1. RE WS bei ABB, 29.07.2009
Organizations
ABB Corporate Research Center

History For Task
Websites
http://www.adiwa.net/index.php?id=125

History
Documents
AdrianHeesElst+09: iDocument: Using Ontol
Projects
iDocument
Websites

Fig. 2: ConTask: Task Diary with open tasks; Task Editor with the current active task and the PID sidebar with proposed concepts from the PIMO and recently accessed resources (the context menu opens the PIMO editor in Fig. 1).

– *History For Task*: This section contains all PIMO elements for which a user action has been observed during the execution of this task. This category does not comprise any concluded or elicited elements, but only directly observed elements such as directly accessed files or websites. The items here stem from the History Service. Items already shown in the Elicitation For Task are omitted.
– *History*: This category contains the most frequently accessed PIMO elements that are not contained in the first two categories, but that have been directly accessed by the user since a certain point of time. These items also stem from the History Service.

These categories were chosen based on the following reasoning: *Elicitation For Task* provides the most relevant elicited information items computed by the Context Service's machine learning techniques. The PID Sidebar supports the user by proposing potentially relevant items which have not been explicitly associated with the task at hand. The effectiveness of these algorithms is a critical factor for the acceptance of this service. Low quality proposals will merely distract and annoy the user while good quality proposals have the potential to increase the user's work performance and foster a successful task execution. Particularly for newly created tasks, there is not enough explicit information available that can be used to automatically learn what this task is about. As a consequence, the Context Service is not able to determine good proposals. For these cases, the PID Sidebar's category *History For Task* contains all information items the user has accessed while working on the task. According to the user observation, these items have been recently touched by the user while the task at hand was *selected as the current task*.

The problem is, we can not expect the user to make every task and every task switch explicit—consider interruptions or phone calls, for example. The task management tool does not know whether the user is actually working on the currently selected task. As a consequence, one can not *automatically* attach recently touched items to the task. Rather ConTask proposes the items allowing users to attach them with a simple drag&drop gesture. The section *Task Prediction* will explain the how ConTask tries to keep track of the user's current task nevertheless.

For example consider the following scenario. The user accesses a website that is associated with the current task, but the Context Service does not determine this website as relevant, as it has to be further stimulated by being re-accessed by the user. In this case, the website would be provided in the *History For Task* section. If the relevance value of the website increased based on further stimulation, the website would be included into the *Elicitation For Task* category and would therefore be removed from the *History For Task*.

A similar reasoning explains the *History* category. As elaborated, today's knowledge workers deal with multiple task switches during their daily work. To assure correct association of user actions and accessed information items with the current task, ConTask uses automatic task prediction that will be explained in one of the following sections. However, as these task determination approaches might sometimes fail in highly multitasking work scenarios, some user actions could be associated with the wrong task. Thus, the corresponding information items might be proactively provided with this task instead of the task they actually belong to.

Manual task switches sometimes have fuzzy boundaries [13]. For instance if a knowledge worker is just reading a web page that is related to his current task but that leads him to switch to a new one: should the resource be associated with the old or with the new task? Additionally, if a knowledge worker explicitly consults a related task (and hence switches to the other task) for reusing there-stored know-how, the *History For Task* for the originating task will not contain the consulted material. A solution for this case provides the *History* category at the bottom of a task's PID Sidebar. It contains all information items which have been recently accessed by the user, but which have neither been associated with, nor elicited for, nor accessed during the task at hand. This overview over all recently accessed items allows an easy reuse of know-how across different tasks (via simple drag&drop a resource can be added as a task attachment).

Agile Task Modelling Based on the proactively provided PIMO elements, ConTask enables agile, on-the-fly task modelling. As knowledge-intensive tasks can not be completely designed or modelled before their execution, agile, lazy modelling was chosen to allow for task refinement during the execution process. Via context menu or drag&drop, new items, e.g., proposed items from the PID Sidebar, can be easily added as task attachments. This enables knowledge workers to associate information objects to tasks and to thereby classify work knowledge to independent task information units. The result is a task-centric structuring of the user's personal knowledge space. As lazy modelling explicitly allows to enrich tasks with relevant information during their execution, this approach reduces the up front modelling effort. Starting new tasks does not require completely designed task models. Knowledge workers are not enforced to enter whatever additional information before actually starting a task. This aligns with suggestions in [3, 24], where minimal analysis and initial modelling overhead are identified as one of the key requirements for successful business process-oriented knowledge management.

Furthermore, task-oriented structuring of the user's personal knowledge space enables intuitive and direct process know-how reuse. For example, while working on a report document for *project x*, the knowledge worker may remember an already completed similar report for *project y*. In case a relevant information item of *project y* is actually relevant for *project x*, ConTask allows to sight, attach, and reuse these items from one task to another with a few clicks. By explicitly attaching reused items to tasks, ConTask supports a light-weight capturing of task-specific knowledge and, hence, provides the basis for the integration of these tasks in organizational workflow systems such as TaskNavigator.

The explicitly attached and conserved task-specific items as well as the task-specific history also facilitate rapid reorientation when switching back to an interrupted task. ConTask reduces the mentioned cognitive and administrative overhead consisting of remembering and reallocating task-specific relevant information items. By double clicking attached or recent documents and resources, task-specific working states can be easily reconstructed and the task can be resumed without much delay.

As the PID Sidebar merely proposes potentially relevant items, ConTask provides the possibility of rejecting non-relevant or unsuitable suggestions. The system remembers these decisions and does not provide rejected information items again. Both kinds of feedback, acceptance and rejection, represent evidences for adapting the context of a task and serve as relevance feedback for the task-specific PID. That way, user feedback enables automatic learning and system improvement [3, 4, 6].

Observing Task Management During the execution of a task, the User Observation Hub (UOH) observes the user's task-specific behavior. Besides tracking actions inside office applications, ConTask also observes the user's explicit task management interactions such as reuse/open, drag&drop, and reject operations within the Task Editor and the PID Sidebar. Automatic, unobtrusive learning is applied to improve the task-specific PID. The observed user events are utilized for the following two goals:

– Relevance feedback on proactively provided information items.
– Automatic task switch detection based on the user's interaction with the system.

To integrate explicit task management operations into the user observation framework, the user action ontology has been enriched with *NOPs* for task operations. These additional NOPs resemble a task operations ontology capable of representing user interaction with an agile task management tool. Observable actions of the ConTask system which are also passed to the UOH are switching to some task or giving relevance feedback on PID items, for example. The task operations ontology is designed to represent NOPs from different task management applications. It comprises a minimal set of operations that are necessary for the purpose of relevance feedback on proactively provided information items and automatic task switch detection. It contains operations such as task creation, attribute modification such as by adding a task description or attachments. Further operations inform about accessing attachments or interacting with the PID sidebar. A *Task Observation Service* inside ConTask describes the user's task actions according to the NOP ontology and report these to the UOH.

Fig. 3 presents the created task operations ontology as UML class-diagram.

Task Prediction As outlined, Context and History Service form the basis of the task-specific PID. Both services observe the user's desktop activity and record or elicit

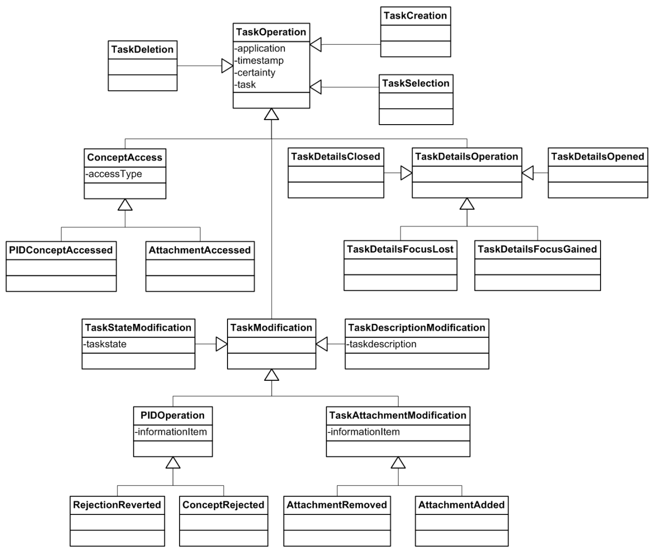


Fig. 3: Task Operations Ontology

relevant information items corresponding to a certain *context thread*. However, both services rely on explicit information about the *currently* processed task. If users are stressed or get interrupted, they will not use the task management tool for every tiny task deviation. Hence, not every task switch is technically observed. To compensate for this, ConTask contains a *Task Elicitation Service* realizing a task prediction. The Task Elicitation Service maps the tasks of the task management tool to context threads maintained by the Context Service. The observed user actions are treated as evidences to predict and update the currently active context thread. If the Context Service detects a context switch, a corresponding task switch is also proposed to the user. And vice versa: If the user switches a task, the Context Service is informed to switch to the corresponding context thread, too. Thus, this service serves to automatically and unobtrusively determine the user's current task and to utilize this to switch the Context and History Service to the corresponding thread.

The detection of the currently processed task is based on the observed user interaction with ConTask, such as opening the Task Editor window or attaching relevant information items to a task. For example, if the recently open task is different to the last one, this in interpreted as a task switch. To realize this, the task operations ontology is divided into the following two categories:

– *Operations with a direct implication on a task switch*: These operations are interpreted as strong indication for the fact that the user works on a certain task. Based on that, any of these user actions immediately leads to the conclusion of a task switch. They comprise all write access operations on a task, such as adding an attachment or editing the task's description, as well as opening the Task Editor window.
– *Operations with a weak implication on a task switch*: These operations only lead to a task switch conclusion if the following condition holds true: *No other operation with*

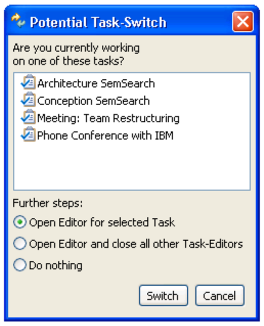
either weak or direct implication on a task switch occurs within a certain amount of time t. Operations with weak implications consist of the selection of a task in the task list and the event that occurs if the Task Editor becomes the active window.

The reasoning for the timeout value, associated with the selection of a task, is that knowledge workers sometimes browse their task list by clicking on each single entry. This serves to get an overview on their tasks and to determine which tasks are most critical at the moment. To avoid that each selection during browsing triggers a task switch and therefore context thread switch, selections are only interpreted as task switches after the timeout.

A similar reason explains the timeout value corresponding to the focus gain event of the Task Editor. As several Task Editor windows might be open at the same time, the user might quickly switch focus between two or more Task Editors to compare the corresponding tasks. The timeout value avoids that every focus change directly determines a task switch.

The automatic task elicitation serves to precisely determine which user actions and corresponding information objects occurred in the context of which task. This aims at increasing the Context and History Service's effectiveness. As the assistance is realized in an unobtrusive way, the user just needs to work and interact with ConTask, without having to deal with explicitly telling the system to perform a task switch. On the basis of this, the user is relieved from actively selecting the currently performed task. Since knowledge workers frequently perform task switches and face interruptions of their current task, automatic switch detection reduces interaction overhead and still guarantees that information items are associated with the correct task.

In addition to this, the Context Service also provides the possibility of automatically detecting whether the last user action(s) match better with different context thread(s) than the current one. In this case, the service notifies interested listeners about a *potential* switch. ConTask utilizes these switch detection capabilities. However, a task switch can not be performed automatically and, hence, the user is consulted via a specific popup listing potential task switches. (for an example see Fig. 4). The window slides up from the bottom of the screen and only remains visible for a short amount of time. If the user does not interact within this timeframe, the popup disappears.



Are you currently working on one of these tasks?

Architecture SemSearch
Conception SemSearch
Meeting: Team Restructuring
Phone Conference with IBM

Further steps:
[✓ Open Editor for selected Task]
Open Editor and close all other Task-Editors
Do nothing

Fig. 4: Potential Task Switch Window

Relevance Feedback and Learning The Task Elicitation Service also utilizes the observed interactions with ConTask for the transmission of feedback to the Context Service. Actions such as the assignment of a PID item to a task are interpreted as positive feedback. They increase the relevance value of the item in the corresponding context thread. On the other hand side, the user's rejection of a PID item or removal of an attached item from a task are interpreted as "negative" feedback.

The relevance feedback serves to increase the Context Service's effectiveness. The cycle of proactively providing contextual information to the user and transmitting user feedback to the Context Service leads to a better synchronization of context thread and task: Relevant information items from the context thread are proposed in the PID Sidebar. Some of them may be added to the task by the user (via drag&drop from the sidebar to the task attachments). In that case, the resulting feedback leads to increased relevance values of these items in the context thread. Fig. 5 shows how user observation data is utilized in ConTask to realize a knowledge improvement cycle.

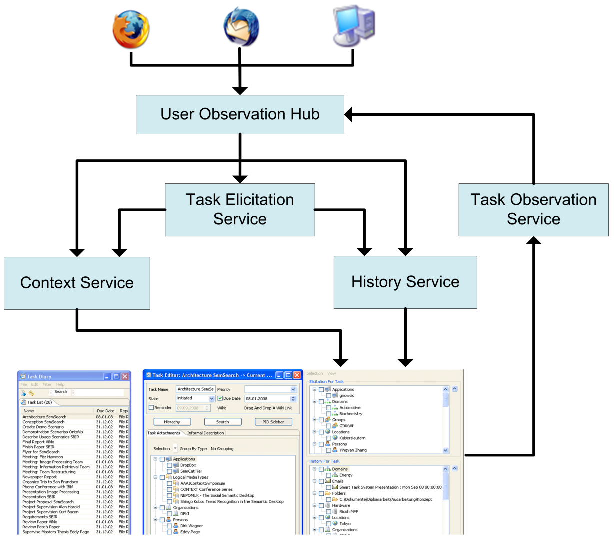


Fig. 5: ConTask components and interrelations

As an extension to the mentioned user observation providers, such as Mozilla Firefox, ConTask utilizes the Task Observation Service to become observable. Interested listeners, such as the Task Elicitation Service, receive the transmitted task events from the User Observation Hub. On the basis of the received task NOPs, the Task Elicitation Service transmits relevance feedback to the Context Service and switches both Context and History Service to the thread belonging to the current task. The History and Context Service gain their data from the UOH and complete the loop by providing information items for the PID within ConTask.

The Task Elicitation Service only depends on the task operations ontology and the assumptions on task-oriented work that are utilized for the purpose of task prediction.

The History and Context Service only depend on the NOP ontology formalizing the user's desktop activity. Thus, any task management application that is compatible with the task operations ontology can be integrated into the created scenario realizing the knowledge improvement cycle. Necessary integration steps involve the extension of the system's user interface with observation calls to the Task Observation Service. If the task management application includes a PID component and allows for agile task modelling, the Context and History Service could be utilized to proactively provide relevant, task-specific PIMO elements to the user.

Feasibility Study The current proof-of-concept implementation of ConTask delivers early indications that the system actually has the potential to assist the user while keeping the additional work at a minimum. In the long run, a robust context identification is essential for keeping the assistance scalable: The context identification algorithm is expected to estimate the correct context in most of the cases. We created a ground truth by tagging a large set of observed user operations, manually assigning user actions to "contexts". A ten-fold cross validation on this ground truth data shows that 78% of the operations are identified correctly, 9% of the guesses were incorrect, and 13% of the cases were not identified at all. Striving for a best-effort strategy, a relatively high number of unidentified cases (13%) is not considered harmful for the user's actual work. An amount of 9% incorrect context guesses is not very high, but this is a critical value as false identifications may lead to false context switch proposals and, hence, to disruptions of the user. One cause for the false identifications is that the user observation software does not recognize some of the user operations. Additionally, users mentally separated some contexts which were technically identical. Additional sensors providing evidences for additional contextual elements will reduce these problems. Hence, we will continuously enhance the user observation and context elicitation technology.

5 SUMMARY AND OUTLOOK

This paper addressed challenges in today's knowledge work: continuously increasing quantities of information, knowledge intensive tasks, and highly fragmented multitasking work scenarios.

The context-sensitive task management system ConTask was designed to address these challenges and alleviate the knowledge worker's job. ConTask is integrated into the Semantic Desktop and combines task management with context-specific assistance. The assistance is based on the combination of user observation, automatic elicitation and proactive information delivery of relevant information items from the user's PIMO. ConTask enables agile task modelling for defining tasks on the fly and striving for a task centric structuring of the personal knowledge space. Observation of the user's interaction with ConTask is utilized for relevance feedback and automatic task prediction to increase the precision of the PID while keeping the required task management to a minimum. Thereby, the system realizes a knowledge improvement and learning cycle.

As ConTask is only capable of detecting task switches to already existing tasks, a possible improvement would be an algorithm for detecting the user is working on a new task which is not yet reflected in the system. Similarly, making proposals for refining a task into subtasks based on the observations (e.g., by detecting that involved resources of a task can be separated in two topic clusters but within the task). This would significantly support agile task modelling.

We currently expand the observed area to the physical desktop of a user by using a digital camera and applying image recognition algorithms to recognize user actions with paper documents on the desk [25]. So far, recognizable actions are placing, removing, and moving a paper document on the desk as well as arranging a pile all enriching the user context. The user context gets enriched with actions such as *user placed a known document of the PIMO onto the desk* or *placed an unknown one* (including extracted text). Users can assign piles to concepts or tasks of the PIMO and later on browse the state of their physical desktop even from a concept or task. Next step will be a tighter integration of ConTask and the desk observation with specific actions for task management.

For getting better suggestions on PID, we currently investigate to embed the ontology-based information extraction system iDocument to extract PIMO entities in observed text snippets as well as to contextualize with the PIMO as background knowledge as it is done in the Nepomuk Semantic Desktop [20] and the TaskNavigator in [7]. In the ADiWa project we investigate the usage of the PIMO-based user context to contribute enriched task context to dynamic business processes and to capture process know-how from process participants. We especially investigate on how user tasks enriched with concepts from PIMO but also from group repositories can contribute to knowledge-intensive business processes.

Acknowledgements This work has been partly funded by the German Federal Ministry of Education and Research (BMBF) in the ADiWa project (01IA08006).

References

1. González, V.M., Mark, G.: Managing currents of work: multi-tasking among multiple collaborations. In: ECSCW'05: 9th European Conference on Computer Supported Cooperative Work, Springer (2005) 143–162
2. Mark, G., Gudith, D., Klocke, U.: The cost of interrupted work: more speed and stress. In: CHI'08: SIGCHI conference on Human factors in computing systems, ACM (2008) 107–110
3. Abecker, A., Hinkelmann, K., Maus, H., Müller, H.J., eds.: Geschäftsprozessorientiertes Wissensmanagement. xpert.press. Springer (June 2002)
4. Riss, U., Rickayzen, A., Maus, H., van der Aalst, W.: Challenges for Business Process and Task Management. Journal of Universal Knowledge Management **0**(2) (2005) 77–100
5. Elst, L.v., Aschoff, F.R., Bernardi, A., Maus, H., Schwarz, S.: Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation. In: IEEE WETICE Workshop on Knowledge Management for Distributed Agile Processes (KMDAP'03), IEEE Computer Press (2003)
6. Holz, H., Rostanin, O., Dengel, A., Suzuki, T., Maeda, K., Kanasaki, K.: Task-based process know-how reuse and proactive information delivery in TaskNavigator. In: CIKM'06. ACM Conference on Information and Knowledge Management. (2006)
7. Rostanin, O., Maus, H., Zhang, Y., Suzuki, T., Maeda, K.: Lightweight conceptual modeling and concept-based tagging for proactive information delivery. Ricoh technology report 2009, no. 35, Ricoh Co Ltd., Japan (December 2009)

8. González, V.M., Mark, G.: “Constant, constant, multi-tasking craziness”: managing multiple working spheres. In: CHI’04: SIGCHI conference on Human factors in computing systems, ACM (2004) 113–120
9. Czerwinski, M., Horvitz, E., Wilhite, S.: A diary study of task switching and interruptions. In: CHI’04: SIGCHI Conference on Human factors in computing systems, ACM (2004) 175–182
10. Iqbal, S.T., Horvitz, E.: Disruption and recovery of computing tasks: field study, analysis, and directions. In: CHI’07: SIGCHI Conference on Human factors in computing systems, ACM (2007) 677–686
11. Mark, G., Gonzalez, V.M., Harris, J.: No task left behind?: examining the nature of fragmented work. In: CHI’05: SIGCHI conference on Human factors in computing systems, ACM (2005) 321–330
12. Stumpf, S., Bao, X., Dragunov, A., Dietterich, T.G., Herlocker, J., Johnsrude, K., Li, L., Shen, J.: The TaskTracer system. 20th National Conference on Artificial Intelligence (AAAI-05) (2005)
13. Stumpf, S., Bao, X., Dragunov, A., Dietterich, T.G., Herlocker, J., et al.: Predicting user tasks: I know what you’re doing! 20th National Conference on Artificial Intelligence (AAAI-05) (2005)
14. Lokaiczyk, R., Faatz, A., Beckhaus, A., Goertz, M.: Enhancing Just-in-Time E-Learning Through Machine Learning on Desktop Context Sensors. In: Modeling and Using Context. Volume 4635/2007 of LNCS. Springer (2007) 330–341
15. Lepouras, G., Dix, A., Katifori, A.: OntoPIM: From personal information management to task information management. In: SIGIR’06 Personal Information Management Workshop. (2006)
16. Grebner, O., Ong, E., Riss, U.: Kasimir - work process embedded task management leveraging the semantic desktop. In: Multikonferenz Wirtschaftsinformatik. (2008) 1715–1726
17. Stoitsev, T., Scheidl, S., Flentge, F., Mühlhäuser, M.: From personal task management to end-user driven business process modeling. In: Business Process Management. Volume 5240 of LNCS., Springer (2008)
18. Grimnes, G.A., Adrian, B., Schwarz, S., Maus, H., Schumacher, K., Sauermann, L.: Semantic desktop for the end-user. i-com **8**(3) (12 2009) 25–32
19. Sauermann, L., Bernardi, A., Dengel, A.: Overview and Outlook on the Semantic Desktop. In: 1st Workshop on The Semantic Desktop at ISWC’05. Volume 175 of CEUR Proceedings. (November 2005) 1–19
20. Adrian, B., Klinkigt, M., Maus, H., Dengel, A.: Using idocument for document categorization in nepomuk social semantic desktop. In Pellegrini, T., ed.: i-Semantics: Proceedings of International Conference on Semantic Systems 2009. JUCS (2009)
21. Schwarz, S.: A context model for personal knowledge management applications. In: Modeling and Retrieval of Context, Second Int. Workshop, MRC 2005. Volume 3946 of LNCS., Springer (2006) 18–33
22. Schwarz, S.: Context-Awareness and Context-Sensitive Interfaces for Knowledge Work Support. PhD thesis, University of Kaiserslautern (2010)
23. Schwarz, S., Kiesel, M., van Elst, L.: Adapting the multi-desktop paradigm towards a multi-context interface. In: HCP-2008 Proc., Part II, MRC 2008 – 5th Int. Workshop on Modelling and Reasoning in Context, TELECOM Bretagne (June 2008) 63–74
24. Holz, H., Maus, H., Bernardi, A., Rostanin, O.: From Lightweight, Proactive Information Delivery to Business Process-Oriented Knowledge Management. Journal of Universal Knowledge Management **0**(2) (2005) 101–127
25. Dellmuth, S., Maus, H., Dengel, A.: Supporting knowledge work by observing paper-based activities on the physical desktop. In: 3rd Int. Workshop on Camera Based Document Analysis and Recognition (CBDAR’09). Proceedings. (2009)