Context-Aware Knowledge Services

Andreas S. Rath
Know-Center Graz
Inffeldgasse 21a/II, 8010 Graz, Austria
arath@know-center.at

Nicolas Weber
Know-Center Graz
Inffeldgasse 21a/II, 8010 Graz, Austria
nweber@know-center.at

Mark Kröll
Know-Center Graz
Inffeldgasse 21a/II, 8010 Graz, Austria
mkroell@know-center.at

Michael Granitzer
Know-Center Graz
Inffeldgasse 21a/II, 8010 Graz, Austria
mgrani@know-center.at

Olivia Dietzel
m2n - consulting and development gmbh
Marienstraße 10, 4020 Linz, Austria
dietzel@m2n.at

Stefanie N. Lindstaedt
Know-Center Graz
Inffeldgasse 21a/II, 8010 Graz, Austria
slind@know-center.at

ABSTRACT
Improving the productivity of knowledge workers is an open research challenge. Our approach is based on providing a large variety of knowledge services which take the current work task and information need (work context) of the knowledge worker into account. In the following we present the DYONIPOS application which strives to automatically identify a user's work task and then contextualizes different types of knowledge services accordingly. These knowledge services then provide information (documents, people, locations) both from the user's personal as well as from the organizational environment. The utility and functionality is illustrated along a real world application scenario at the Ministry of Finance in Austria.

Author Keywords
Personal Information Management, Context Observation, Usage Data, Knowledge Representation, Context-Aware Information Retrieval.

ACM Classification Keywords
H1.2 [Information Systems] User/Machine Systems, H.3.1 [Content Analysis and Indexing]: Indexing methods, H5.2 [Information interfaces and presentation]: Graphical user interfaces.

INTRODUCTION
Our vision of personal information management involves supporting the individual knowledge worker in her work tasks by providing effortless access to related personal as well as organizational knowledge. This perspective puts the individual knowledge worker [8] – her flexibility, her needs, her goals, her interests – into the center of attention. At the same time organizational knowledge structures (e.g. process and domain models) which used to dominate the knowledge management discussions are moved off-focus – but are used to understand a user's context better. Thus, we strive to understand the individual knowledge worker in her relationship to organizational and community knowledge structures in order to support her better.

When putting the knowledge worker in focus we need to learn as much as possible about this individual and her habits. This needs to be done automatically in the background and if possible without explicit user interaction. We also want to understand this individual context in relation to surrounding organizational environments to gain even deeper insights. Based on this understanding of the user context our objective is then to support the task of the knowledge worker by providing knowledge services. Such knowledge services will include recommendation of relevant tasks and processes previously performed by other users, intelligent delivery of relevant resources from the personal as well as organizational memory, and visualization of relationships between resources, people and locations contained within the resources.

Based on this understanding and vision of personal(ized) information management we have developed the DYONIPOS application [28]. DYONIPOS on the one hand aims to automatically detect a user's work context and on the other hand provides a number of context-aware knowledge services to support the user. We have applied DYONIPOS within the IT-department of the Federal Ministry of Finance (FMF) in Austria.

The structure of this paper is as follows: First we shortly discuss the application setting at the FMF. Based on this background we then lead the reader through two usage scenarios (learning PIM habits, supporting PIM) which illustrate the DYONIPOS functionality and explain their value to individual FMF employees. After each of the scenarios we discuss the functionality and their implementation in more detail and compare our approaches to related work. Finally we present the architecture of the DYONIPOS system and an early analysis of the usage data collected within the first FMF usage trial.

FMF APPLICATION SCENARIO SETTING

The Federal Ministry of Finance (FMF) in Austria is a knowledge-intensive administration institution that strives at supporting and enhancing their knowledge-intensive tasks and processes by using cutting-edge technology. In the context of DYONIPOS we specifically are involved with the IT-department of the FMF. Their role is to provide us with a real business environment where we can test and refine the algorithms, the techniques and the technology that are developed in DYONIPOS.

At the beginning of the DYONIPOS project we visited the organization and talked to our key users about their tasks and processes they are involved in and how they are executed. From these interviews we modeled not only their processes but also their tasks. We asked for detailed task and role descriptions. Such a task description included the roles and persons involved, documents used and previous and follow-up tasks. Based on the insights we gained from the interviews we started to decide which sensors and sensor data are required for recording the important features of the user's actions.

USE SCENARIO

Roles involved in the scenario

Sarah is a 32 years old employee of the FMF. She has been working in the IT section of the ministry for 3 months and is not yet familiar with all the peculiarities of her job. Her task among others is the evaluation of project tenders from diverse IT and consulting companies. She is responsible for checking the formal criteria, the background relations and the accuracy to the project tender. Sarah's position also includes presenting her evaluation results to the project management team.

John is a 48 year old man who is in the middle management and responsible for project management in the FMF's IT section. He has been working in the ministry for several years and has gained knowledge in various areas of the section. When John was at Sarah's age, he occupied the same position as she does at the moment.

Top Level Overview of Usage Scenario

Sarah uses the DYONIPOS client in order to accomplish the task "Project Tender Evaluation". In this task she has to assess a product concept catalogue handed in by an IT-company and to prepare a presentation with her results for the next management team meeting. While parts of the task are clear to her, there are open issues regarding how to perform the assessment itself. DYONIPOS assists Sarah in finding useful information and related experts. Sarah contacts her co-worker John who helps her to sort out the details.

LEARNING PIM HABITS

This section shows how DYONIPOS learns the ways how people manage their work information at FMF. With the term "learning PIM habits" we mean the discovery of relations among information objects, user interactions, tasks and processes. Our approach is to observe the user's behavior on her desktop, i.e., the habits how the user organizes, creates, stores, shares/distributes and manipulates knowledge.

Use Scenario Learning PIM Habits

On a Monday morning, Sarah comes into her office, switches on her computer, and starts the DYONIPOS client. After activating the context observation, she begins reading her emails and sees that her boss asks for an assessment of the recently handed in project concept catalogue of the IT-Experts Consulting company. While Sarah is reading this email, DYONIPOS automatically records the following information: the email's metadata (subject, to, from, attachments, sent time, etc.) and also the content of the email. Via text analysis DYONIPOS recognizes objects such as companies, persons, locations within the mail text and relates them to each other. Sarah navigates via the windows explorer to the place where she normally stores project tender evaluations, creates a new folder and stores the attached files there. She opens the project concept catalogue file in Microsoft Word and starts reading. When Sarah comes across a link to the IT-Experts Consulting company, she opens it in her favorite web browser to find more background information about this company. Some details about the reference customers of this company she copies & pastes to a newly created notes document. She returns back to the project concept catalogue and finishes reading it. All the interactions she has done so far have been recorded as event blocks and are displayed on the action view of the DYONIPOS Task Recognizer XP sidebar (see Figure 3, top right). Since Sarah knows she will be doing such assessments rather frequently, she wants to train DYONIPOS in recognizing assessment events in the future. This is why she takes the time to select the relevant assessment event blocks, assigns them to a new task and labels it as "IT-Expert Consulting Project Tender Assessment".

Features of DYONIPOS – Learning PIM Habits

Context Capturing

DYONIPOS observed the reading of the word document and the different web pages, the switch between the applications, the creation and saving of the notes file, the storing of the attachments and also the copy & paste actions including clipboard content. On a low level, the user interactions (mouse clicks, keyboard inputs) and their durations were recognized. The current content focus of the user, e.g., which paragraph of the word document the user reads at the moment, is also stored and semantically related.

We built various context observation sensors which gather the context information from standard office and email applications and from the operating system itself. An overview of the sensors and their capabilities can be found in Tables 1 and 2. For further details about the technologies and techniques behind the context observations we would like to refer to [23].

Application	Observed Metadata and Data
Microsoft Word	document title, document url, folder, user name, language, text encoding, content of visible area, file name
Microsoft PowerPoint	document title, document url, document template name, current slide number, file name, language, content
Microsoft Excel	spreadsheet title, worksheet name, folder, spreadsheet url, user name, authors, language, content of the currently viewed cell, file name, file uri
Microsoft Internet Explorer	currently viewed url, urls of embedded frames, content as html and content as plain text
Microsoft Explorer	currently viewed folder/drive name, url of folder/drive path
Mozilla Firefox	currently viewed url, urls of embedded frames, content as html
Mozilla Thunderbird	(html/plain text) content of currently viewed or sent email, subject, unique path (uri of email/news message) on server, user's mail action (compose, read, send, forward, reply), received/sent time, email addresses and full names of the email entries: from, to, bcc, cc
Novell GroupWise email client	(create, delete, modify, and distribute) tasks, notes, calendar entries, and todos, and data about email handling like in the Mozilla Thunderbird application

Table 1. Context sensors for standard applications. This table lists applications from which we retrieve data about the user's context. The sensor data from the respective application sensors is listed next to the applications name.

Sensor	Observed Metadata and Data
File System Sensor	copying from/to, deleting, renaming from/to, moving from/to, modification of files and folders (file/folder url)
Clipboard Sensor	clipboard changes, i.e., text copied to clipboard
Network Stream	header and payload content from network layer packets (http, ftp, nttp, smtp, messenger, ICQ, Skype, …)
Generic Windows XP System Sensor	mouse movement, mouse clicks, keyboard input, window title, date and time of occurrence, window id/handle, process id, application name

Table 2. Context sensors for the operating system. In deep hooks into the operating system allow us to record user interactions and network transfer data on a fine-granular basis.

Context Representation

The conceptional model we use to represent the context of a user is supported by semantic technologies. We use an ontology that describes the user's context by taking into account the user's interactions, the resources on which the user acts on and the corresponding (automatically extracted and manually generated) metadata relations. The process and task executions and corresponding models are encapsulated in the user's context representation as well.

The user's context model can be seen as a pyramid, the knowledge worker's semantic pyramid. It connects the user's *actions* with *resources* and *information needs* (see Figure 1). The semantic pyramid describes the continuous evolution of contextual information through different, semantic layers. Starting at the bottom with events that are executed by one knowledge worker and ending with processes where many knowledge workers can be involved. Figure 1 illustrates the semantic pyramid from a knowledge worker's perspective. Each level of granularity (events, event blocks, tasks and processes) provides a different representation of the data regarding the semantic quality.

User interactions with the system and reactions from the system to the user's interactions represent *events*. Events form atomic units within an event block. Events can be user inputs, such as mouse movements, mouse clicks, starting a program, creating a folder, a web search, or opening a file.

An *event block* is defined by a sum of events which are totally ordered. An example of an event block is "editing a document on page 2". Event blocks are formed using predefined static rules, which map a set of events to an event block. Event blocks are combined into *tasks* by grouping together similar event blocks into semantic sets. Thus one resulting set represents one task.

A *task* equals a well-defined step in a process, which can not be divided into sub tasks and in which only one person is involved. In DYONIPOS tasks are not modeled yet. The tasks are automatically learned from low-level events and event blocks which are a result of the sensor data aggregation (see Task Learning paragraph).

A *process* emerges by concatenating individual tasks that were conducted by a number of different knowledge workers. The transitions from tasks to processes are not visualized in Figure 1 but can be interpreted as reaching the top of the (knowledge worker's) semantic pyramid. In comparison to tasks where only one person is involved, well-defined steps from several persons constitute a process.

Information needs are needs that emerge from an active information request or from a change of the knowledge worker's context. In the first situation the knowledge worker directly requests information from an *information source*, e.g., an external or local database, an application's data storage, or the web. The discovery of a change in the knowledge worker's task and process context requires in-depth application inspection.

Resources are elements that fulfill the knowledge worker's information needs, e.g., documents, links, task and process guides or topic experts. Resource detection and modeling is as performed as a visualization of the discovered resources.

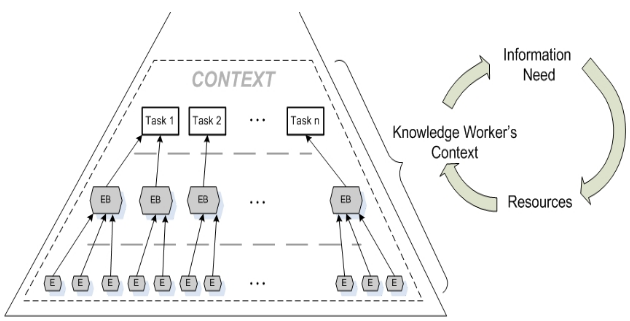

Figure 1. The semantic pyramid from a knowledge worker's perspective comprises event (E), event block (EB) and task (T) layer. The user's context is described by events, event blocks and tasks. Information needs can occur on each layer and are satisfied by resources.

Task Learning

The manual assignment of the event blocks to a task is used to train a classifier. The training is based on the context features we observe. Some context features, which provide valuable information, e.g., the window title or the application name, are used directly for the training. Other context features, e.g., user input or the content of a currently viewed document or web page, require preprocessing (e.g., stemming, stopword removal). Our approach on task learning and recognition is to utilize not only content features but also structural ones [14].

Related Work

Context observation can be found in many areas like e-learning (Apposdle [15], LIP [27]), knowledge work support (SWISH [21], Haystack [1], TaskTracer [7], Phooney [12]), personal information management [10], information retrieval [11] and attention metadata [29]. Context observation data can also be seen as attention metadata. Attention metadata captures information about the user's handling of digital content. It can be utilized to describe what a user likes, dislikes, reads, publishes, produces, watches and listens to. Several research projects have been and are carried out to collect this attention metadata from the user's desktop. Once the information about the user's behavior is collected, a wide area of possibilities for exploitation is open. Our approach of the context capturing process compared to the mentioned ones, is quite similar. The difference lies in the context processing, i.e., how the sensor data is aggregated, processed and semantically enriched (see "Supporting PIM section"). A further add-on is the usage of semantic technologies to describe the user's context and the relations to the used information objects (resource). Automatic text analysis algorithms, such as named entity recognition and text clustering, enrich the context with discovered relations. Relations like "this document belongs to certain topic clusters" and "specific experts, locations, and organizations/institutions are mentioned in a specific document, webpage or email" are integrated.

In the research area of *context* there are several context models around. In 1994 the term "context-aware", was defined by Schilit and Theimer who referred to locations, identities of nearby people and objects and changes to those objects [26]. In 2001 Dey et al. [6] provided a well elaborated list of historical definitions of the term context and introduced a more general one which describes context as "*any information that can be used to characterize the situation of entities that are considered relevant to the interaction between a user and an application, including the user and the application themselves*". At the moment various applications and systems make use of contextual information (see above).

In the area of *task learning*, Shen et al. [25] use a classifier to predict the current task based on features, e.g., window title and the file pathname, extracted from the window in focus. In [21] task assignment is based on relations of the windows on the user's desktop. Lokaiczyk et al. [18] evaluate five algorithms for task detection based on observed context features and report achieved accuracy of over 85% whereas the support vector machine (SVM) approach with the sequential minimal optimization algorithm performs best. Currently, we are also utilizing SVMs for the classification task. The difference to our approach is that we are also experimenting with graph kernels which have not been investigated yet for task classification to the best knowledge of the authors [14]. In the area of task-centered information management Catarci

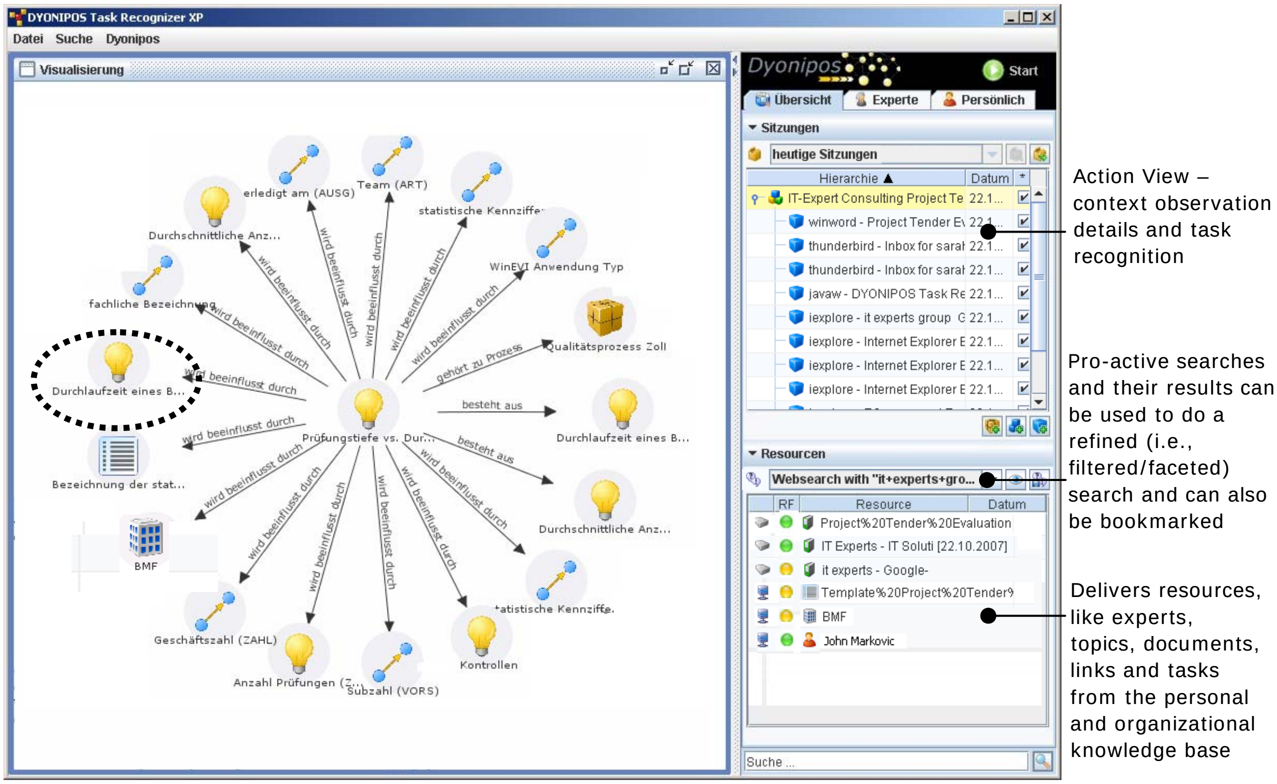

Figure 2. DYONIPOS Task Recognizer - While the person is doing her work, our application observes her interaction with various resources, categorizes them and displays a summary on the first part of the graphical user interface - the action view. The green "Start" button with the "Play" symbol enables the context observation process. Event block to task assignments can be done manually by the user (learning/training phase) or automatically via task recognition (classification phase). Further it is possible to create new tasks to allow the inclusion of "offline" activities, e.g., a meeting in a conference room or a telephone conference. Context specific resources, like documents, experts, links, topic clusters and related tasks are shown in the resources view on the bottom right. All interactions and used resources, pro-active and active searches with their results are semantically related to each other. The personal and organizational knowledge base can be visualized and navigated. In Figure 3 a navigation step is shown, whereas the concept "Durchlaufzeit eines B..." (dotted black circle on the left) is extended.

et al. [4] developed a top-down approach to task inference in which users define the main aspects of tasks using forms of declarative scripting. They developed a task specification language and proposed an architecture for supporting task inference.

SUPPORTING PIM

This section presents an overview of the knowledge services of the DYONIPOS application for enhancing the PIM experience. We show how we ease and pro-actively support the information finding and re-finding processes [5]. Furthermore we give an outlook of how we image a possible transfer of insights and knowledge gained from an individual's PIM to the organizational memory. For showing the possibilities with the DYONIPOS application, we continue with our scenario.

Use Scenario Supporting PIM

Continuing from "Use Case Learning PIM Habits", where Sarah has organized the received documents and gathered some additional background information about the IT-Experts Consulting company, she is now ready to begin her assessment. Actually, she is not completely ready to start because she is missing the template for project tender evaluations. Luckily, DYONIPOS has observed her context so far and can now provide her with the template for project assessments right away. Without requiring an active search, DYONIPOS delivers resources like templates, experts, topics and similar tasks. As Sarah recognizes that DYONIPOS has found the template she was looking for, she opens it and starts to fill in the form fields. Since some inputs of the form fields are similar to the handed in project concept catalogue, she copies and pastes the data from one document to the other one. DYONIPOS tracks which elements from one document are pasted to the other one and builds a relation between them. One assessment criterion asks for the credibility and liquidity of the applying company. Sarah is not quite sure how to handle this criterion, so she glances at the DYONIPOS sidebar (see Figure 2) in order to find an expert. John has been proposed as an expert in the field of Sarah's current task. She does not hesitate to contact him in order to get clarification on the unclear assessment criterion since his contact details (email, office building, phone number, etc.) are readily available. Sarah writes an email to John about this issue. John replies with an explanation how he would handle such a criterion assessment based on his experience. The email correspondence and its metadata are recognized by DYONIPOS and related to her current task.

DYONIPOS is well connected to the organizational knowledge spaces like email, asset management, workflow, shared file systems and so on. In order to get a clearer picture about how colleagues handle similar tasks, Sarah selects one of the tasks, that have been suggested to be similar, and visualizes it with all its relations (see Figure 2 and Figure 3). On the screen a star visualization shows up with the task in the center. Around the task she finds the used resources, the involved persons with their organizational roles, the process the task belongs to, topic clusters and some more relations. Sarah re-finds the template for project tender evaluations. Since this template has a relation to the completed project tender evaluation document, she is able to navigate to this one and open it. For getting a clearer picture about this specific assessment, she visually browses through the knowledge space. After Sarah has finished her task, she decides to contribute the important facts about her task execution to the organizational knowledge base. They can be used later by others as a guideline. Since she decides which facts and relations are submitted to the organizational knowledge base, she feels in control and her privacy is ensured.

Features of DYONIPOS – Supporting PIM

Pro-active context-based information retrieval

The context of the user and her behavior is continuously observed when the context observation is enabled. The information collected about the user's context is utilized to search in the personal and organizational knowledge space. The results (documents, links, experts, …) are non-intrusively presented and continuously updated in the resource view of the DYONIPOS application. The actuality assures that the user has the most suitable results available right away when she needs them. The importing, indexing, retrieval, and text analysis functionalities are provided by our service-oriented knowledge discovery framework, the *KnowMiner* [13]. For more details on the pro-active context-based information retrieval we would like to refer to [23].

Knowledge Relationship Browsing

DYONIPOS is well connected. Based on the m2n Intelligence Management framework [19] integration capabilities and its connectors to diverse organizational information sources, DYONIPOS makes the personal and organizational knowledge space browsable and navigatable. The user can zoom onto a fine-granular level where she sees metadata about a certain resource. Furthermore she can view the knowledge space on a top level and get an

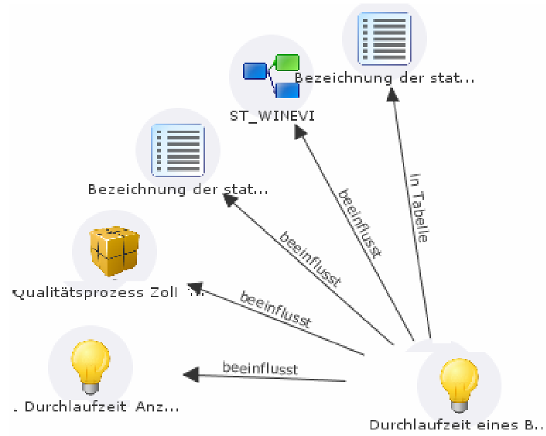

Figure 3. From every concept in the ontology it is possible to browse to a top level view or to dig deeper to inspect the relations and concepts in greater detail.

overview of the organizational structure of her company for example. In DYONIPOS the m2n Intelligence Management framework provides access to the shared file systems, the email/calendar server, the asset management system and the EDIAKT [9] system. All the information objects from various sources are inter-related by an ontology.

Knowledge Sharing

The knowledge Sarah gained about the execution of her task, has been contributed to the organizational knowledge space. With this contribution Sarah makes sure that others can view how she managed the task fulfillment and which resources were of great value to her. When sharing information about actions and resource usage on a fine-granular level, privacy is a hot topic. By empowering the user to select what should be contributed to the organizational knowledge base, her privacy is assured.

Related Work

Just in time information retrieval (JITIR) [24] combined with context observations is comfortable and time-saving (no active query formulation is needed) for the user. The information retrieval step enhanced with context information offers the possibility of significant quality improvements [11] and ad-hoc accuracy of the results. DYONIPOS combines both, JITIR and context-aware information retrieval.

DYONIPOS ARCHITECTURE

In this section a brief overview of the architecture of the DYONIPOS system is given. DYONIPOS uses the KnowMiner framework and the m2n Intelligence Management framework as underlying bases for development. On top of these two frameworks the DYONIPOS functionality and graphical user interface components were developed. The DYONIPOS system has been designed as a client-server architecture which allows multiple Dyonipos Task Recognizer XP clients to connect to one central server. The architecture of the client is shown in Figure 4 and the one of the server in Figure 5.

Client Architecture

The client consists of five major components, the context observers, the Dyonipos Task Recognizer, the KnowMiner framework, the m2n client and the graphical user interface component for the Dyonipos Task Recognizer. The components are presented in Figure 4. The following paragraphs describe the components in greater detail.

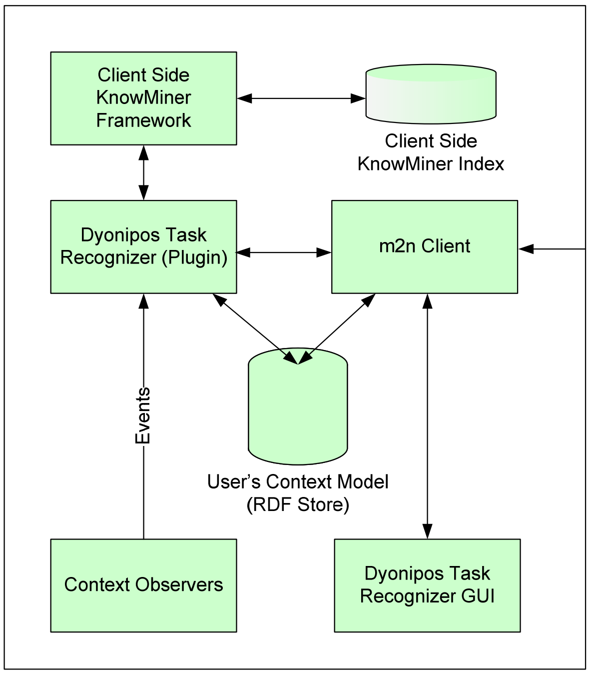

Figure 4. The client side architecture of Dyonipos.

Context Observers/Sensors

Context observers, also referred to as context sensors, are programs, macros or plug-ins that provide the functionality to observe the user's behavior on her desktop. We distinguish sensors based on the origin of the sensor data they deliver.

System sensors: System sensors capture data related to system events by utilizing a deep hook into the operating system. System events are input device data streams, like e.g., keystrokes, mouse movements, mouse clicks, and information about clipboard events (copy & paste actions) and file system changes.

Application sensors: Application sensors collect information about the user's behavior when interacting with specific applications. Typically application sensors are plug-ins (e.g. Mozilla Thunderbird), macros or small programs (e.g., network stream sensor) that utilize application specific libraries for the usage data collection.

Current status of the context capturing mechanisms is that we have already developed context observers for capturing interactions of the user with the Microsoft Windows XP Environment (keystrokes and mouse clicks), standard Microsoft Office applications (Excel, Word, PowerPoint), the Windows Explorer, the Internet Explorer 6 and 7, the Mozilla Thunderbird, the Mozilla Firefox browser, and the Novell GroupWise email client. A complete list of the data we capture from applications and the operating system is presented in Table 1 and Table 2.

Dyonipos Task Recognizer

The Dyonipos Task Recognizer is the component on the client side which is responsible for the handling of contextual data. Receiving the observed contextual data from the context sensors, further processing and analyzing the data are the main tasks of this component.

The raw contextual data is continuously sent from multiple sensors to the Dyonipos Task Recognizer via an XML based event stream. From the received event stream event objects are built and handed over to the context processors and context analyzers. Further processing includes the aggregation to blocks of related events (event blocks) and resource discovery and analysis. Resource discovery results in an indexing of the found resources and the extraction of metadata based on the KnowMiner framework (see next paragraph). The context analyzers are also responsible for detecting a user information need (e.g., based on a web or desktop search request, the user's interaction behavior with documents or presentation slides) and for fulfilling this need by triggering a search request to the KnowMiner framework. All the findings of the processing and analysis step are used to derivate relations between actions, resources and information needs. These relations contribute to the user's context model which is a continuously growing RDF model/graph. The Dyonipos Task Recognizer is developed as a plug-in for the m2n client.

Client Side KnowMiner Framework

The KnowMiner framework is a service-oriented knowledge discovery framework. It provides access to indexing, search, information extraction, clustering, and classification services that we utilize in the Dyonipos Task Recognizer. Indexing and search services are used to store the discovered resources in a fast index for text based information retrieval. The information extraction services provide named-entity recognition for finding persons, locations and organizations. In DYONIPOS we use a whitelist mechanism to map the found entities to concepts in the user's context model. The clustering services we utilize to generate a first suggestion of a mapping from event blocks to tasks. For the user it turned out to be easier and faster to refine a task instead of selecting all the event blocks and assign them manually to a task. The KnowMiner classification services support the task learning of tasks from selected features.

m2n Client

m2n Intelligence Management is a framework for a model-driven development of applications based on Java technology. These applications use RDF graphs in order to represent all system data. Application schemes and data, access rights, the graphical user interface, but also all parameterizations and configurations of the program logic are semantically described. The m2n client provides only some elementary functions, in particular the remote method invocation (rmi)-based methods for communicating with the server. Nevertheless the client is able to work off-line, i.e., without being connected to the server. Thus, the client gains the status of a complete application. Again the basic functions can be extended by means of plug-ins. Next to components for access rights and document management, an execution engine is also part of the Dyonipos client. It reacts on all incoming events and is able to forward them to the server-side execution engine, and vice versa. The graph visualization and the graphical user interface are provided as plug-ins as well. Furthermore the client facilitates a cache in which current and often used data are temporarily stored. This minimizes the frequency of accessing the server and client-side repositories.

Dyonipos Task Recognizer GUI

As mentioned above, the graphical user interface (GUI) of the Dyonipos client is yet another plug-in. The whole GUI, i.e. the description of the windows, of the individual buttons and labels, and also the functions that are performed as a result of certain user events, are modeled and presented in RDF notation. This allows a configuration of the complete functionality of the GUI during runtime. Finally, the m2n GUI plug-in uses the information contained in the RDF graph in order to render a web or java swing interface, whereby the Dyonipos Task Recognizer GUI represents the second variant.

Server Architecture

The server side of DYONIPOS consists of the following five major components – the m2n Intelligence Management server, the KnowMiner plug-in for the server, the connectors to various data sources, the Dyonipos Server Repository and the Task & Process Mining component. A picture of the server-side architecture is presented in Figure 5. A description of the components follows in the next paragraphs.

Server Side KnowMiner

The KnowMiner framework on the server side is packaged in a plug-in for the m2n server. This plug-in has two main functionalities. The first one is to store the received context models from the m2n server in the Dyonipos Server Repository. The second one is to provide indexing, searching, clustering and information extraction services to the m2n server. The m2n server which can also be seen as an integration facility for multiple data sources uses the KnowMiner framework services for indexing and searching diverse data formats. During the indexing process information extraction is carried out to extract persons, locations and organizations, like on the client side. In addition to the already described features of the client side KnowMiner framework, we implemented a possibility to cluster all the available documents. In the clustering process we automatically compute (topic) labels for the resulting clusters. This feature enables the user to explore and access available resources, e.g., documents, persons, and organizations, in the global knowledge base based on topics.

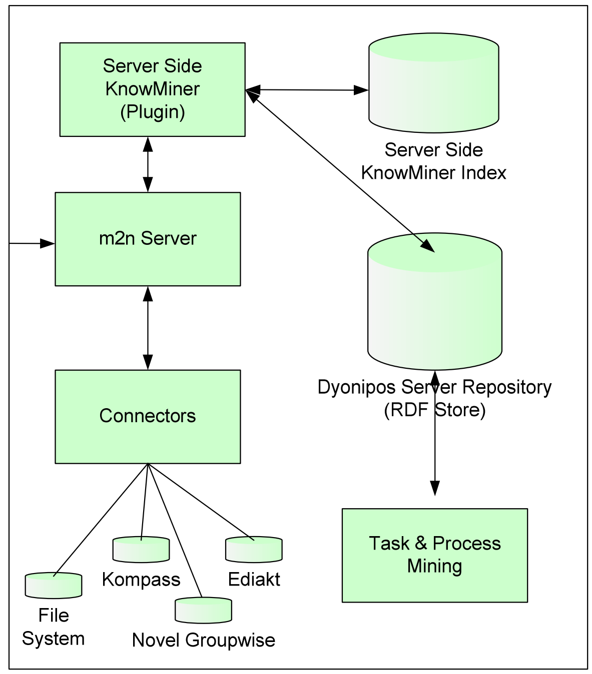

Figure 5. The server side architecture of Dyonipos.

Dyonipos Server Repository

The Dyonipos Server Repository is a Sesame RDF store [3]. In this RDF store we save and relate context models from multiple users. Furthermore this Sesame RDF store acts as a central component for doing data analysis on a multiple user scale (see next paragraph Task & Process Mining). For assuring the user’s privacy, the user decides actively what kind of data leaves her machine and is stored about her. It was an important requirement of our key users in the ministry of finance to be able to actually look at the pure data that will be transmitted. In the first prototype test we received about 4.5 million triples from our key users.

Task & Process Mining

The Task & Process Mining component is directly connected to the Dyonipos Server Repository and is the central point where we conduct context data analysis. Several algorithms for statistical evaluations, feature selection and task analysis and classification were implemented. We are planning to utilize the ProM framework [22] for task- and process mining.

m2n Server

A single server provides multiple applications as virtual hosts. Additional features can be added by plug-ins, which register their methods at a central execution engine. Besides the KnowMiner framework plug-in this is one of the most important components of the Dyonipos server. The sequence and parameters of execution of those methods is defined by means of rules, which are in turn represented as RDF graphs. Thus, both, the program logic and data stream processing, are model-driven as well. Finally, access to the different repositories is provided by connectors (see below).

Connectors

Connectors translate data from different repositories into a RDF-based representation, which conforms to the ontology of the respective application. The m2n Intelligence Management framework provides two types of connectors. The first variant reads out the data from a repository, transforms it into a RDF notation and writes the data into a suitable triple store, whereto the queries to the original repository are forwarded as well. The second variant does not transform the data, but translates RDF queries into the querying language of the respective repository using dynamic mappings. The results are then delivered as RDF. The data that comes from the connectors is combined with the user's context model. For instance, a person is found via information extraction in a document and mapped to a concept person. If this person is also available in the asset management system, further details about the person from the external system is contributed to the user's context model, e.g., room and telephone number.

Currently, there are three connectors available for DYONIPOS: a Kompass[1] connector for the FMF's people database, a file system connector for a rights-managed access of server-side documents, and an EDIAKT [9] connector. Further ones for Novell GroupWise [20] and Livelink [17] are planned, but not implemented yet.

PRELIMINARY DATA ANALYSIS

For getting a clearer picture about how knowledge is organized and handled by FMF employees, we created a questionnaire. Eight key users were asked about what kind of information sources they use and how they access them, the tools and applications they utilize during their work and about the enactment of tasks and processes. The main findings are summarized in the following:

- The most frequently used applications are Microsoft Word, Excel, PowerPoint, Internet Explorer, Novell GroupWise and the EDIAKT system.
- Used information sources are the internet, files and folders on personal hard drives and email inbox, shared documents on the FMF's servers, paper based documents, social networks and events (seminars and conferences).
- Search is predominantly performed via internet search (Google, Wikipedia), Windows Search functionality for searching personal and server files and the integrated search facility of their email application.
- 50% to 90% of all their work is exclusively done on the computer (mean μ=74.4%).
- 25% to 90% of their communication is computer based (μ=60.6%).
- 2% to 60% of their time they spend for searching for information (μ=16.6%).

Work interruptions are caused by emails, telephone calls, personal enquiries, planned discussions and meetings.

The testing period of the first prototype was approximately 5 weeks and 8 key users from the FMF installed and used the first prototype of the DYONIPOS client. The following summarizes our findings:

- Key users employed the DYONIPOS client together with the context observation component for durations between 3 and 21 days (μ=9.1 observed days per user).
- 800 MB of usage data from the execution of real tasks and processes were collected (μ=98.3 MB per key user).
- The number of the semantic relations we computed from the recorded sensor data from all users combined is approximately 4.2 million triples.
- The most used applications were Microsoft Word with ~55000, Novell GroupWise with ~40000 and the Microsoft Internet Explorer with ~7000 interactions.
- The data included around 253000 user interactions with applications or the operating system, so called events, from 8 key users. From these events we computed 4782 event blocks which are a rule-based grouping 23 of events to blocks of related events (μ=45, median=5 events per event block).
- The users manually assigned the 2947 event blocks to 124 tasks. The event blocks were aggregated from 116763 events.

In order to improve the usability of DYONIPOS and to reach a better understanding of the subjectively experienced utility of the different knowledge services, we informally interviewed the key users. This first informal feedback of the key users included following statements:

- They appreciated the pro-active context-sensitive search capabilities of DYONIPOS.
- They fancied the possibility to use the DYONIPOS client as a desktop search engine.
- Some key users mentioned that they did not like the manual assignment of event blocks to tasks because of the additional effort it took. Also the benefit of this action was not immediately apparent to them. Instead of the manual task assignment and labeling they would prefer an automatically or semi-automatically computed task

[1] Kompass is the internal FMF's asset management system.

assignment – with the possibility for user corrections. The second prototype will have the requested automatic and semi-automatic task recognition features included. A direct benefit for the user's labeling effort will then be an automatic labeling of the detected tasks.

FUTURE WORK

The preliminary data analysis we showed in the previous section is only a starting point. We are planning to conduct further statistical analysis of the FMF 1st prototype test data, like correlation analysis between resources & tasks and events & event blocks & tasks and significance analysis of context features for specific tasks or actions in general.

Since tasks are represented within an ontology, graph-based machine learning techniques seem suitable for identifying similar task usage patterns and for deducing task models. Experiments based on the collected task execution data, is one of our next steps.

Visualization techniques for various concepts/elements in the knowledge space of the user and/or organization is also one of our next steps. We are heading into the direction of evaluating different techniques for resource relations visualization (non-directed graph), tasks and processes (directed graphs). One of our great challenges we see in identifying and comparing process and task executions and presenting them visually for identifying deviations.

The next evaluation in a real working environment will include the new graphical user interface showed in Figure 2, the further developed context sensors and an automatic task recognition and training component. The 2nd prototype test has started in December 2007 and lasts for 5 weeks with approximately the same number of key users.

CONCLUDING REMARKS

This paper presented a first realization of our context-aware knowledge services approach. A first type of knowledge services here ranges from context capturing to task learning services. These knowledge services aim at automatically identifying a user's work context consisting of work task and current information needs. Based on an improved understanding about the user a second type of knowledge services has been illustrated: from pro-active information retrieval to knowledge relationship browsing. Here the focus is on providing users pro-actively with information needed in order to execute their work task. A third type of knowledge services includes information sharing between people who are involved in similar work tasks. The knowledge services introduced here are first representatives of a multitude of other possible knowledge services. In addition, totally different types of knowledge services can be envisioned: competence identification services [16], work-integrated learning services [15], collaboration services, etc.

ACKNOWLEDGEMENTS

The project results have been developed in the DYONIPOS project (DYnamic ONtology based Integrated Process OptimiSation). DYONIPOS is financed by the Austrian Research Promotion Agency (http://www.ffg.at) within the strategic objective FIT-IT under the project contract number 810804/9338.

The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria.

REFERENCES

1. Adar, E., Kargar, D., Stein, L. A. Haystack: per-user information environments. In *Proc. CIKM '99*. ACM Press (1999), 413-422.
2. Budzik, J., Hammond, K.J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proc. IUI '00*, ACM Press (2000), 44-51.
3. Broekstra, J, Kampman, A, Harmelen, F. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. In *The Semantic Web - ISWC 2002*, Lecture Notes in Computer Science, Springer, 2342, 54—68, 2002.
4. Catarci, T., Dix, A., Katifori, A., Lepouras, G., and Poggi, A., Task-Centered Information Management. In *DELOS Conference 2007 on Working Notes*, Pisa (Italy), 253-263, 2007.
5. Cutrell, E., Dumais, S.T., Teevan, J. Searching to eliminate personal information management. Commun. ACM (2006), 49, 1, 58-64.
6. Dey, A.K., Salber, D., Abowd, G.D., A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Human-Computer Interaction (2001), 16.
7. Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L., Herlocker, J.L. TaskTracer: a desktop environment to support multi-tasking knowledge workers, In *Proc. IUI '05* (2005), 75-82.
8. Drucker, P. F. Post-Capitalist Society, New York, Harper Business, 1994.
9. Ediakt, standardized e-Form exchange system: http://www.cio.gv.at/elektronischerAkt/, http://www.egov-iop.ifib.de/downloads/GPC_IOP_in_EDIAKT2_Austria.pdf.
10. Fernandez-Garcia, N., Sauermann, L., Sanchez, L., Bernardi, A. PIMO Population and Semantic Annotation for the Gnowsis Semantic Desktop, In *Proc Semantic Desktop and Social Semantic Collaboration.* CEUR-WS (2006), 202.
11. Fuhr, N. Information retrieval - from information access to contextual retrieval, Designing Information Systems.

Festschrift für Jürgen Krause, 47–57. UVK Verlagsgesellschaft, 2005.

12. Van Kleek, M., Bernstein, M., Karger, D.R., Schraefel, M. 2007. Gui --- phooey!: the case for text input. In *Proc. UIST '07*, ACM Press (2007), 193-202.

13. Klieber, W., Granitzer M., Sabol, V., Kienreich, W., Kern, R. KnowMiner: Ein Service orientiertes Knowledge Discovery Framework. In *Proc. Berliner XML Tage*, 2006.

14. Kröll, M., Rath, A.S., Weber, N., Lindstaedt, S., Granitzer, M. Task Instance Classification via Graph Kernels, In *Proc. Mining and Learning with Graphs 2007*, Firenze, Italy.

15. Lindstaedt, S. N., Ley, T., Mayer, H. APOSDLE - New Ways to Work, Learn and Collaborate, In *Proc. WM2007*, 381-382, GITO-Verlag (2007), Berlin.

16. Lindstaedt, S., Ulbrich, A. Integration von Arbeiten und Lernen – Kompetenzentwicklung in Arbeitsprozessen, Semantic Web - Wege zur vernetzten Wissensgesellschaft, Berlin, Germany, Springer Verlag (2006).

17. Livelink, RSH GmbH, http://www.livelink.de/.

18. Lokaiczyk, R., Godehardt, E., Faatz, J., Görtz, M., Kienle, A., Wessner, M., Ulbrich, A. Exploiting Context Information for Identification of Relevant Experts in Collaborative Workplace-Embedded E-Learning Environments. In *Proc. EC-TEL 2007*, Springer LNCS (2007), 217-231.

19. m2n Intelligence Management Framework, m2n - consulting and development gmbh, http://www.m2n.at.

20. Novell GroupWise, Collaboration and Productivity Software, http://www.novell.com/products/groupwise/

21. Oliver, N., Smith, G., Thakkar, C., and Surendran, A.C. 2006. SWISH: semantic analysis of window titles and switching history, In *Proc. IUI '06*, ACM Press (2006), 194-201.

22. The ProM Framework, http://prom.sf.net/.

23. Rath, A.S., Kröll, M., Lindstaedt, S.N., Granitzer, M. Low-Level Event Relationship Discovery for Knowledge Work Support, In *Proc. WM2007*, 227-234, GITO-Verlag (2007), Berlin.

24. Rhodes, B., Maes, P. Just-in-time information retrieval, IBM Systems Journal 39 (2000), 3–4, 685–704.

25. Shen, J., Li, L., Dietterich, T., Herlocker, J. A hybrid learning system for recognizing user tasks from desktop activities and email messages. In *Proc. IUI '06*, ACM Press (2006), 86–92.

26. Schilit, B., Adams, N., Want, R. Context-aware computing applications. In Proc. 1st International Workshop on Mobile Computing Systems and Applications, IEEE (1994), 85-90.

27. Schmidt, A. Impact of Context-Awareness on the Architecture of Learning Support Systems, Architecture Solutions for E-Learning Systems, Idea-Group Publishing, 2007.

28. Tochtermann, K., Reisinger, D., Granitzer, M., Lindstaedt, S.N. Integrating Ad Hoc Processes and Standard Processes in Public Administrations, In *Proc. OCG eGovernment Conference 2006*, Austria, 2006.

29. Wolpers, M., Najjar, J., Verbert, K., Duval, E. Tracking Actual Usage: the Attention Metadata Approach, International Journal Educational Technology and Society, ISSN: 1436-4522, 2007.