Deutsches
Forschungszentrum
für Künstliche
Intelligenz GmbH

Document

D-04-01

EPOS: Evolving Personal to Organizational Knowledge Spaces

Milestone M1

J.-T. Bähr, P. Dannenmann, L. van Elst,
A. Hust, A. Lauer, H. Maus, S. Schwarz

January 2004

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH

Postfach 20 80
67608 Kaiserslautern, FRG
Tel.: + 49 (631) 205-3211
Fax: + 49 (631) 205-3210
E-Mail: info@dfki.uni-kl.de

Stuhlsatzenhausweg 3
66123 Saarbrücken, FRG
Tel.: + 49 (681) 302-5252
Fax: + 49 (681) 302-5341
E-Mail: info@dfki.de

WWW: http://www.dfki.de

EPOS: Evolving Personal to Organizational Knowledge Spaces

Milestone M1

**J.-T. Bähr, P. Dannenmann, L. van Elst,
A. Hust, A. Lauer, H. Maus, S. Schwarz**

DFKI-D-04-01

EPOS: Evolving Personal to Organizational Knowledge Spaces

Milestone M1

J.-T. Bähr, P. Dannenmann, L. van Elst,
A. Hust, A. Lauer, H. Maus, S. Schwarz

Abstract

EPOS will leverage the user's personal workspace with its manyfold native information structures to his personal knowledge space and in cooperation with other personal workspaces contribute to the organizational knowledge space which is represented in the organizational memory.

This first milestone presents results from the project's first year in the areas of the personal information model, user observation for context elicitation, collaborative information retrieval and information visualization.

Contents

Chapter 1

Motivation

EPOS will leverage the user's personal workspace with its manyfold native information structures to his personal knowledge space and in cooperation with other personal workspaces contribute to the organizational knowledge space which is represented in the organizational memory.

In the following, we motivate the research thread of EPOS with the help of the application scenario within which a consultant's work at a German IT-consulting company is investigated.

1.1 A consultant's workspace reflects his activities, concepts, views, and way of thinking

A consultant is involved in several projects, often in different domains. Each consultant is an expert in his specific field, but also asks his colleagues for information. The consultants have access to various information sources. The consulting company provides processes, handbooks, project templates and high-level structures. Furthermore, a consultant is obliged to take part in the company's knowledge management activities.

A consultant's workspace reflects his activities, concepts, views, and way of thinking. He relies on his personal (knowledge) workspace, as well as, his connections to "peers" (colleagues), and therefore maintains them. However, the organizational KM activities are perceived as obtrusive, additional work.

This application scenario now leads to observations, which are described by the following sections.

1.2 Effort in structuring individual information spaces provides valuable input for Knowledge Management

Consultants use various tools for conceptualizing their domains: generic operating system structures (file folders) and dedicated information management applications (address books, mail tools, outliners, mind managers).

Advantages of these native structures concern knowledge utilization and acquisition: They reflect, at least temporarily, the consultant's individual view and can therefore easily be exploited by the consultant. They are regularly extended and maintained.

Problems arise from the lack of clear semantics: With easily extendable structures, often redundant and contradictory models are created which are difficult to utilize by automatic services and hard to share with other knowledge workers. The (ascribed) ad hoc semantics is typically not stable. Therefore, the usefulness over time even for one knowledge worker is not given.

In EPOS, a formally grounded personal information model, fed by the native structures, bridges individual and organizational Knowledge Management.

Chapter 2 introduces the personal information model.

1.3 Consultants spend a huge amount of their time with creative work within their personal workspace

Consultants' behavior provide clues about their goals and needs, because they build up habits for solving problems, and their actions are triggered by specific goals, tasks, and roles. Their work is highly context-sensitive.

KM tools are usually separated from the user's creative desktop work. The KM tools know nothing about a user's current situation, and thus, the biggest amount of work is not adequately supported.

A consultant could work much faster if his normal, daily work would be supported by context-aware services. However, services shall come at no cost for the user, and services shall not disturb his true work.

Observing the user's behavior, EPOS will elicit his current context in an unobtrusive way and enable context-aware knowledge services.

Chapter 3 details the modelling of the user's workspace and the context model.

1.4 Knowledge workers in collaborative environments build up valuable search expertise which is not accessible

Supporting information search is still an important topic: According to a DFKI study, "already more than 50% of all office workers spend 10-30% of their time in information search". Consultants frequently request their colleagues' expertise for obtaining appropriate information.

Traditional information retrieval technology does not support the reuse of search expertise. Even the same user issues similar queries several times (either for verification or to find updates). Other users can not make use of successful former searches (queries), these are not transferred to similar new search situations.

Search expertise is not only reflected in one *good query*, but also in longer search paths. The DFKI study further shows, that users seldom tend to use more than 2 keywords in the first step.

EPOS pursues *collaborative information retrieval (CIR)* to capture and utilize expertise that is latent in search processes.

The research that was done in EPOS in this area is presented in chapter 4.

1.5 Information presentations are most times hand-tailored to just one specific visualization goal

Consultants are very familiar with their structures on the workplace. It is easier for them to see information (e.g., query results) if it is related to their own structures. Information objects may be of very different form ranging from text documents to process models.

Up to now visualizations are hand tailored to specific problems. Visualization metaphors are hard-coded for the application.

An adequate visualization for a consultant adapts to user's visualization needs in different situations, considers his context as well as his intention, and provides a customizable implementation of a number of visualization metaphors.

EPOS provides a flexible, context-driven configuration system for problem-specific visualizations.

Chapter 5 details the ontology of visualization objects and introduces the architecture of the visualization framework.

Chapter 2

Leveraging Individual and Shared Ontologies From Native Structures

2.1 Analysis of native structures

EPOS aims at a better balance of the knowledge worker's individual needs and the company's more global knowledge management goals by connecting today's personal workspaces with an Organizational Memory Information System. The personal workspace with its *native structures* like file- and mail-folder hierarchies reflects the worker's personal view of his or her information space. The underlying conceptualizations are therefore a valuable aid not only to guide the worker's information management tasks like storage and retrieval, but also to the internalization and, ultimately, utilization of new information. Furthermore, due to their continuous development by the knowledge workers, the *personal* structures provide an excellent input for the acquisition of *organizational* knowledge. However, today's native structures also have some serious drawbacks:

- They are often built ad hoc, which means they only reflect a snapshot of the worker's view that may not be effective for future situations.
- They lack formal semantics. Therefore, they are hard to exploit by automatic information services.
- There are neither defined processes nor advanced means to make the knowledge contained in native structures available to other workers or to organizational KM.

Figures 2.1, 2.2, and 2.3 exemplify some typical modelling flaws in native structures: Figure 2.1 shows a top-level structure that contains concepts that belong to quite separate domains: *adREAD AP52 Deliverable* for example refers to a project, namely *adRead*, a specific subprocess in the execution of that project (work package 52), and a specific document type that is typical for projects, namely a deliverable. The structure also uses concepts that refer to people (e.g., *Tristan*), to document formats like Powerpoint slides (*PPT-Folien*, specific tasks or to do's (*Hausaufgaben*), and access rights (*public*). Obviously, all these dimensions of description are important for the knowledge worker, but the file system as a native structure doesn't support a clear separation between them. An additional top-level structure could be introduced, but makes browsing cumbersome. Moreover, many file systems do not allow documents to be members of multiple folders, so just one specific view on that document is supported.

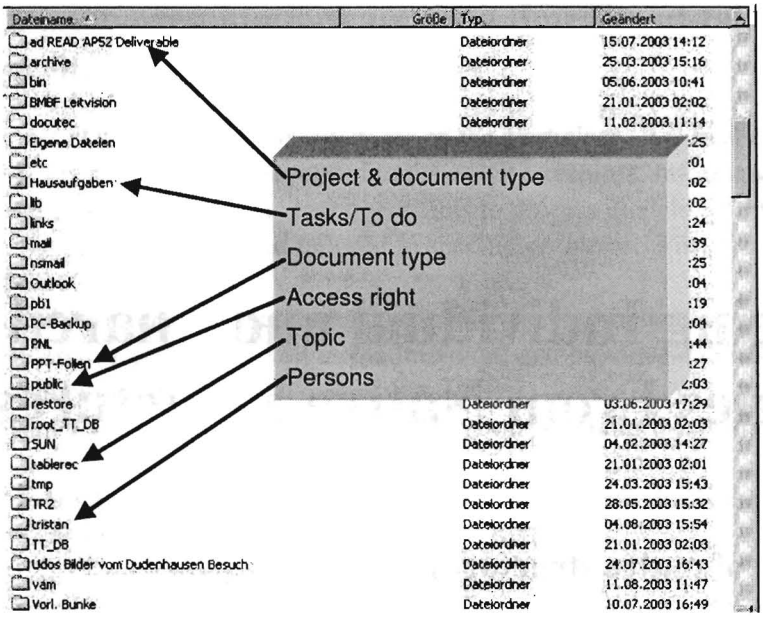

Figure 2.1: Example for a native structures with no clear separation of domains.

Figure 2.2 shows a typical problem that arises from the limited expressiveness of many native structures. The knowledge worker in this example actually uses two types of relations: *Students* or *friends* are `subclass-of` *people*, but the persons (e.g., *Bertin Klein*) are neither `subclass-of` nor `instance-of` *DFKI*, but they are `member-of` (what can be seen as `part-of`) *DFKI*. However, one has no possibility to express this distinction.

The problem in example in Figure 2.3 is not due to the restriction to one relation type in the file system, but due to the fact that no semantics is enforced by the native structure so that "modelling mistakes" are neither indicated nor blocked. Such mistakes arise often, because knowledge workers act with bounded resources and take decision with a local view (originating in the current business process). Typically, there is only limited consistency checking with prior modelling decisions nor are future options anticipated. So, in this example the knowledge worker used a time concept (*year 2002*) as top-level distinction and *projects* and *companies* on the second level. Then, with more and more documents in 2003, have introduced the year 2003 on top-level and copy the substructures from 2002 to 2003; however, this is a costly operation and he just introduced the category *2003* as substructure of *EnterpriseX*, probably due to the fact that he received many documents from this company in 2003. Supposedly, the resulting structure served his current needs, but is neither compatible with a `subclass-of` nor a `part-of` semantics of the (unnamed) relation.

EPOS' concept of a *Personal Information Model* (PIM) which provides a formally sound model of the knowledge captured in the native structures is intended to alleviate the problems that arise from flaws showed in this examples and further deficiencies (like the lack of uniqueness of concepts etc.). In the next section, we show how such a PIM will be embedded in EPOS' general architecture.

Figure 2.2: Example for mixing `is-a` and `part-of` relations in native structures.

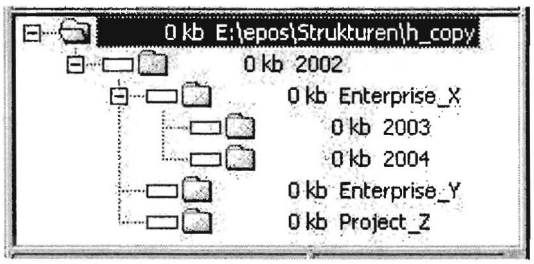

Figure 2.3: Example for violating the `is-a` semantics in a native structure.

2.2 The concept of a Personal Information Model

The Personal Information Model as envisioned in EPOS is driven by four requirements:

- Sound formal basis: The PIM must support various knowledge services, among them logics-based services (e.g., ontology-based information retrieval). Therefore, the PIM must employ an expressive representation language and has to wipe out the contradictions and redundancies of the native structures.
- Bridge between individual and organizational Knowledge Management: The PIM has to incorporate global ontologies, but also has to reflect the changes and updates of native structures. The PIM itself should be a source of input for OM-wide ontologies.
- Maintenance: Adequate means have to be provided that assist the user with stepwise formalization of native structures and inspection of the PIM.

Figure 2.4 shows how such a Personal Information Model is embedded in the EPOS information landscape and what basic functionality has to be provided to link the PIM to the native structures as well as to the envisioned knowledge services. In the first year of the project, we concentrated on two aspects: i) Leveraging and integrating several native structures into one PIM. ii) Generating evidence for mapping and matching PIMs.

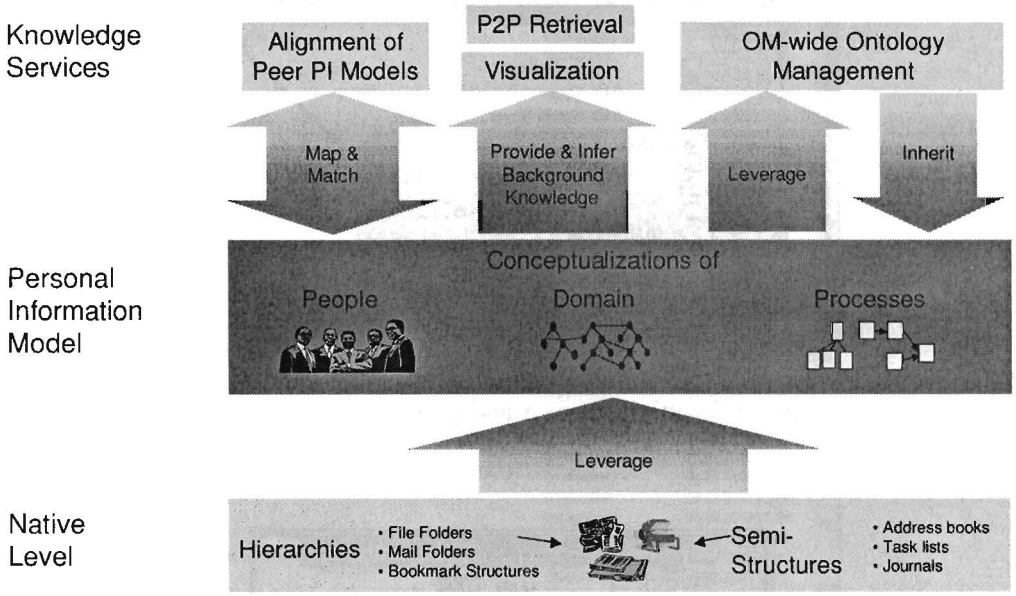

Figure 2.4: The Personal Information Model as *semantic middleware* between native structures and knowledge services.

Together with Brainbot technologies AG – with core competencies in information retrieval and information management – we developed the proFiler, a system which allows for multicriterial classification of documents and provides functionality such as boolean search and document similarity evaluation.

proFiler is the first step in building a PIM as envisioned in EPOS because it allows to import native structures such as email folders, bookmarks, and file directories together with included documents. The structures are shown as trees (usually interpreted as a *is-a* hierarchy[1]). The nodes (interpreted as concepts) get their meaning by a term-similarity vector generated by assigned documents. Then the user is able to construct his personal information space by creating new structures, assigning documents to concepts, and making relations between concepts (a concept can have multiple parents).

Figure 2.5 shows the user interface of the proFiler application. The interface combines an file explorer-like look-and-feel with a search engine interface. On the left hand side, various views on the user's documents are shown, e.g., a *project* view, a *topic* taxonomy, a *document type* description, and a *people*-centered view. These categorization schemas are either handcrafted or imported from various native structures (e.g., mail tools or file folders) and afterwards adjusted. The right hand side shows the user's documents that are classified according to the categorization schemas. The assignments of documents to categories may either be done manually or automatically, based on classifiers that are learned from the explicit assignments.

With respect to the desired *mapping and leveraging functionalities* of PIMs, we designed and implemented the basic evidence generation mechanisms based on the *terms* in the model, its *topology*, and its *grounding with annotated documents*. The prototypical implementation is built on top of Protégé as for the representational framework and user interface, and proFiler for

[1]These structures can be im- and exported as RDF schemas.

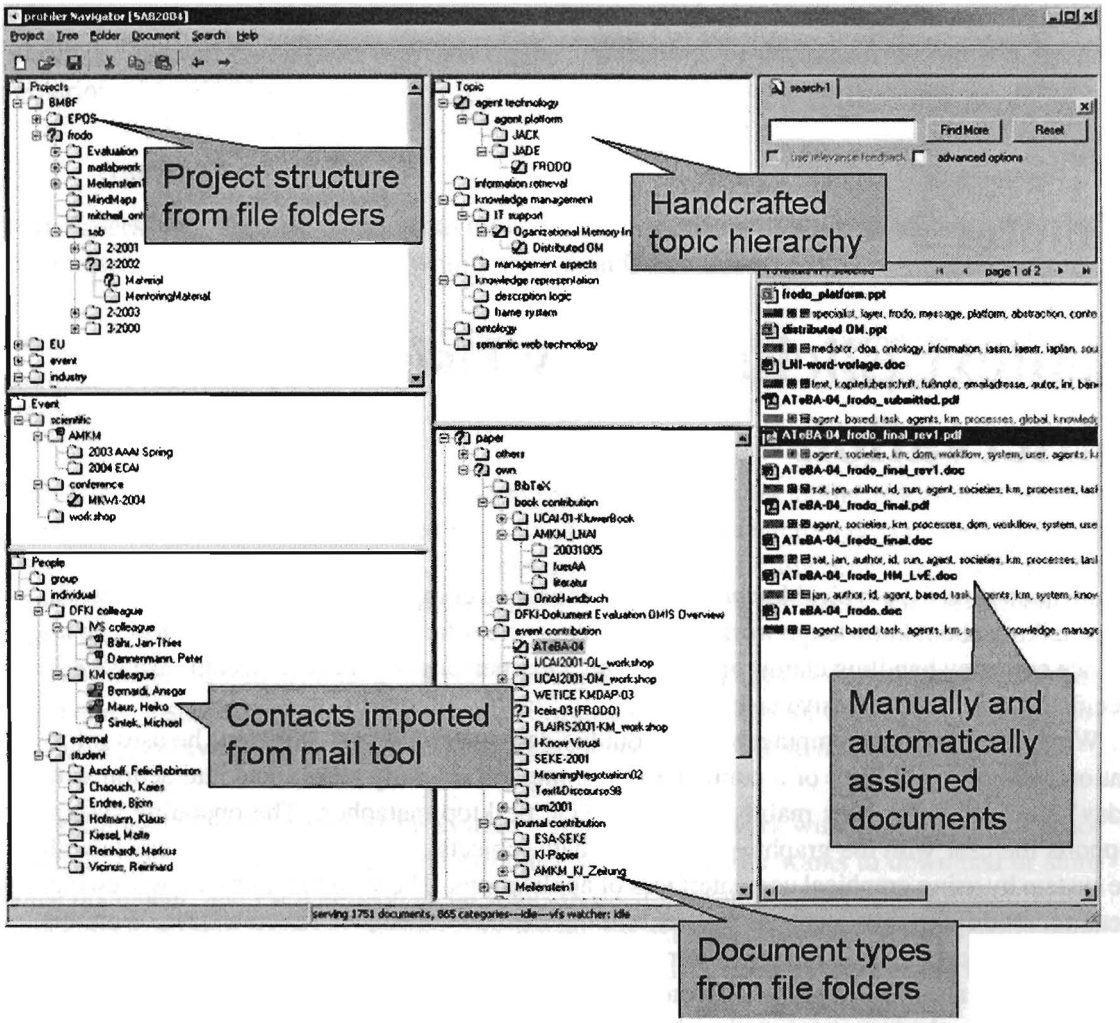

Figure 2.5: The Personal Information Model as *semantic middleware* between native structures and knowledge services.

computing document-based concept similarities. (A detailed desription of evidence generation is subject to a diploma thesis that is in progress).

Chapter 3

Context Elicitation by User Observation

We envision to assist a typical knowledge worker using a computer to carry out his daily work. When talking about knowledge workers, we think of scientific researchers, consultants in an insurance company handling claims and such, generally persons who have to accomplish weakly specified, knowledge intensive tasks.

What does "using a computer to carry out his daily work" mean? It means, he uses information technology in form of a computer to accomplish his daily tasks. The interaction with today's computers happens mainly via a graphical desktop metaphor. The operating system supports the user with the graphical display of data (objects), and the user then interacts with the system by using graphical user interfaces of applications. These applications allow viewing, creating, modifying documents as well as sending them to colleagues via email, etc. With this document work, including retrieval, storage and categorization of documents, the knowledge worker tries to accomplish his tasks grounded in the real world.

EPOS research aims at supporting the user's work by context-sensitive assistance. For this purpose, we will elaborate techniques to elicit the user's context by observing his behavior (i. e. his computer usage) and by the information he is handling. The user's context is estimated as far as possible by the system in order to support the user as unobtrusively as possible.

Structure of this chapter: The following section 3.1 will specify a designated subset of the user's workspace relevant for the "EPOS user". When talking about this workspace subset, we talk about the subset of *applications*, *actions* and *objects* of a user's workspace, which is modeled and captured by a "workspace model". Section 3.2 gives detailed information about the techniques and architecture of the user observation in EPOS. Section 3.3 presents the formal specification of our context model. Section 3.4, gives an overview on applications of the elicited context. The chapter closes with section 3.5, which discusses the next research steps to be made.

3.1 The User's Knowledge Workspace

In EPOS we will not observe all of the user's interactions with the real world, but rather focus on his interaction with the *relevant* applications running on his PC. As already denoted,

document work is the main part of the user's job, hence we focus on document handling applications. We will give a short overview on the applications we declared to be relevant to our scenario:

3.1.1 Applications

Text Processor The user's main activity will be the operation of a text processor. We think of something like MS Word or the Word in the Open Office Suite.

Email Client Sending and receiving and replying to emails is today's typical means of communication and collaboration with colleagues. Email is also used to send important documents around.

Web Browser The Web Browser is the most used interface for doing research. Even the enterprises' intranet is often searched with a web browser. Research, indexing and clustering done by search engines are used to find relevant information on the user's task. Hence, the browsing behavior, that is the navigation of the interlinked pages can give some clues about the user's *goal*.

Additionally a web browser allows for printing out visited pages as well as for downloading documents hosted somewhere in the world. Printing or downloading documents (web pages) provide a good evidence for the hypothesis, that the user is willing to *read* these documents.

File Explorer / File Manager / File System Unfortunately, work with a computer still is not possible without something like a file explorer. Whenever a user wants to download or store some document, and even as soon as he wants to open a document, the user still has to cope with a file exploring interface. Saving a document, for example, still requires the specification of a targeting folder together with a "filename". Analogously, opening a document requires the same parameters.

In order to be able to cope with the typically vast amount of files (documents) a user stores today, he manages these using more or less elaborated folder structures and maybe his individual file naming convention (see chapter 2). Often, these folder structures represent his real-world categorization and understanding. Hence, these structures are an interesting observation object. It can also make sense to change the name of a folder or a file, namely, if the user's understanding of the world changes or if the file is used in a different context or for a different purpose from now on.

The operating system / desktop interface As already denoted, the interaction with today's computers happens mainly via a graphical desktop metaphor. The operating system supports the user with the graphical display of data (objects), and the user then interacts with the system by using graphical user interfaces of applications. However, not only starting / closing applications and viewing / editing documents is needed to realize a desktop-like metaphor. Further interactions like application/view switching or view minimization ("put that out of my sight") are very important and will be observed as evidences for context switches/shifts.

3.1.2 Objects

Documents Documents carry mainly text content, maybe also hyperlinks like in an HTML document (a web page). They are often organized using folders.

Emails Emails are like documents. The difference is merely, that emails are of an other information type. They are often organized using folders, too.

Files When a knowledge worker handles a file, it will be a *document* in most of the cases. In other words: documents are the most interesting files for EPOS.

Bookmarks In order to remember interesting locations, resources and documents in the world, users can store links to these in form of bookmarks. Also bookmarks are often organized using folders.

Folders Folders provide a hierarchical structure for storing and classifying documents, emails, bookmarks, etc..

Address Book Entries The user's address book gives interesting information about his direct email partners. Receivers as well as senders of email are held in there together with further information like their homepage.

3.1.3 Operations provided by the applications

For each application we specify a list of operations we pursue to observe in EPOS. For less obvious cases, some additional information regarding the (different alternatives of the) meaning of an operation is supplemented, too.

Text Processor

- View a document (load and display a document).
- Cursor movement and scrolling (page up/down)
- Modify a document (enter or change characters of the document)
- *Changing the order of some passages would be a nice operation, but instead we have to use the following only:*
- Cut, copy, paste passages of text
- Select passages
- Print out document
- Save—*is this operation really interesting? The following is much more interesting:*
- "Save as"—this can have one of the following reasons:
  - *reuse:* use an old or other document as "input" to create something similar
  - *versioning:* create a new/newer/better version of a document
  - *branching:* starting from the old document to go *partly* into a different direction

Email Client

- Receive emails—a passive, but yet interesting "operation"
- View an email (for reading, replying, "edit as new", ...)
- Compose a (new) email
- Edit an old email as new
- Send later / save composed email as "draft"—this email can (will!) be refined later on
- (Re-)open an email draft (load recently stored email into composer)
- Send an email
- Reply to an email
- Forward an email
- Address Book Handling (this is nearly a separate application)
  - Add an address book entry
  - Change, delete address book entry
  - Search for an address book (search for a person)
- Move an email to a specific folder—This is a classification action. Mind the difference between moving an email from the inbox to a folder (*classification*) vs. moving an email from one folder to another (*re-classification*)
- Delete an email—There is a difference between deleting emails in the inbox (this means: *unwanted/uninteresting email*) or emails in other folders (*not needed/interesting any more*)
- Print out, save, export email
- Create, delete folder
- move, copy folder
- Browse folder (searching for a mail)

Web Browser

- Open web page, navigate directly to (entering) a specific URL
- Go (navigate) to a bookmark (remember: bookmarks are organized in folder structures)
- Navigate back, forward (using the navigation history)
- Click at a link on the currently visible web page
- Press a button on the web page
- Fill in some text into forms on the web page
- Search for a string or a link on the web page
- Scroll up/down (lines/pages)
- Add a bookmark for the current web page
- Print out the currently visible document
- Download a file, save the target of a link

- Save web page (export format could be interesting)
- Send web page (remarkably interesting case: the user sends it to himself)
- *Handling of different tabs may give clues about similar context, goal or topic.*
- Bookmark folders: create, delete, move, copy change name
- Bookmarks: create, delete, move, copy, change name/URL

File Explorer / Manager

- View available folders (top-level folders, favorites, ...)
- View contents of a folder
- Browse folder structure (browse/switch to folder)
- Create, delete a folder
- Change a folder's name or other properties like the owner or the time stamp
- Move a folder to a different location (different super folder)
- Copy a folder (including its contents)
- Delete a file
- Change a file's name or other attributes like the owner or the time stamp
- Open, view a file (call an appropriate application for viewing this file; note: the open/view operation happens at that application and not here, really)
- Move a file (to a different folder)—meaning: "I want that file at a different place"
- Copy a file (to a different folder)—meaning: "I want that file *also* at a different place"
- Copy file to the same folder, but with a different filename
- Add links to files, especially: add links to files and put them onto the desktop
- Remove, change properties of such links
- Search for a file or folder (by name, pattern, time, etc.)

File System Interface – "Save As" and "Open" dialog

- Save a document to a specific folder and with a specific filename (the chosen folder as well as the contained files are visible to assure, the chosen folder is the right one)
- Open a document (including selecting a document) for viewing / editing / whatever (parent folder plus sibling files may be visible during the selection process)

The operating system / desktop interface

- Start, close an application
- Switch to another application
- Open, close a view
- Switch to another view (changes focus and maybe visibility)
- Minimize, restore an application or view

- Undo the last action—this could be quite difficult to observe as it forces the observation of *absolutely every* action a user can take.

3.1.4 Four Levels of User Activity

Observation and analysis of the potential system operations, the user can trigger, will not deliver a sophisticated understanding of the user's real intentions and goals. As the user is thinking in different levels of granularity and semantics, we have to keep this in mind while observing and analysing his behavior.

Figure 3.1 depicts four different levels of user activity: The first level, called **Workspace Level**, represents the operating system and the applications that provide access to files, objects and information structures. Observation at this level results in workspace events such as various mouse clicks, entering of some text, or starting and handling applications.

The user's momentary intentions, expressed by his *user actions*, are independent of the currently used workspace. The **User Action Level**, thus, contains such user actions as *create new text document* or *revise document*, rather than atomic mouse-clicks or actions like *start text editor* or *activate File-new menu*.

While the user tries to solve his problems with the OS and some applications, he always has some higher medium-term goal in mind such as *write down results* or *write project proposal*. Those user goals are captured in the **Task Concept Level** and are represented by *task concepts* which are concepts in an ontology about such user goals. EPOS will elicit the user's goal(s) from a sequence of the user actions needed to be carried out to achieve this goal. One research topic in EPOS, "context elicitation by user observation", focuses on the investigation and realization of mechanisms for such a goal recognition / elicitation.

Knowing about the user's goal(s), EPOS can provide goal specific support to the user, such as relevant documents / information. Furthermore, if we know about goal specific information-needs, we can fill the user's information gap by presenting him respective documents.

And, last but not least, the **Process Level** connects to the organizational structures and processes which might be explicitly modeled (e. g., with a business process modeling tool) and/or enacted by a workflow management system (WfMS). If there is such a WfMS available, we can connect / assign the user to running workflows. Workflows can be semantically described using the same set of task concepts [Schwarz, 2003] which have been elicited from the user's behavior. So, we can use the task concepts to identify the workflow tasks the user is (or seems to be) currently working on. That way we can use and offer workflow knowledge indirectly—i. e., without *direct* interaction with a WfMS.

3.1.5 A Formal Model for the User's Workspace

The workspace model, that is the designated subset of applications, actions and objects an "EPOS user" uses has been modeled and specified using the Protégé tool and resides in form of an RDF/S model.

This workspace model will be used by the context elicitation process as it contains information about relationships of action and object classes. So, for instance writing an email is a messaging action and has a relationship to the addressees and to an email object.

The workspace model is generic, which means, that the real workspace used by a human is just an instance of this workspace model. For example, every typical workspace allows for

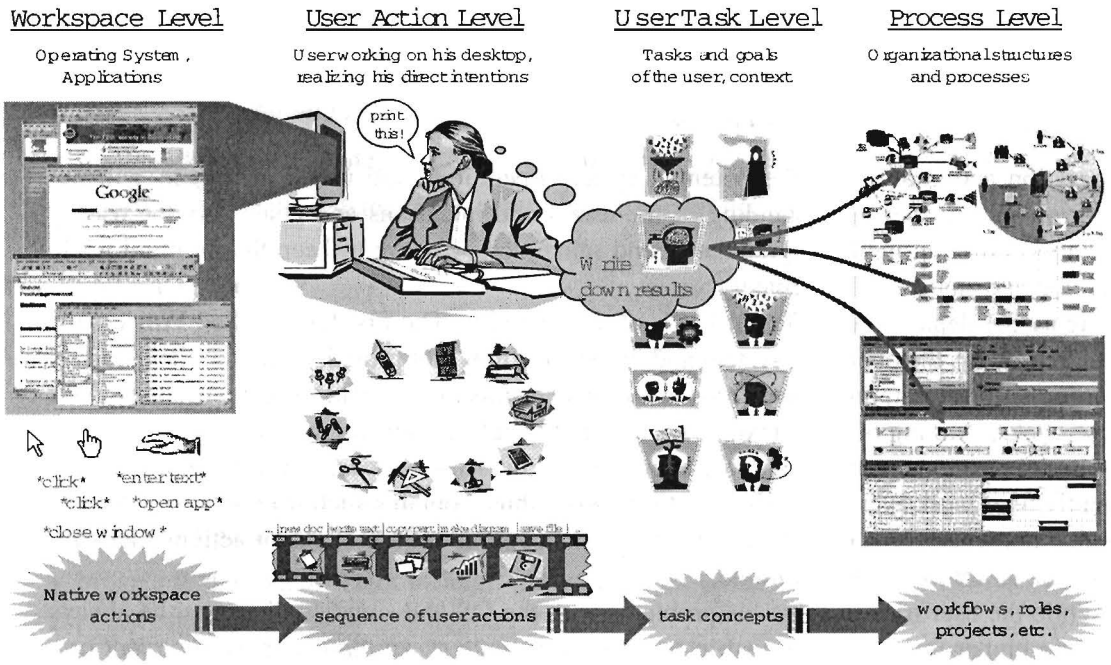

Figure 3.1: The user observation architecture handling different abstractions of user interactions.

writing an email. On some workspaces "MS Outlook" is used for this concern, however, on the "EPOS workspace" this is done via the Mail Client "Mozilla Thunderbird".

3.2 User Observation

3.2.1 From User Observation to Context Elicitation

We specified a pipelined architecture to elicit the user's current context (see figure 3.2). We start with the observation of native workspace operations, which already present some (however low-level) context, and gradually enrich this context using higher-level context elicitation components later in the elicitation pipeline. At all levels of the context elicitation process, we can use the contextual information below (lower-level context).

Due to the *automatical* process of the context elicitation, the context is, of course, just an estimation of the real user context. This holds especially for the higher-level aspects of the context like the user's goals. This problem will be tackled via two ways: (1) The user may help the context elicitation by sparse and good balanced interaction. (2) Observed user actions are interpreted as evidences for context hypotheses. A continuous sequence of observations incrementally narrows the set of all possible interpretations. Hence, after a specific observation time the estimated context gets more and more certain.

3.2.2 User Observation Techniques

As already pointed out in section 3.1.1 we want to observe mainly four applications. User operation of these applications will be observed in the following way:

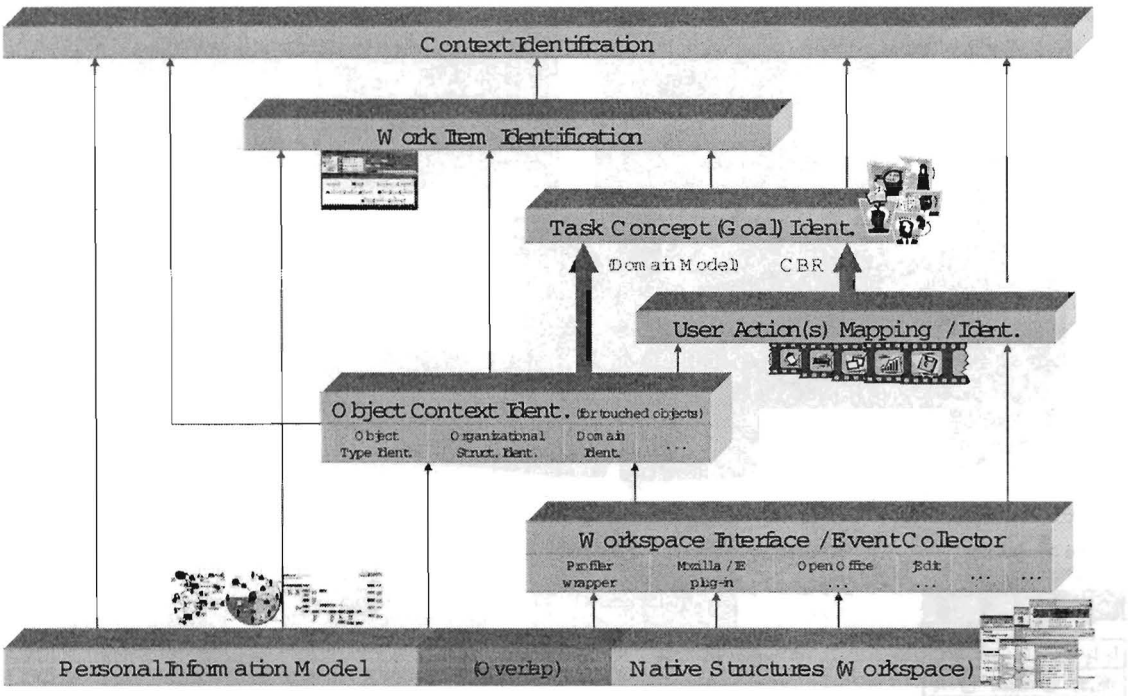

Figure 3.2: The components for the context elicitation process realize a pipelined architecture.

Text Processor: Open Office Open Office (`www.openoffice.org`) is an open-source office suite. Since version 1.1 Open Office offers a Java interface. We use this interface to implement a Java-based observation of the text processing component of Open Office.

Email Client, Web Client: Mozilla Thunderbird, Firebird Mozilla Thunderbird and Firebird are realized using the Mozilla architecture. Hence the graphical user interface is specified using XUL (XML User Interface Language), which is a markup language for describing user interfaces. XUL only tackles the graphical part of a user interface. The interface behavior is implemented in Java Script. We added event observation routines to the Java Script code of Thunderbird and Firebird. For every user operation like e. g. *compose mail* there is a Java Script function. In order to get informed about the Thunderbird / Firebird activity, we inserted own code there sending off some XML-RPC call to our EPOS user observation module.

Note: This observation method is not very smart, because you will not be able to upgrade to a newer version of Thunderbird or Firebird without some migration work, but this rapid prototyping technique is sufficient for our concern. We are not interested in sophisticated application observation methods. Instead we focus on utilizing the user observation to elicit the user's context.

File Explorer: proFiler The proFiler system (see chapter 2) developed by BrainBot allows for a multi-criterial document (files) storage and retrieval. Moreover the system assists the user by proposing potential categorizations (using text content similarity). This is especially useful for newly created or downloaded documents. Sooner or later the proFiler can substitute

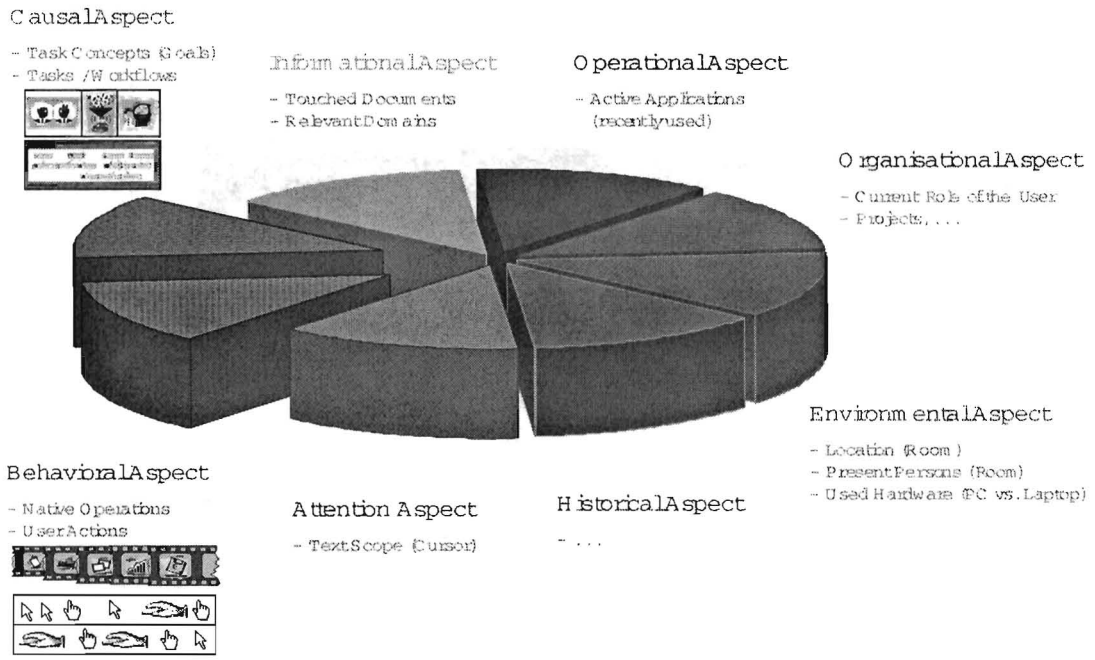

Figure 3.3: The user model comprises different contextual aspects.

the normal file explorer. Then, the user can rely solely on the proFiler for his document (file) management and search.

As a result of our strong collaboration and fruitful discussion, BrainBot provides an XML-RPC interface to the proFiler system. We will use this XML-RPC interface to realize an observation of the user interacting with the proFiler system.

3.3 Context Model

Representing the user context is no trivial task. The user context can not be represented by a single item of information. Complex structures of data without any semantics or relationships do not solve the problem either. The point is, that context is not something real and graspable. One can only describe some *aspects* of the user's current context. Representing his whole context is impossible as we will not be able to capture all the relevant evidences within the whole PC environment.

Instead, what we have to do is, first of all think about the precise aspects of the user's context we want to recognize and utilize, and then think about the representation of contextual information items within each aspect.

As we aim at supporting the user by providing him documents potentially relevant to his *informational aspect*, we consequently have such an aspect in the EPOS user context model; Assisting the EPOS user with know-how expertise implies including a *causal aspect*; etc.. Figure 3.3 shows all aspects of the EPOS user context model and some information about the contextual information items we envision to represent.

Analysis of typical knowledge work has led to decisions about interesting contextual as-

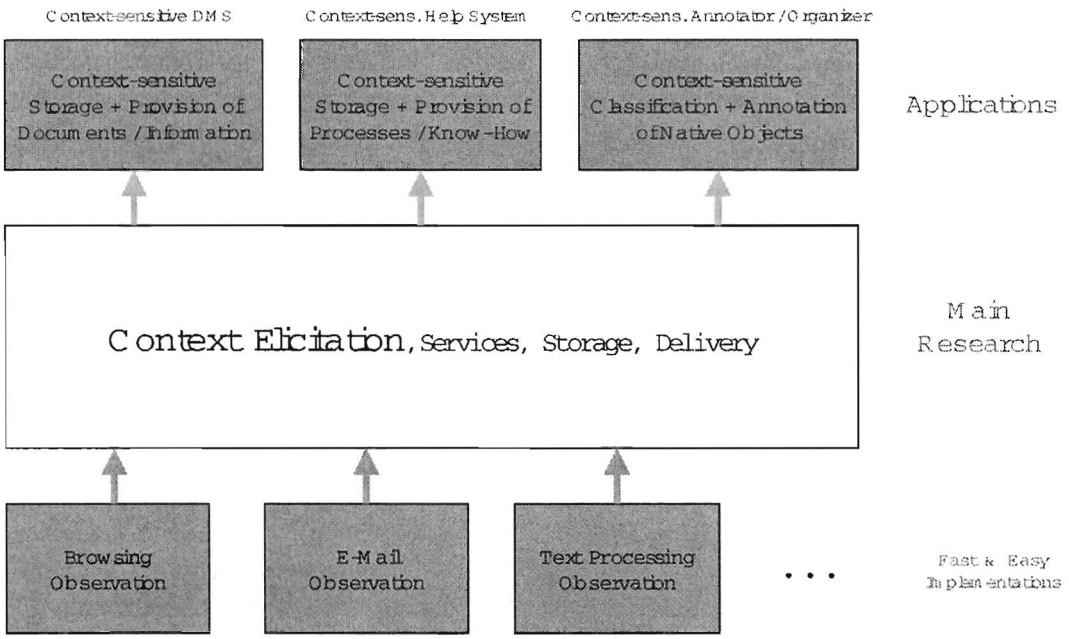

Figure 3.4: Generation and usage of the context elicitation.

pects and items. The EPOS user context model has been modeled using the Protégé tool. The model is formally available in RDF/S and can as such be used for further inferences at the meta-level. Storage, delivery, attachment, etc. of the user's context will always be an instance of this context model.

It is important to point out, that such a context instance contains as many formal items (respectively references to these formal items) as possible. Ideally these will be elements of the "Personal Information Model" or even elements of the organizational model, but also the native operations used in the behavioral aspect of the user context will not be simple literals. Furthermore they will be exactly the RDF classes or instances from the formal workspace model (see section 3.1.5).

3.4 Applications of User Context

Figure 3.4 gives an overview on the generation and usage of context elicitation. Our main research focuses on the context elicitation process. Surely, some user observation has to be done as it feeds the context elicitation, but the precise techniques involved here are out of our focus. Hence, we will not investigate on this topic. The application layer, however, are a second topic of interest as it shows the utility of our envisioned context elicitation, storage and delivery.

A first prototypical application is `EPOSNotes`, a tiny but effective notes tool. The idea behind `EPOSNotes` is to annotate short text notes with automatically generated user context. The context is stored together with the note as its "creation context". Hence, the user can concentrate on the note text only, but nevertheless can find a certain note by searching for context-specific details. For instance, the user is able to search for notes, which have been cre-

ated after having visited some web page or after having searched the web for some keywords. The annotation of the notes comes at no cost for the user, and searching for context-specific details, the user remembers, is much more natural (intuitive) than searching for keywords or abstract domain concepts only. Of course, full-text search for words in the notes will still be applied.

Beside this simple notes application, the list of potential applications for (semi-)automatical context elicitation is quite extensive. A few examples for such applications can be found in the following, not necessarily complete, list:

- A context-sensitive document management system providing a context-sensitive storage, search and delivery of documents. EPOS Notes is quasi an instance of such an application.
- A context-sensitive help system providing context-sensitive know-how such as procedural expertise.
- A context-sensitive annotator and organizer of objects (documents). In cooperation with the user, the EPOS system allows for a (semi-)automatic annotation / classification of objects.

3.5 Outlook

By now we've only started context elicitation with capturing the user's native operations. Our very next steps will be enriching the context with higher-level contextual information. For this purpose we will realize the respective context elicitation components as depicted in figure 3.2. To be more precise, we will start to identify the user's intentions ("user actions"), which means, we will have to realize the "user action mapping / identification" component. For this purpose we will surely have to realize the "object context identification" component, too. The latter component will investigate the objects (documents) touched by the user and deliver meta-information like related topics (domains) of object. We think, that such meta-information (of touched objects) is as important as user actions and goals for the context elicitation.

Having higher-level context at hand, we will show the utility of rich context by implementing more sophisticated user assistance applications. As an example, a best-practice (know-how) assistant will support the user as soon as the user's estimated context provides highly probable information about the users potential goals: The EPOS system will propose next steps to solve his problems, as well as, telling him the colleague with the best expertise on that problem (goal).

As soon as we start enriching the context with higher-level information, thinks get more and more complicated. In other words, measuring the quality of the context elicitation is crucial, because otherwise we end up, evaluating subjectively only. So, one of our next steps will be discussions and concepts about more objective evaluation methodologies. Of course, the perfect output would be a "ground truth" for context elicitation by user observation, but it is doubtful if this destination can be reached, really.

Chapter 4

Precise Satisfaction of Information Needs

4.1 Analysis of requirements and review of existing approaches for Collaborative and Peer-to-Peer Information Retrieval

4.1.1 Collaborative Information Retrieval

Introduction

Collaborative Information Retrieval (CIR) is a new research direction in the research area of Information Retrieval. This research was started at German Research Center for Artificial Intelligence (DFKI) in the 'Adaptive Read' project in the years 2000 to 2002, which was supported by the German Federal Ministry of Education and Research (bmb+f) under Grant 01 IN 902 B8.

We call our approach Collaborative Information Retrieval (CIR), learning to improve retrieval effectiveness from the interaction of different users with the retrieval engine. CIR on top of an IR system uses all the methodologies that have been developed in this research field. Moreover, CIR is a methodology where an IR system makes full use of all the additional information available in the system, especially

- the information from previous search processes, i.e. individual queries and complete search processes
- the relevance information gathered during previous search processes, independent of the method used to obtain this relevance information i.e. explicitly by user relevance feedback or implicitly by unobtrusively detected relevance information.

The collaborative aspect here differs from other collaborative processes. We do not assume that different users from a working team or a specific community collaborate loosely or tightly through some information exchange or workflow processes. Instead we assume that users can benefit from search processes carried out at former times by other users (although those users may not know about the other users and their search processes) as long as the relevance information gathered from these previous users has some significant meaning.

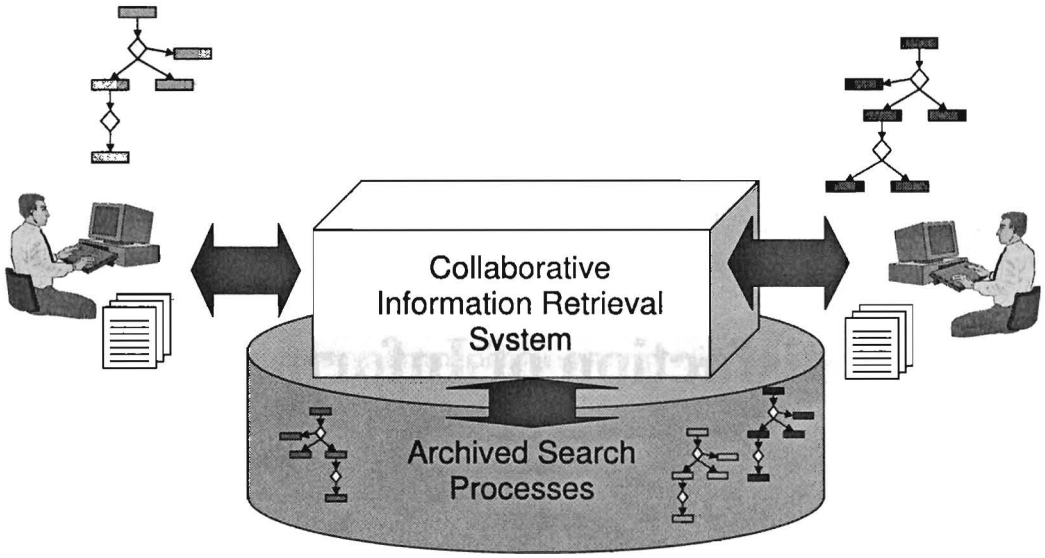

Figure 4.1: Scenario of Collaborative Information Retrieval

Figure 4.1 illustrates the general scenario of CIR. An information retrieval system is typically used by many users. A typical search in a retrieval system consists of several query formulations. Often, the answer documents to the first query do not directly satisfy the user so that he has to reformulate his query taking into consideration the answer documents found. Such refinement may consist of specializations as well as generalizations of previous queries. In general, satisfying an information need means going through a search process with many decisions on query reformulations. Hence gathering information for fulfilling the information need of a user is an expensive operation in terms of time required and resources used. The same expensive operation has to be carried out if another user has the same information need and thus initiates the same or a similar search process.

The idea of CIR is to store these search processes as well as the ratings of documents returned by the system (if available) in an archive. Subsequent users with similar interests and queries should then benefit from knowledge automatically acquired by the CIR system based on the stored search processes. This should result in shorter search processes and better retrieval quality for subsequent users if the following basic assumptions can be fulfilled by an CIR system:

- relevance judgements for retrieved documents can be derived from users' actions
- previous queries by some users will be useful to improve new queries for other users

Subject to these assumptions we expect that collaborative searches will improve overall retrieval quality for all users.

Research Areas

Research in the areas "personalization" and "context" integrates modelling and representation of the personalization aspect of the user and the context information, and integrates this in-

formation into the IR processes. Because queries can be vague, it might be possible to use knowledge about the scope the user is working in to influence the query processing for a more detailed specification of the information need and achieve better retrieval results. Figure 4.2 shows the overlapping research areas.

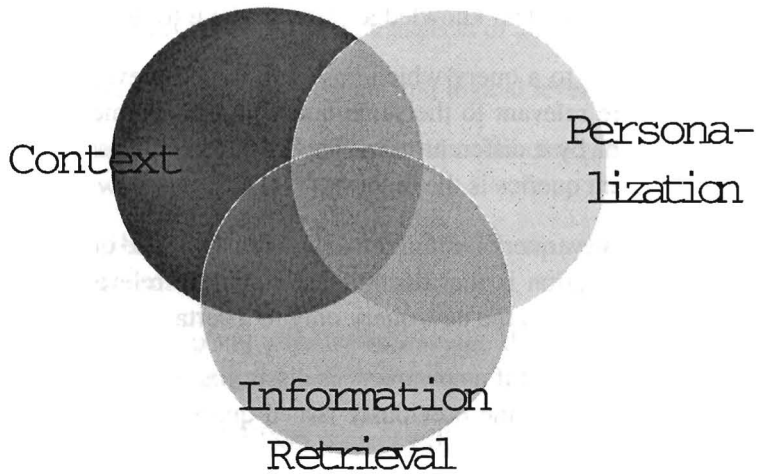

Figure 4.2: Overlap of Research Areas Information Retrieval, Personalization and Context

Let us state an example to show the different preference relations users may have. A physician, a chemist and a lawyer may query an IR system for information about the medicament 'Lipobay' or its American name 'Baycol'. While the physician may be interested in medication, indication and contra-indication, the chemist may by interested in chemical structure and undergoing reactions of the active ingredient; the lawyer may be interested in legal cases, lawsuits, court decisions and compensations. It is clear that each of these users has his or her own personal preferences as to which documents an IR system presents in response to the query. This is denoted by the term 'personalization' and should be considered in the development of new IR systems in the future.

These preferences may also be influenced by the 'context' the user is working in and it is clearly possible that these preferences may change somehow over time. Some of the aspects of the user's context (according to [Henrich, 2002]) are: which tasks the user is busy with at the time of the query, which documents have been viewed within the last few minutes, which document is currently being processed by the user. As an example, a user may query an IR system for information about 'the speed of a jaguar'. If we knew that the user was currently browsing the web-sites of car manufacturers we could assume that he or she is interested in the car 'jaguar' and eliminate documents that are referring to the 'jaguar' animal.

Motivation

How users can improve the original query formulation by means of relevance feedback is an ongoing research activity in IR [Baeza-Yates and Ribeiro-Neto, 1999].

[Manning and Schütze, 1999]. In our approach we use global relevance feedback which has been learned from previous queries instead of local relevance feedback which is produced during execution of an individual query.

The motivation for our query expansion methods is straightforward, especially in an environment where document collections are static, and where we avoid further complexity by ignoring personal preferences and context knowledge (refer to the introduction in section 4.1.1):

- If documents are relevant to a query which has been issued previously by a user, then the same documents are relevant to the same query at a later time when that query is re-issued by the same or by a different user. This is the trivial case, where similarities between the two different queries is the highest.
- In the non-trivial case a new query is similar to a previously issued query only to a certain degree. Then our assumption is that documents which are relevant to the previously issued query will be relevant to the new query only to a certain degree.

It does not necessarily follow that if a new query is dissimilar to a previously issued query, the documents which are relevant to the previously issued query are not relevant to the new query.

Current Results in CIR

Research results from this area fall into three categories:

- methods that are based on the co-occurrence of terms in a set of queries and the occurrence of these terms in their relevant documents ([Klink *et al.*, 2002a], [Klink *et al.*, 2002b], and [Klink, 2004]).
- methods that are based on the similarities of queries, using query expansion terms from relevant documents of the most similar queries ([Hust *et al.*, 2002a], [Hust *et al.*, 2002b] and [Hust *et al.*, 2003])
- methods that are based on the similarities of queries, reweighting document and/or query terms based on the relevant documents of the most similar queries ([Hust *et al.*, 2004], [Hust, 2004])

4.1.2 Peer-to-Peer Information Retrieval

Traditional IR systems consist of a centralized server which is responsible for managing and administrating the centralized document collection. It has the responsibility for most of the tasks to be carried out, for example, the indexing of the documents, the searching and the ranking process.

In some cases such an architecture may not be preferable. A centralized server represents a single point of failure, which means that the complete IR system is unavailable, if the server is down. Additionally, a centralized server has a limited scalability, which means that the workload of the server is limited, and hardware upgrading cost may be disproportional

compared to the performance.

Peer-to-peer networks work without centralized components. Every peer works as a server and as a client. In its role as a server, a peer offers its services to the whole network, and in its role as a client, the peer queries the network for data and information.

In the scientific community, typical research aspects of peer-to-peer networks are:

- how to organize the network on a physical and logical level
- how to re-organize the network if the number of connected peers changes (peers entering or leaving the network)
- how to distribute information in the network, such that the information is redundant enough to overcome some hard or soft failures
- how to detect malicious peers who try to damage the network or trv to deceive other peers, and how to eliminate these malicious peers

The peer-to-peer scenario in the EPOS project will focus on the information retrieval aspect only. In this scenario, we expect that

- document collections are distributed over the network, i.e., an individual peer only has a small amount of knowledge of the complete network
- peers qualify for different levels of expertise for different knowledge areas in the network, i.e., there are specialists, experts, newcomers, etc. for a certain knowledge domain and the level of expertise of a peer varies over the different knowledge areas

Research in the peer-to-peer scenario in the EPOS project will focus on

- how to determine the quality of information that is delivered as a response to a query from other peers
- how to estimate the qualification of a peer to answer queries
- how to build up a rank of estimates of the qualification of other peers
- how to find the best neighbors in a network fitting best to the peers' information need (seen as a short-term information need, e.g., an ad-hoc query, and as a long-term information need, e.g. an interest)

4.2 Information Need of a Knowledge Worker

Knowledge workers essentially rely on *relevant* information at the right time. EPOS tries to infer a user's current information need in order to support his work. To reach this goal, EPOS needs a representation of a user's information need as well as means to satisfy it.

This chapter discusses in section 4.3 several notions of information need and with this background states in section 4.4 the requirements as well as the concept for an information need in EPOS.

4.3 Concepts for information need

In this section several relevant notions of information need are introduced.

4.3.1 Information Need in KnowMore

The DFKI project KnowMore (Knowledge Management for Learning Organizations) aims at supporting workflow participants dealing with knowledge-intensive tasks (so-called *kits*). With the help of workflow context, KnowMore retrieves relevant knowledge from an organizational memory (OM).

There, the information need of such a knowledge intensive task is stated with the help of a *kit-description*. With this and the corresponding context from the workflow activity, an information agent accomplishes an extended, ontology-based information retrieval to satisfy the information need by presenting relevant information.

The kit-descriptions extend the conventional definition of a workflow activity with a support specification. It specifies the information need as generic queries together with the responsible information agent which is shown in the following:

```
( name: ask-specialist,
description: "email to specialist for the wanted product",
precondition = {product-name=null}, // ask only if no idea yet;
// product-name is variable of the workflow data flow
agent-spec: "person-competence-agent select ($p-type)",
parameters: {product-type},
from: {enterprise-competence-base},
contributes-to: {product-name, supplier-id} )
```

4.3.2 Information Need in FRODO

The EPOS predecessor-project FRODO[1] (Framework for Distributed Organizational Memories) focussed on supporting weakly-structured processes with the concept of weakly-structured workflows. Such a workflow can - but needs not - follow a predefined workflow model and can be modified during runtime. A knowledge worker can add, delete, and change tasks as well as control and data flow. In order not to overwhelm a knowledge worker with complex queries, FRODO used an easy way of expressing an information need, which is wellknown from search engines: within a task, information needs can be added as boolean terms with elements from domain ontologies or as free keywords. The user is able to additionally weight different elements as some way to express importance. These information needs are then interpreted by an information agent. Further information needs can be inherited from the respective parent task at which the weighting is reduced (e.g. by 50%).

Another reason for this simple way of expressing information needs was to allow the user to pose search requests within a task which then are captured and stored within the task for later reuse.

[1] `http://www.dfki.de/frodo/`

4.3.3 Information Retrieval

In the area of information retrieval, one can distinguish between boolean search and more sophisticated approaches which use weighted terms. However, the user's way to express his information need is still by listing several keywords within a query – possibly with boolean operators.

If information retrieval systems use for example the vectorspace model, each of the terms can be weighted. But this is done transparently to the user, for instance, because of similar queries (see section 4.1.1) or by using a user's profile which gives evidences for the importance of used terms.

Because of this restriction, researchers investigate different ways to overcome the limitations of the query interface of todays ir-systems. Two prominent approaches are presented in the following sections.

WordSieve

WordSieve [Bauer and Leake, 2001] [Turner *et al.*, 2001] use the notion of a task context which is specified by observing the users using a search engine within a internet browser and investigate visited documents (to which they refer as *document access patterns*).

By observing these documents, relevant keywords are identified and a term index is constructed. This term index serves as a kind of current task context for the user. If a major change in the relevant keywords of a user occurs they infer that the task context of a user has also changed, and likewise, the user's information need.

If a previously protocolled task context is recognized, the respective term index is used to improve the current search. With this approach they are close to ideas used in CIR. The evaluation of their approach with TF/IDF showed significant improvements.

DynaCat

Instead of focussing of detailing queries for finding appropriate information within search results, Pratt et al. [Pratt *et al.*, 1999] provide with the DynaCat system, a tool that dynamically categorizes search results into a hierarchical organization by using knowledge of a set of previous queries and a model of the domain terminology. As domain terminology DynaCat uses the terminologies provided by the National Library of Medicine (NLM) and the Unified Medical Language System (UMLS).

In an evaluation, they showed that DynaCat helps users to find answers to questions more quickly and easily than when they use relevance ranking or clustering system because they can can refer to the clustering by the domain terminology.

4.4 Information need in EPOS

EPOS tries to infer a user's current information need in order to support his work. In the given scenario, EPOS has to deal with an information need stemming from the current user goal, influenced by the user's personal view on information as well as influenced by the presence of and interaction with other users, and various relations to models ranging from formal organizational ontologies to the more informal personal information model.

Whereas the predecessor projects FRODO and KnowMore were able to infer the information need from a given workflow activity, in EPOS a workflow is only a source among others. Therefore, EPOS tries to infer the current information need with the help of user context from user observation (see chapter 3) within the user's workspace. As EPOS focuses also interconnected workspaces of users in groups, EPOS also relates the user to his current group and role and with that gets further hints on the information need for specific tasks and goals.

4.4.1 Requirements

A problem to face when designing a model for information need, is the difference between the real information need of the user which he has in mind and the representation he uses for interfacing a search engine.

For example, a consultant wants to find information about current developments in information retrieval for supporting the project he is involved in. Now, a current study on user search behavior states, that users only use 2 keywords in average for a search [AdaptiveRead03, 2003]. Keeping that in mind, appropriate search terms for the above information need are hard to find.

Considering this, it is easy to understand that an information need is that amount of information which is needed by the user to adequately solve his current problem and the query he states is just a – usually not comprehensive – mapping of the information need in mind.

The problematic word is "adequately" which requires to consider different things:

- provide only new information which is unknown to the user
- provide also information previously known to the user but he currently is not aware of
- the user is expert in a topic, he needs no introductory material vs. the user is new to a topic where introductory material is considered helpful
- the user's context especially his task influences the needed information
- the words of the user do have a semantic which is not necessarily the one used in the information source(s)

These statements provide requirements on features which should be considered when designing an information need-model.

Therefore, EPOS considers the following elements for gaining an appropriate model for the user's information need:

user profile elements of the information need should be grounded in the Personal Information Model which give indications for interpreting a stated information need (see chapter 2)

context-dependency the user's context needs to be considered especially his current information state, goal, and task at hand (see chapter 3)

history previous searches of the user will be considered as they provide indications for the current information need-interpretation (see chapter 4)

4.5 Basic term-based query manipulation services

4.5.1 Machine Learning in Information Retrieval

Machine learning has been used in Information Retrieval for years now. Several research fields in the area of Information Retrieval have benefitted from Machine Learning. The learning techniques used in Information Retrieval are taken from the full spectrum of learning techniques which use Artificial Intelligence as a basis.

Most of the work has been done in the fields of

- categorization or classification or clustering of text documents (refer to [Lewis and Gale, 1994], [Krulwich, 1995a], [Martin, 1995], [Cohen, 1995], [Joachims, 1998])
- summarization of contents of text documents (refer to [Turney, 1997])
- learning user interests and personalization (refer to [Krulwich, 1995b], [Benammar *et al.*, March 2002])
- learning query expansions (refer to [Gauch and Smith, 1993])
- learning similarities between queries (refer to [Raghavan and Sever, 1995], [Wen *et al.*, 2001], [Tombros and van Rijsbergen, 2001], [Tombros *et al.*, 2002], [Wen *et al.*, 2002])
- learning similarities between documents (refer to [Hofmann, 2000])
- learning similarities between documents and queries (refer to [Bartell *et al.*, 1994] [Bartell *et al.*, 1998])
- Term ambiguities learning (refer to [Yarowsky, 1992], [Pirkola, 1999], [Pirkola, 2001])
- learning ranking functions according to users' preferences (refer to [Stahl, 2001], [Stahl, 2002], [Stahl and Schmitt, 2002])

In recent years researchers have turned to newer artificial-intelligence based learning techniques [Chen, 1995] including neural networks [Mandl, September 1998], [Mandl, September 1999], symbolic learning, and genetic algorithms and support vector machines (SVM) [Vapnik, 1995].

4.5.2 Learning Similarity Functions

The motivation for learning similarity functions arises from the achieved performance improvements of our query expansion methods in CIR. These algorithm are based on this idea: if a newly entered query has a similarity to one or more old queries above a specific threshold, than the documents which are relevant to the new query can be derived from the documents which are relevant to the old queries.

Similarity between queries as it is used in CIR up to now is solely based on syntactical elements of the underlying collections. Although we have used some normalization and

cleaning operations (stemming and stopword-elimination) there is no further processing beyond the syntactical level. Similarity between two queries is high if they use the same words. Similarity is low if they use different words.

As we have stated before, the same information need can be expressed in different queries. Different queries may then have a low inter-query similarity although they are querying for the same facts, and thus may have the same relevant documents. However, the methods developed up to now only use the inter-query similarity on the syntactical level, they do not consider the information need of the user.

After learning has been done the similarity functions may return different queries that are similar to the new query. This is illustrated in figure 4.3, where the area of nearest neighborhood may change dramatically if the newly learned similarity functions are applied. In this way we can identify queries as nearest neighbors of a new query, even if they are far away (according to the standard cosine-similarity) from the new query.

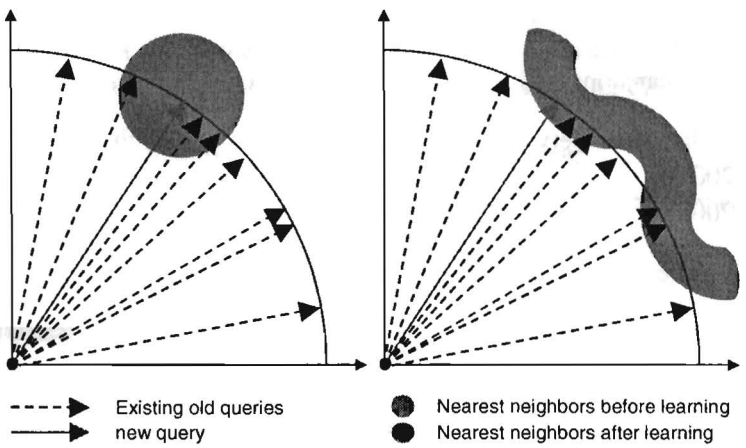

Figure 4.3: Motivation for Learning Similarities: usage of the nearest neighbors

Thus we try to learn the similarities of the information need although it is expressed in syntactically different queries. On a semantic level the similarity of queries has to consider synonyms, homonyms (one of two or more words spelled and pronounced alike but different in meaning, as 'cleave' meaning 'to cut' and 'cleave' meaning 'to adhere' [Cooper, 2003]) and polysemy (ambiguities of words , as 'bank' meaning 'financial institution' or 'dike').

Chapter 5

Visualization of Models, Structures and Query Results

5.1 Objectives

The objectives within the project's visualization thread are to present to the users the the structures within their personal knowledgespaces or within the company's organizational memory that are of relevance to their current work. Since these structures may be of very different form like documents, processes, user models, workflows or personal or organizational information models, the main objective of this project line is to establish a flexible visualization framework that can present the different visualization metaphors that correspond to the different forms of knowledge with appropriate views. This requires a flexible, pluggable and extendable framework where customizable view elements can be integrated according to the actual visualization needs and presented to the user within a generalized environment.

5.2 Related Work

A survey on graph visualization and navigation techniques as used for information visualization purposes is given in [Herman *et al.*, 2000]. The survey focuses on applications such as web browsing, state–transition diagrams, and data structures. However, several of these techniques also are of interest for visualizing content and relations within a user's personal workspace. Moreover, the paper also looks at the specific requirements for graph visualization techniques in the context of information visualization topics as e.g. the avility to visualize and to navigate in these potentially large and abstract graphs.

Apart from this general survey, in the next sections a selection of projects is presented that are of interest to the visualization tasks within the EPOS project.

5.2.1 Project "INVITE"

The project INVITE see [Beinhauer and Ziegler, 2003] aims at developing technologies, that promote creativity, knowledge-exchange, and continuous learning in order to support the establishment of a future information-based "Knowledge-Society". Therefore, the project is targeted at making the use of the increasing information capacities and the complex functionality of information and communication systems for humans controllable, effective and attractive.

In order to achieve these objectives, work is performed in three domains:

- Information representation: In order to support the understanding of complex information, perception-fair forms and different means of visual and acoustic presentation are used. 2D and 3D presentations thereby support orientation, overview and flexibility.
- Multilevel-Control: Conventional input techniques are enhanced by language processing and gesture recognition techniques in order to enable the adjustment to different levels of user-knowledge and user-preferences.
- Intelligent Assistants: Complex information and complex system functionalities are made controllable by the application of active and intelligent assistants. They provide user support by filtering, editing and adjusting the data. Unobtrusive and user-oriented systems thereby increase the privacy and acceptance of the system.

Their work on information representation and presentation within the intelligent assistants can give valuable impulses to our work in the related areas of the EPOS project.

5.2.2 Matrix Browser

In [Ziegler *et al.*, 2002] Ziegler et. al. present a new approach for visualizing and exploring large networked information structures. These structures represent, for instance, linked information resources or metadata structures such as ontologies. They use an interactive matrix to display relations between concepts and concept hierarchies presented along the two axes of a matrix. Their approach also focuses on the engineering process to create these information networks. While the creation of the relations among the different information objects is done differently within the EPOS system, matrix browser could be an interesting visualization metaphor to be used in our context, too.

5.2.3 Visual Database for Example-based Graphics Generation

Example-based graphics generation systems automatically create new information visualization views by learning from existing graphic examples. As part of their effort to develop a general-purpose example-based graphics generation system, [Zhou *et al.*, 2002] are building a visual database of graphic examples. In doing this, they address two main issues involved in constructing such a knowledge base: example selection and example modeling. They present a visual database that contains a diverse collection of well-designed examples plus a feature-based scheme to model all examples uniformly and accurately. Such an approach is very useful for generating and retrieving the view elements that within the EPOS system will be used to represent the instances within the ontology of visualization tasks and objects.

5.2.4 DocMiner: Visual Knowledge Management with Adaptable Document Maps

Structuring and condensing corporate document collections is an important task of knowledge management. For supporting that task [Becks and Jarke, 2001] have developed the corpus analysis tool DocMINER, which provides interactive visual access to text collections. The system can be used for critiquing technical document collections such as use-case descriptions

in software engineering, or user manuals of complex engineering systems. These techniques described here will be evaluated with respect to how the system's visualization metaphors can be adapted and used within the EPOS visualization system.

5.2.5 Perspective Layered Workspace for Collaborative Work

The work presented here addresses the sharing of information within a group by using visual shared workspaces for synchronous group-ware. In such a scenario, users will be moving frequently between their personal workspaces for personal and asynchronous work and shared workspaces for communication and synchronous cooperation. For the task of visualizing the relations between a user's personal workspace and common workspaces shared among groups of users, [Shiozawa *et al.*, 1999] have developed the "Perspective Layered Workspace". This workspace consists of a set of layered virtual screens in pseudo-3-dimensional graphics. In this way, the workspaces shared among groups are shown as the background of the users' personal workspaces like as looking from a top personal layer down to a bottom public layer. Within the EPOS project, this metaphor can also be used for representing the relations between the users' personal workspaces and their company's organizational memory.

5.2.6 Scalable Framework for Information Visualization

This project addresses the question of dealing with large and heterogenous information spaces. As a solution, [Kreuseler *et al.*, 2000] present the concept of a scalable information visualization framework. The framework is based on the assumption that the exploration of heterogenous information spaces at arbitrary levels of detail requires a suitable preprocessing of information quantities, the combination of different graphical interfaces and the illustration of the frame of reference of given information sets. In order to satisfy these assumptions, their system includes dynamic hierarchy computation and user controlled refinement of those hierarchies for preprocessing unstructured information spaces. Additionally, they present a new Focus+Context technique for visualizing complex hierarchy graphs, a new paradigm for visualizing information structures within their frame of reference and a new graphical interface that utilizes textual similarities to arrange objects of high dimensional information space in 3-dimensional visualization space. For the work within the EPOS project, these visualization paradigms are of high interest.

5.3 Fields of Application

In cooperation with the project's other threads, relevant tasks within the envisaged user's work environment have been evaluated. This includes a categorization of the users' activities that has been performed in cooperation with the definition of a model of the user's personal information space objects (see section 3.1). The result of this evaluation showed needs for visualizing knowledge structures in two different domains:

5.3.1 Visualizing Information Objects and their Relations to the User's Personal Workspace and Tasks

User observation produces sequences of user actions which partly belong to more abstract tasks that are in turn performed to satisfy a high-level goal. Our survey showed the user's need

for visualizing on one hand the current action in the context of these higher level tasks and goals. On the other hand, the survey also revealed the need for representing the relation of the documents that are handled within one action to similar documents of the user's personal workspace.

5.3.2 Visualizing the Network of the EPOS Personal Workspaces of the Company

One special task within the user's workflow is the search for certain information. Within the EPOS framework, search is to be performed in a network of the single EPOS personal workspace agents within the company. In this domain, our survey also resulted in twofold needs for visualization of these search processes:

Visualization of Network Simulation

On system level, for supervising the functionality of the system, we identified the need for visualizing the network of the personal workspaces itself. Here special attention should be paid to visualizing the network activities and presenting to the user the functions of single network nodes and their contributions to the current search. This will enable a network administrator to ensure the proper functioning of the network and get him the necessary information for maintenance or debugging tasks in case of some abnormal behavior of the network.

Visualization of Queries to the Network

From the point of a user that performs a query, it is irrelevant, how the network computed the search results. For him only the results themselves are of interest and he wants to know how trustworthy they are and possibly which justifications the network has got for these results. So, in contrast to presenting the user the complete network and the activities of the nodes involved in the search, for him a representation of the search results is of interest, accompanied by e.g. a sort of trust-clusters or topic-clusters that justify the results and that indicate their trustworthiness to the user.

5.4 An Ontology of Objects to Visualize

On the basis of these visualization-needs identified, a first version of a categorization of objects to visualize has been established as described within the workpackage "VIS-OBJONT" in cooperation with the workpackages "PW-DEF" and "OL-IDENT". On top level we have identified three classes of objects that influence the visualization of information objects. These classes will now be described in more detail.

5.4.1 Categorization

Metaphors

A visualization metaphor is a means for describing similarities between application domain entities and visual objects. By using the metaphor, the idea behind the visual objects for representing specific information is presented by some generally known concept from a well-known domain.

The metaphors described in our ontology include on top-level

- Graph-like
- Diagram-like
- Timeline-like and
- Workflow-like

data.

These top-level classes of course can be subdivided down to classes that finally will be realized in the form of specific implementations. These include, e.g.,

- Undirected Graph
- Tree
- Equidistant Timeline or
- Pattern Map.

Paradigms

Orthogonal to the metaphors there need some concepts to be specified, how elements within a metaphor can be visualized. These concepts may include providing an overview or detail view of the data or they can deal with handling single elements. We have subsumed these concepts in the class of paradigms for handling information visualization tasks. Some well-known examples of such paradigms are

- Focus and Context
- Fisheye Lens or
- Zoom and Pan.

Implementations

Finally, a specific visualization method has to be instantiated. This is done within an element of the class Implementation. Such an element describes a specific realization of one metaphor, possibly including one or several paradigms. Examples for implementations known in the information visualization communite include

- Hyperbolic Tree
- Arc Diagram
- Pie Chart
- GANTT Chart or
- Arc Diagram.

An excerpt of the items within our ontology of objects to visualize is given in figure 5.1.

5.4.2 Representation

This ontology of objects to visualize described within the previous section has been implemented within the ontology modeling tool Protégé. Figure 5.1 shows a screenshot of the ontology within this tool. An example of an implementation, the hyperbolic tree is displayed including its relations to the other objects within the ontology in figure 5.2.

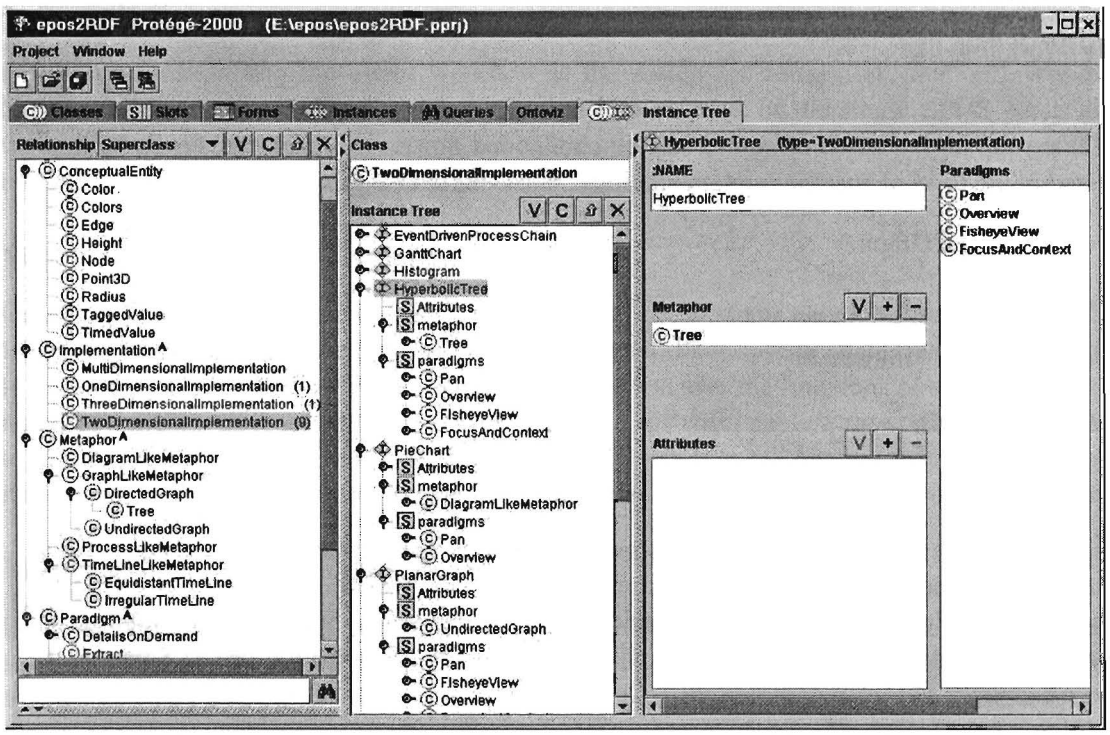

Figure 5.1: Representation of the ontology in Protégé

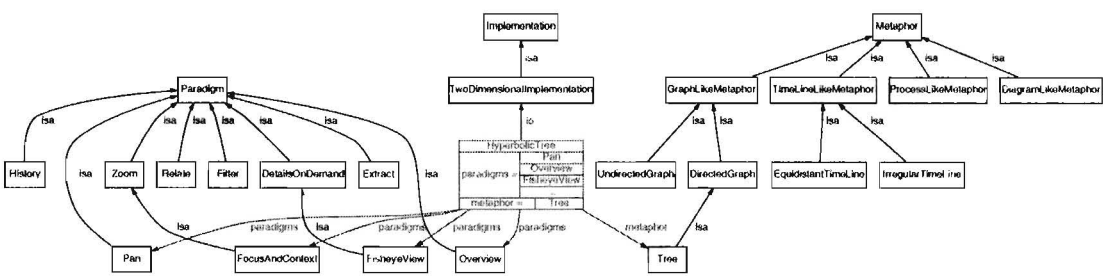

Figure 5.2: Example of an implementation and its relations within the ontology: the Hyperbolic Tree

Due to restrictions on our resources, for the implementation of the prototype system, we will for now restrict on graph-like data and the corresponding implementations.

5.5 Visual Representation of Objects

Within the ongoing workpackages "VIS-BASICFKT" and "VIS-MAPPING", first existing frameworks for visualizing graph-like data have been evaluated. In this context, an overview of existing visualization techniques for this domain is currently compiled with the objective to select a representative subset of implementations that in the final system will be used to represent the visualization metaphors of the ontology of objects to visualize. Additionally, a visualization architecture has been defined, that permits the integration of the different visualization tasks into the EPOS prototype software system and yet is flexible enough to be adapted during the progress of the project according to possible extensions in the metaphors and implementations to be presented to the user. This is achieved by using an architecture based on software-components in a pluggable pipe-filter architecture.

5.5.1 Architecture

The objective of the EPOS visualization component is to provide a flexible context-driven visual Interface on the given data for the user. This is to be achieved both interactively and statically. The data can be provided in different forms like, e.g., graph-like data or a workflow. In our opinion the best way to create a similar look and feel for different source-data in different user-contexts is to use a pipeline-architecture with pluggable processors that filter the relevant data from the raw data according to the user intention on the one hand and that select the appropriate metaphor- and implementation-templates in order to use them to build up the visual representation of the relevant data on the other hand (see figure 5.3). We believe that this approach would enable future enhancements of the framework by adding new metaphor and implementation templates and corresponding data filtering and metaphor- / implementation-selection rules for different forms of input. Thus a wider range of data can be processed than is known at design time of the framework.

We thereby define the need of the raw-data to be of RDF-form. This data is then processed in a first step to extract the relevant data from the raw-data according to the user intention. An appropriate filter is selected from a rulebase mapping user intention to possible filters.

The relevant data extracted that way form together with the user intention and the visualization context (e.g., presenting data on a PDA or on a graphics workstation, printing data, etc.) the input for the selection of the appropriate visualization metaphor and implementation. Built up on a set of implementation atoms like nodes, edges, curves, lines, pies, boxes, etc., a database contains a given number of metaphor and corresponding implementation templates that are realized in a declarative description language, that we call the View Modeling Language (VML). A rulebase that maps the input triple of relevant data, user intention and visualization context on a pair of a metaphor and one of its implementations then selects an appropriate template from the database.

In a final step the selected template is filled with the relevant data, thus building up the specific instance of the implementation. If needed, in this step the relevant data will be converted into a format fulfilling the needs of that implementation, too. The RDF-based implementation-object that is generated in that process can then be used in either an interactive or a static way.

If the output of the visualization generation is to be static, no further processing is needed. The graphical representation object is sent to an SVG-generator (Scalable Vector Graphics) that produces a static rendering of the object. The latter is embedded into a Java JPanel-Object which is then given back to the EPOS-System.

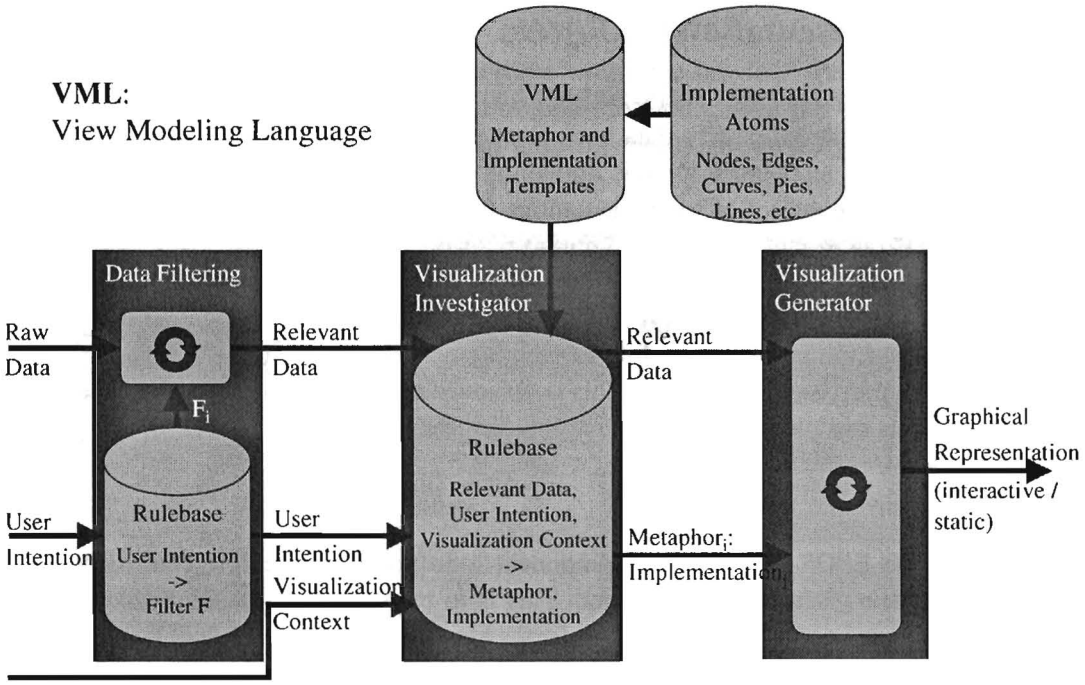

Figure 5.3: Visualization pipeline

Should the output be of interactive kind, the SVG-creator is not used but the data is directly rendered onto a specialized Java JPanel-Object that provides direct means of interaction by context-menus and/or one or more embedded Java JMenu-Objects that are to be included into the calling system.

5.5.2 Implementation Aspects

Being designed as a framework, the EPOS visualization-architecture is ensured to be both highly flexible in adding an extension-support and highly reusable in hiding frequently used parts from the developer after having built them once. This can be achieved by using a JavaBean-structure for larger modules whilst using abstract base-classes as caller-arguments that can be derived for adding specialized functionality.

The gain we get by making the system a framework on the other hand is compensated by the need of a stricter design-process since changing the structure of a partially running framework is much harder than it is with a simple architecture. By the use of Software-Components we can design whole parts of the system not directly in the form of code but only in specifying an interface and an expected functionality. This design-process tends to stay simple.

One thing to be thought of before starting the implementation is memory-management. By providing a garbage-collector Java-programs normally do not need a special memory-management. However, the communication of components within a framework rises issues in the management of links that can become inactive. To prevent such problems we propose the use of connection-management objects that use weak references.

The structure of a bean-based framework tends to need a lot of synchronizing communica-

tion if the amount of components is high. To prevent problems rising from this issue we propose to form modules that are implemented as components. So by providing a few main-modules, easily declarable by functionality, we keep inter-module communication low.

Aside from the framework-aspects, in our opinion the use of a pipe-filter-architecture matches best the different needs of the EPOS-visualization software. In specifying the input data to be required in RDF-form and the output to be a JPanel, we have got a strict format definition for both input and output, yet we have got a great flexibility in its content.

If a direct interaction between the surrounding software system and the visualization components is needed, the proposed architecture also provides the means for interacting with the implementation components. This is achieved by implementing the interaction methods directly within the visualization components that then are presented to the user within the JPanel (see figure 5.3).

5.5.3 Evaluated Graph Visualization Frameworks

For implementing the visualization architecture described above, we plan to use an already existing visualization framework. In the context of the work performed within the EPOS project a number of these frameworks has been evaluated. The most important ones will be introduced in this section.

Giny

The Graph INterface librarY (GINY) is an Open Source Graph Library. It has been developed for viewing complex networks through a zoomable user interface. The visual side of GINY is implemented using the twodimensional graphics toolkit Piccolo, which has been implemented in the Java programming language. Algorithms available in the GINY library include All Pairs Shortest Path, Spring Embedded Layout and Sugiyama Hierarchical Layout. However, in this toolkit the visual representation is not very well separated from the user interaction.

jGraph

JGraph is an open-source graph-drawing component for the Java programming language. It is accompanied by JGraphpad, a diagram editor that offers reading and writing XML-documents and Drag and Drop drawing capabilities. However, for our purposes this tool is not useful since its intention is more on drawing (like, e.g., Microsoft Visio). The tool offers little user-interaction, it is difficult to visualize tree-oriented implementations, and there are no auto-layouting algorithms available like in other toolkits.

jrdf

Since the input data dealt within the EPOS visualization is provided in RDF-form, the Java RDF (JRDF) API set is also a candidate for representing the graph structures to be dealt with in this project. Among other interfaces for storing, querying and creating RDF statements, JRDF also intends to provide a graph API. However, the toolkit is in a much too early development phase to be used within the EPOS project (in December 2003 version 0.2 has been released).

gvf (GraphVisualizationFramework) and Royere

The Graph Visualization Framework is a flexible and extensible set of Java packages that can serve as a foundation for applications that either manipulate graph structures or visualize them. The libraries implement several basic modules for input, graph management, property management, layout, and rendering. Layouts paradigms include Reingold-Tilford (hierachical - adapted for directed acyclic graphs), Fruchterman-Reingold (force-directed), Radial, Ring, Barycentric, Random. The user can add his own pluggable layouts, metrics, clusterings, coloring. The library contains readers for GML, GraphXML, and CNS (Newick Format). The "focus and context" and "fish-eye view" paradigms are also already implemented in this toolkit. New behavior can easily be implemented in this framework since it is OpenSource and well-documented. Royere is a software-tool for displaying graphs that is built upon this framework. This toolkit will be used as a basis for the EPOS visualization framework.

5.6 Current Implementation Status and Future Work

Currently there exists a first prototype implementation of the visualization component based on previous work performed in the predecessor project FRODO. Here the basic components of our architecture as described in section 5.5 are realized as first prototypes with limited functionality. The data filtering mechanism is performing the extraction of relevant data according to a specific view on the workflow context. These relevant data are then represented as a graph, that can be presented to the user within different implementations that the user can select directly. The whole architecture of the existing prototype is shown in figure 5.4.

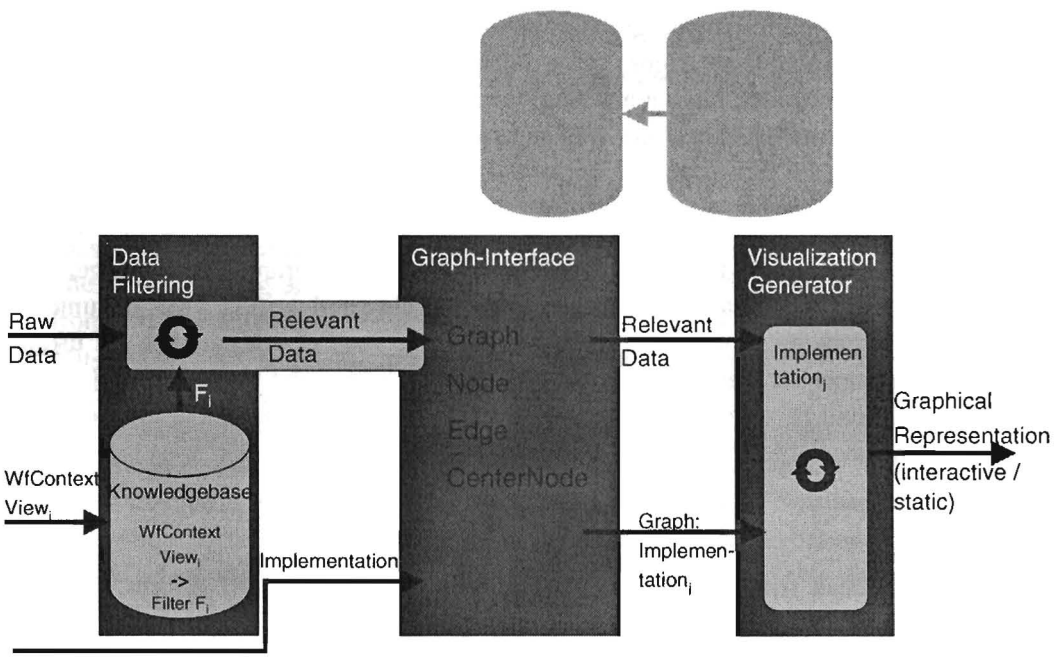

Figure 5.4: Current Prototype Implementation

The next steps in our work now will focus on filling the relevant rulebases described in

figure 5.3. We will use the ontology of visualization objects that we described within section 5.4 to realize the templates for metaphors and implementations in our envisioned declarative description language VML (View Modeling Language). Additionally, we will develop the knowledgebase used within the visualization investigator component for selecting the appropriate metaphor and implementation according to the relevant data, the user intention and the visualization context. Accordingly, we will refine the rulebase of the data filtering component used for choosing the appropriate filter to select the relevant data depending on the current user intention.

Bibliography

[AdaptiveRead03, 2003] Empirische Studie zum Werkzeugeinsatz im Umfeld der Dokumentverwaltung. Studie, AdaptiveRead Konsortium, 2003.

[Baeza-Yates and Ribeiro-Neto, 1999] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Publishing Company, 1999.

[Bartell *et al.*, 1994] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Learning the optimal parameters in a ranked retrieval system using multi-query relevance feedback. In *Proceedings of Symposium on Document Analysis and Information Retrieval*, Las Vegas, NV, 1994.

[Bartell *et al.*, 1998] B. T. Bartell, G. W. Cottrell, and R. K. Belew. Optimizing similarity using using multi-query relevance feedback. *Journal of the American Society for Information Science and Technology*, 49(8):742–761, 1998.

[Bauer and Leake, 2001] T. Bauer and D. B. Leake. Real time user context modeling for information retrieval agents. In *CIKM*, pages 568–570, 2001.

[Becks and Jarke, 2001] A. Becks and M. Jarke. Visual knowledge management with adaptable document maps. *ERCIM News*, (46):54 – 55, July 2001.

[Beinhauer and Ziegler, 2003] W. Beinhauer and J. Ziegler. Intuitive interaction in complex information spaces - results and exploitation of invite. In *Proceedings of Human Computer Interaction Status Conference*, Berlin, Germany, June 2003.

[Benammar *et al.*, March 2002] A. Benammar, G. Hubert, and J. Mothe. Automatic profile reformulation using a local document analysis. In F. Crestani, M. Girolami, and C. J. van Rijsbergen, editors, *Advances in Information Retrieval, 24th BCS-IRSG European Colloquium on IR Research, ECIR 2002, Proceedings*, volume 2291 of *Lecture Notes in Computer Science*, pages 124–134, Glasgow, UK, March 2002. Springer.

[Chen, 1995] H. Chen. Machine learning for information retrieval: Neural networks, symbolic learning and genetic algorithms. *Journal of the American Society for Information Science and Technology*, 46(3):194–216, 1995.

[Cohen, 1995] W. W. Cohen. Text categorization and relational learning. In A. Prieditis and S. J. Russell, editors, *Proceedings of ICML-95, 12th International Conference on Machine Learning*, pages 124–132, Lake Tahoe, US, 1995. Morgan Kaufmann Publishers, San Francisco, US.

[Cooper, 2003] A. Cooper. Homepage, 2003. `http://www.cooper.com/alan/homonym.html`.

[Gauch and Smith, 1993] S. Gauch and J. B. Smith. An expert system for automatic query reformulation. *Journal of the American Society for Information Science and Technology*, 44(3):124–136, 1993.

[Henrich, 2002] A. Henrich. IR research at university of bayreuth. Homepage of the IR-research group, 2002. `http://ai1.inf.uni-bayreuth.de/forschung/forschungsgebiete/ir_mmdb`.

[Herman *et al.*, 2000] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, /2000.

[Hofmann, 2000] T. Hofmann. Learning the similarity of documents: An information-geometric approach to document retrieval and categorization. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 914–920, Cambridge, MA, 2000. MIT Press.

[Hust *et al.*, 2002a] A. Hust, S. Klink, M. Junker, and A. Dengel. Query Expansion for Web Information Retrieval. In S. Schubert, B. Reusch, and N. Jesse, editors, *Proceedings of Web Information Retrieval Workshop, 32nd Annual Conference of the German Informatics Society*, volume P-19 of *Lecture Notes in Informatics*, pages 176–180, Dortmund, Germany, October 2002. German Informatics Society.

[Hust *et al.*, 2002b] A. Hust, S. Klink, M. Junker, and A. Dengel. Query Reformulation in Collaborative Information Retrieval. In M. Boumedine, editor, *Proceedings of the International Conference on Information and Knowledge Sharing, IKS 2002*, pages 95–100, St. Thomas, U.S. Virgin Islands, November 2002. ACTA Press.

[Hust *et al.*, 2003] A. Hust, S. Klink, M. Junker, and A. Dengel. Towards Collaborative Information Retrieval: Three Approaches. In J. Franke, G. Nakhaeizadeh, and I. Renz, editors, *Text Mining - Theoretical Aspects and Applications*. Physica-Verlag, 2003.

[Hust *et al.*, 2004] A. Hust, M. Junker, and A. Dengel. A Mathematical Model for Improving Retrieval Performance in Collaborative Information Retrieval. *Kluwer Information Retrieval Special Issue: Advances in Mathematical/Formal Methods in Information Retrieval*, 2004. to appear.

[Hust, 2004] A. Hust. Introducing Query Expansion Methods for Collaborative Information Retrieval. volume 2956 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, New York, 2004. Springer-Verlag. to appear.

[Joachims, 1998] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In C. Nédellec and C. Rouveirol, editors, *Proceedings of 10th European Conference on Machine Learning ECML-98*, number 1398 in Lecture Notes in Computer Science, pages 137–142, Chemnitz, Germany, April 1998. Springer Verlag, Heidelberg.

[Klink *et al.*, 2002a] S. Klink, A. Hust, M. Junker, and A. Dengel. Collaborative Learning of Term-Based Concepts for Automatic Query Expansion. In *Proceedings of ECML 2002, 13th European Conference on Machine Learning*, volume 2430 of *Lecture Notes in Artificial Intelligence*, pages 195–206, Helsinki, Finland, August 2002. Springer.

[Klink *et al.*, 2002b] S. Klink, A. Hust, M. Junker, and A. Dengel. Improving Document Retrieval by Automatic Query Expansion Using Collaborative Learning of Term-Based Concepts. In *Proceedings of DAS 2002, 5th International Workshop on Document Analysis Systems*, volume 2423 of *Lecture Notes in Computer Science*, pages 376–387, Princeton, NJ, USA, August 2002. Springer.

[Klink, 2004] S. Klink. Improving document transformation techniques with collaborative learned term-based concepts. volume 2956 of *Lecture Notes in Computer Science*. Springer, 2004. to appear.

[Kreuseler *et al.*, 2000] M. Kreuseler, N. López, and H. Schumann. A Scalable Framework for Information Visualization. In *Proceedings of IEEE Symposium on Information Visualization*, Salt Lake City, Utah, October 2000.

[Krulwich, 1995a] B. Krulwich. Learning document category descriptions through the extraction of semantically significant phrases. In *Proceedings of International Joint Conference on Artificial Intelligence IJCAI-95, Workshop on Data Engineering for Inductive Learning*, Montreal, August 1995. Morgan Kaufmann.

[Krulwich, 1995b] B. Krulwich. Learning user interests across heterogeneous document databases. In *Proceedings of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, Stanford, March 1995. AAAI Press.

[Lewis and Gale, 1994] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In W. B. Croft and C. J. van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12, Dublin, Ireland, July 1994. Springer-Verlag, New York, NY, USA.

[Mandl, September 1998] T. Mandl. Learning similarity functions in information retrieval. In H.-J. Zimmermann, editor, *EUFIT '98. 6th European Congress on Intelligent Techniques and Soft Computing*, pages 771–775, Aachen, Germany, September 1998.

[Mandl, September 1999] T. Mandl. Efficient preprocessing for information retrieval with neural networks. In H.-J. Zimmermann, editor, *EUFIT '99. 7th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, September 1999.

[Manning and Schütze, 1999] C. D. Manning and H. Schütze. *Foundations of Natural Language Processing*. MIT Press, 1999.

[Martin, 1995] J. D. Martin. Clustering full text documents. In *Proceedings of International Joint Conference on Artificial Intelligence IJCAI-95, Workshop on Data Engineering for Inductive Learning*, Montreal, August 1995. Morgan Kaufmann.

[Pirkola, 1999] A. Pirkola. *Studies on linguistic problems and methods in text retrieval: The effects of anaphor and ellipsis resolution in proximity searching, and translation and query structuring methods in cross-language retrieval.* PhD thesis, Department of Information Studiesm University of Tampere, Tampere, Finland, 1999.

[Pirkola, 2001] A. Pirkola. Morphological typology of languages for ir. *Journal of Documentation*, 57(3):330–348, 2001.

[Pratt *et al.*, 1999] W. Pratt, M. A. Hearst, and L. M. Fagan. A knowledge-based approach to organizing retrieved documents. In *Proceedings of the 16th National Conference on AI (AAAI '99) held at Orlando, Florida*, pages 80–85, 1999.

[Raghavan and Sever, 1995] V. V. Raghavan and H. Sever. On the reuse of past optimal queries. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 344–350, Seattle, Washington, USA, July 1995. ACM Press, New York, NY, USA.

[Schwarz, 2003] S. Schwarz. Task-Konzepte: Struktur und Semantik für Workflows. In U. Reimer, A. Abecker, S. Staab, and G. Stumme, editors, *WM 2003: Professionelles Wissesmanagement - Erfahrungen und Visionen, Beiträge der 2. Konferenz Professionelles Wissensmanagement, 2.–4. April 2003 in Luzern*, volume 28 of *LNI*. GI, 2003.

[Shiozawa *et al.*, 1999] H. Shiozawa, J. ya Node, K. ichi Okada, and Y. Matsushita. Perspective layered workspace for collaborative work. In *Proceedings of International Workshops on Parallel Processing*, Wakamatsu, Japan, September 1999.

[Stahl and Schmitt, 2002] A. Stahl and S. Schmitt. Optimizing retrieval in CBR by introducing solution similarity. In *Proceedings of the International Conference on Artificial Intelligence, IC-AI'02*, Las Vegas, USA, June 2002.

[Stahl, 2001] A. Stahl. Learning feature-weights from case order feedback. In D. W. Aha and I. Watson, editors, *Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR 2001*, pages 501–516, Vancouver, BC, Canada, July/August 2001. Springer-Verlag, Berlin, 2001.

[Stahl, 2002] A. Stahl. Defining similarity measures: Top-down vs. bottom-up. In *Proceedings of the 6th European Conference on Case-Based Reasoning, ECCBR 2002*, Aberdeen, Scotland, UK, September 2002.

[Tombros and van Rijsbergen, 2001] A. Tombros and C. J. van Rijsbergen. Query-sensitive similarity measures for the calculation of interdocument relationships. In *Proceedings of the 10th ACM CIKM Conference*, pages 17–24, Atlanta, Georgia, USA, November 2001.

[Tombros *et al.*, 2002] A. Tombros, R. Villa, and C. J. van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing & Management*, 38(4):559–582, July 2002.

[Turner *et al.*, 2001] R. M. Turner, E. H. Turner, T. A. Wagner, T. J. Wheeler, and N. E. Ogle. Using Explicit, A Priori Contextual Knowledge in an Intelligent Web Search Agent. In V. Akman, P. Bouquet, R. Thomason, and R. Young, editors, *Modeling and Using Context. 3rd International and Interdisciplinary Conference, CONTEXT'01*, volume 2116 of *LNAI*, pages 343–352. Springer, 2001.

[Turney, 1997] P. D. Turney. Extraction of keyphrases from text: Evaluation of four algorithms. Technical report NRC-41550, National Research Council of Canada, Institute for Information Technology, 1997.