K-Discovery: Identification of Distributed Knowledge Structures in a Process-Oriented Groupware Environment

Stefan Smolnik
Department of Business Computing 2, University of Paderborn, Germany
Stefan.Smolnik@notes.uni-paderborn.de
http://gcc.uni-paderborn.de

Ludwig Nastansky
Department of Business Computing 2, University of Paderborn, Germany
Ludwig.Nastansky@notes.uni-paderborn.de
http://gcc.uni-paderborn.de

Abstract:

Scenarios in groupware-based environments show the problems of accessing knowledge structures in general and organizational knowledge structures in particular. By applying topic maps, as defined in ISO/IEC 13250, to groupware-based organizational knowledge bases, it is possible to close a gap between knowledge and information. In this paper, the aims of the K-Discovery project – applying topic maps to groupware-based environments – are presented. Based on this, an architectural model will be introduced creating knowledge structures by generating topic maps in a process-oriented groupware environment.

After a brief introduction and overview, two concrete scenarios of groupware-based office systems are described. The basic ideas of the K-Discovery project are presented and followed by an architectural model and two implementation approaches. This paper ends with some conclusions.

1. Introduction and Overview

In many companies, groupware-based environments are the basis of communication and information management. The increasing integration of these environments in internal business processes leads to growing information memories. These shared databases, which often exists in groupware-based office systems, enable the transformation of the *individual* knowledge of single employees into a *common* knowledge of all employees ([Wagner 1995], p. 3). Looking at the technical perspective of knowledge management groupware seems to be a suitable platform for collecting and distributing the organizational knowledge base [Schliwka 1998].

The growing flood of available knowledge requires powerful concepts and mechanisms to support users who search for relevant information and knowledge objects. Mechanisms for navigating and linking, as well as functionalities for extensive searches and investigations, are needed to explore and use the complex information and knowledge offer. They are a necessary condition for the core processes of knowledge identification and knowledge use [Probst et al. 1999]. Therefore, effective search mechanisms, which provide an improved organizational use of existing individual and common information and knowledge objects,

contribute to the process of knowledge generation (the development or collection of new knowledge) ([Güldenberg 1998], p.248).

Topic maps – as defined in ISO/IEC 13250 – used on information sets create knowledge structures and form a structured semantic linked network upon a great set of information [Rath/Pepper 1999]. Therefore, topic maps are a perfect basis to realize the mentioned mechanisms and functionalities for the identification of relevant information and knowledge objects. The fundamental idea of the described project K-Discovery is to use topic maps for the identification of distributed knowledge structures in a process-oriented, groupware-based environment. In addition to the existing techniques for searching and navigating, like the hierarchical navigation in categories of views or full text searching, the user can be supported by the strong concept of associative navigation in semantic networks [cf. Ahmed 2000]. Further, a groupware-based implementation of topic maps participates in several aspects in groupware technologies.

Chapter 2 describes two concrete scenarios in groupware-based environments to demonstrate the problems of accessing knowledge structures in general and of organizational knowledge structures, in particular. The principle aims and underlying ideas of the K-Discovery project are presented in Chapter 3. Based on this, in Chapter 4 an architectural model will be introduced in which knowledge structures are created by generating topic maps in a process-oriented, groupware-based environment. Furthermore, we describe two approaches of implementations that provide some of the functionalities of the architectural model in Chapter 5. Chapter 6 outlines concluding remarks and the current state of the project.

2. Scenarios in Groupware-based Environments

As pointed out above, groupware-based office systems provide an excellent environment for organizational knowledge management. In many cases, however, the groupware infrastructure is used to support teams in communication, coordination and collaboration. (The delimitation of these three terms is not homogeneous in literature. For an overview of differing controversial viewpoints please consult Bornschein-Grass [Bornschein-Grass 1995]. We are refering to the definition by Lotus [Lotus 1995].) Effective access mechanisms are needed because of an increasing amount of information and knowledge objects in shared information spaces of such a groupware-based infrastructure. The following two small cases illustrate the need for these mechanisms:

A) The groupware-based web publishing system NetFicient, designed and implemented by Deutsche Bank AG and Lotus Development GmbH, forms a complex information network and allows users to publish web pages in the organizational intranet or on the Internet without having knowledge in web programming or server administration (v. [Deutsche Bank 2000]). The core of this application consists of five Lotus Notes databases: Homepage, PowerPublishing, Newspanel, Feedback and Archive. Additionally, there are databases for moderated discussions, a container of files, and a bulletin board as well as databases to support project teams in communicating and coordinating. The elementary components of NetFicient are the PowerPublishing databases in which users publish information by means of several predefined document types. Two kinds of workflows are provided: First, a content expiry workflow with resubmission and archiving functionalities controls the publishing date and the publishing period. Second, a content approval process ensures that all published contents adhere to the organizational standards. The capabilities of accessing the information and knowledge objects are restricted to a hierarchical navigation and a basic full text search. [Deutsche Bank 2000]

   Within Deutsche Bank, NetFicient is used to build up several more or less linked intranets. A lot of departments and teams have set up their own NetFicient implementation to publish information

relevant to their intranets. As a consequence, several isolated islands of knowledge have arose with only a few weak linkages – or even no linkages – between them.

B) In the Lotus Notes/Domino environment, all the benefits of integrated correspondence, office, document, workflow, knowledge and archive management are brought together by PAVONE Enterprise Office (v. [Pavone 2001]). This office system is based on a powerful process management toolset and consists of modular components, allowing users to mix and match to meet their requirements. Supporting components are databases that capture and reflect the organizational structure, to provide text building blocks and models as well as to deal with processes (e.g. definition, tracking, and escalation). Two external tools enable the graphic definition, creation and maintenance of the organizational structure and of the business processes. The core application provides functionalities for office organization, office communication, address management, correspondence and information management, and process optimization. Again, the capabilities for accessing the information and knowledge objects are restricted to basic full text searching and navigating through context sensitive views and categories. [Pavone 2001]

   Within the Groupware Competence Center of the University of Paderborn, Enterprise Office is used in almost every part of operative work. There are several core applications for different contexts, such as usual office work, web publishing, teaching and examinations, projects and cooperation, research, and literature. Even though there are a lot of semantic relations between the information in these databases, it is not possible to navigate between them or to identify knowledge structures.

As described in the above-mentioned practical cases, many organizations have a strong integration of groupware-based systems both in their IT-infrastructure and in their working environment. With growing organizational knowledge bases on the one side, and the lack of powerful, effective mechanisms and functionalities for navigating, linking, searching and investigating on the other side, the need for enhanced access mechanisms for exploring and using a complex information and knowledge offer is becoming more than evident.

Digression: The Design of Groupware-based Applications

For a better understanding, we think it is quite useful to shortly describe the basic design elements of a groupware-based application. All applications consist of one or more databases. These databases hold the data, logic, and design elements for the application (v. fig. 1).

The basic design elements of a groupware-based application are pages, forms, fields, and views (v. [Lotus 2001]). A *page* is a design element that displays information. Pages can be used anywhere in an application where text, graphics, or an embedded control are presented to users. *Forms*, like pages, display information. Everything that can be done with a page can be done with a form. The difference between forms and pages is that forms can be used to collect information. A form provides the structure for creating and displaying documents, and documents are used to store data in a database. The information collected in a form is saved as a document. Further, a form is used by a document as a template to provide the structure for displaying data. A *field* is the part of an application that collects data. Fields are created on forms in particular. Each field stores a single type of information, such as text, numbers, dates, or names. The information in a field is stored in an individual document. The contents of fields can then be displayed in documents and views. A *view* is a sorted or categorized list of documents. Views are the entry point to the data stored in a database. Every database must have at least one view, usually most databases have more than one view. [Lotus 2001]

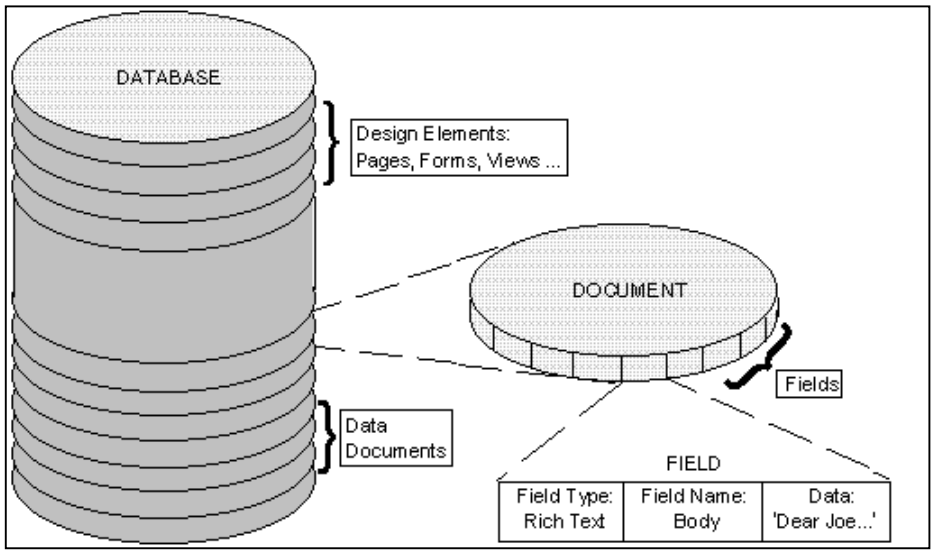

Figure 1. Design of a Groupware-based Application [Lotus 2001]

Apart form that, groupware-based applications consist of some more design elements that are not of interest in this context; they are therefore not described here. For a comprehensive overview, refer to Lotus [Lotus 2001].

3. Using Topic Maps in Groupware-based Environments

By applying topic maps, as defined in ISO/IEC 13250, to groupware-based organizational knowledge bases, it is possible to close a gap between knowledge and information. So far, views and categories in groupware systems have enabled the creation of little knowledge by bringing documents in different contexts. But this concept is limited to a poor identification of relations between them. Topic maps provide appropriate concepts to overcome this limitation and to build a semantic link network from the documents of a database. The basic design elements of a groupware-based application, especially the forms, can be used to identify the main subjects (topic types) and relationships between them (association types) (v. fig. 2).

While text analyzing methods with the support of artificial intelligence concepts could provide more or less reasonable results in identifying the main topics of a document, we have chosen another approach in the K-Discovery project. To identify the relevant subjects of documents, the fields of the forms must be considered. As mentioned in the earlier described scenarios, in many groupware-based systems the information and knowledge objects, e.g. documents, contain many fields of various kinds and purposes: obvious information, like the parts of an address (e.g. city), or the category of a report (e.g. balance sheets 2000), or keywords and linking information (e.g. persons, locations, time, free or given keywords). These fields form a set of potential candidates for topic types. Moreover, by looking at the forms as a whole and at the views, a lot of basic associations can be found between topic types. For example, in the context of an address form, the associations “person *lives in* city” or “person *works for* company” can be identified. Similarly, a report form contains associations like “report *is composed by* person” or “report *belongs to* category”. When considering views, fields of forms are displayed in columns. These fields form a base for additional associations, like

"report *is in work by* person" in a workflow view. Obviously, the different forms, i.e. the different document types, can be regarded as occurrence roles, while documents are occurrences in a concrete topic map.

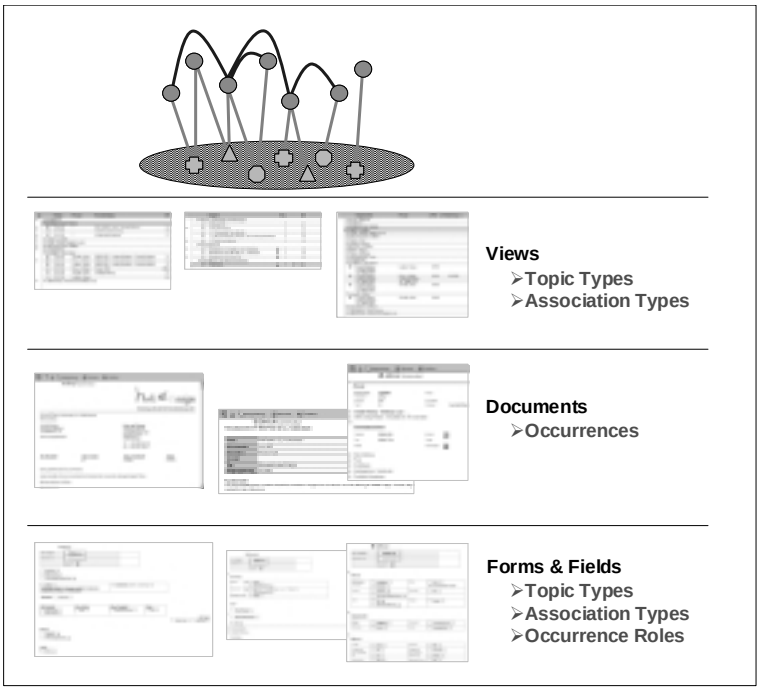

Figure 2. Concepts of Topic Maps and Groupware

To sum it up, applying topic maps to groupware-based knowledge bases provides a huge potential and a lot of advantages for the organizational knowledge management. Seen from the other side, groupware-based systems leverage the benefits of topic maps through their basic concepts, like distributed database architectures and replication, workflow management, or security and access control mechanisms [Nastansky et al. 2000]. A groupware-based environment enables the definition, generation and maintenance of a topic map in a consistent manner, on both a stationary and a mobile workstation. Thus, the distributed management of a topic map is independent of time and location. Differentiated access rights can be defined for a certain set of information objects, single information objects, and any part of an information object, as well as for functionality and presentation elements and design elements (like forms and views). These mechanisms enable the realistic adaptation of access structures against the background of complex and real organization models. This adaptation is supported by the assignment from persons to abstract organization and structure entities, called roles and groups. Thus, the different tasks, steps and skills involved of managing topic maps can be modeled with these roles and groups of a groupware-based system. As an example, some organizational members form an editorial team, which ensures that all published components of a topic map like topics, associations or occurrences adhere to organizational standards. Another key concept of groupware-based systems, the workflow management, can be used to describe and support the whole process chain of topic map publishing. We can realize that in this example two workflows (as described in the first scenario in Chapter 1) can be established, both to control publishing dates and periods, and to control the published contents.

The described concepts and functionalities of groupware-based systems make possible the team-based and organizational wide use of topic maps. Both the groupware-based systems and the concept of topic maps leverage the mutual potentials.

4. The K-Discovery Architectural Model

From the previous chapter, we have learned that the organizational knowledge management benefits on one hand from applying topic maps to groupware-based systems. On the other hand, these groupware-based systems provide an excellent environment for the integration of topic maps. In subsequent paragraphs, the basic architectural model of the K-Discovery project is introduced. The main purpose of the K-Discovery architectural model is to explain how functionality can be provided. The aim of this model is to specify separated modules that match the various requirements needed to gain the benefits addressed with the ideas of K-Discovery.

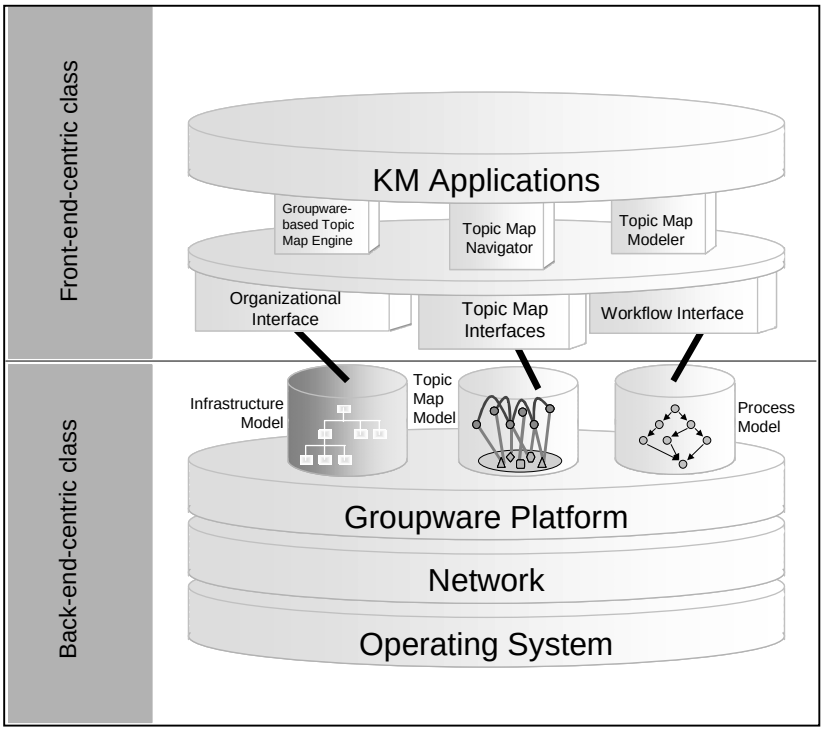

Figure 3. Architectural Model of K-Discovery

To keep the advantages of a flexible, structured and well-defined system, the architecture of the K-Discovery system is subdivided in several modular layers (v. fig. 3). These layers can be categorized into two main classes by considering their intended functionalities and purposes: the back-end-centric and the front-end-centric class. The back-end-centric class contains the technical infrastructure, and the data and information repositories. In contrast, the front-end-centric class encloses all the applications and tools, helping end users to interact.

The four layers of the back-end-centric class perform basic functionalities, like the communication and linkage of computer systems, and provide the groupware functionalities described in Chapter 3. Moreover, the repositories for the elements of a topic map (for a detailed description see Chapter 5), the organizational structure, and the process model are placed in this class.

The upper main half, consisting of three layers, forms the class of front-end-centric functionalities. The first of these layers provides several interfaces to the repositories. Their services are used by a set of tools which refer to the management of topic maps. First, the Groupware-based Topic Map Engine (GTME) allows the definition and creation of topic maps. In particular, topic types and association types can be defined. Based on these definitions, a topic map can be created automatically. Second, the Topic Map Navigator (TMN) provides functionalities, especially for end users, to navigate and to search through a topic map. Third, to modify an existing topic map, the Topic Map Modeler (TMM) offers the necessary capabilities. Because of the modular and open architectural approach, some of these tools are exchangeable, e.g. instead of the Topic Map Navigator described in the following chapter, a different product, like the Ontopia Navigator, can be integrated (v. [Ontopia 2001]). Several knowledge management applications can be developed upon these tools to support the organizational development and identification of knowledge. For example, consider the repository of processes. By applying a topic map to the set of run workflows, it is possible to identify relationships between them, as well as implicit and inherent process knowledge. The resulting knowledge can be used in two ways: First, to group similar workflows and to create predefined workflows. Second, it can support the analysis of the process structure as well as the organizational structure.

5. Implementation Approaches

In the following paragraphs, we describe and evaluate two approaches that implement the core functionalities of the introduced architectural model. The first approach is completely self-contained in a groupware-based framework; whereas the second approach uses a relational database management system as a repository for the objects of a topic map.

As pointed out, the first prototype is completely implemented in a groupware-based framework. A Lotus Notes database functions as both an environment for generating and configuring a topic map and a repository for the objects of a topic map (v. fig. 4). To determine and to reference the organizational knowledge bases to which a topic map should be applied, database connection documents are created (in the groupware-based environment). They define all necessary information to locate these knowledge bases, e.g. the server, the location on the server, and an unambiguous identifier (the “ReplicaID”). Topic types and association types are also defined in documents, each of which consists of an identifier and a base name. Based on these definitions software agents that can be started manually or scheduled create a document for each topic and for each association. In addition to an unambiguous identifier and to a base name the topic documents contain the topic type and the occurrence of the topic. Associations are described through an association type and the involved topics. These topics are referenced by their association roles, which are again topic types, and by their unique identifiers. All these documents describing topics and associations are displayed in a view. This view is embedded in a page and treated as XML (Accessed through a web browser this construct would form a XML document).

A Java servlet provides the navigation functionality and the front-end design (v. fig. 4). This Java servlet retrieves both the topic map as a XML document and the XSL documents which define and control the appearance in the web browser. These XSL documents and all other resources needed for the visualization, e.g. HTML documents, reside in the groupware-based environment.

Figure 4. Configuration Environment and Front-End of the first Implementation Approach

This implementation approach gains several benefits from the underlying groupware-based environment. Because all topic and association documents, resources as well as configuration documents reside in the Lotus Notes database, the full concepts of security and access control, replication, and workflow management can be applied (v. Chapter 3). As pointed out in Rath/Pepper [Rath/Pepper 1999] topic maps can be considered portable semantic networks. This portability is directly supported through the mechanisms of replication. Thus, a topic map or parts of it can be managed and used in a distributed way and even on a local system.

Nevertheless, some shortcomings have to be taken into consideration. In general, topic maps can consist of hundred of thousands of topics and associations. A lack of performance emerges by creating a document in the groupware-based environment for each topic and for each association. The performance of the accesses to documents and views decreases and the management and usability weaken because the groupware-based architecture is not designed for such a quantity of documents. Furthermore, in the current implementation of the described prototype, the response time to a user interaction is not optimal because of some executive code in the XSL documents as well as a customized LotusXSL processor.

To overcome these shortcomings, a different implementation approach has been chosen. In contrast to the first prototype, a relational database management system (concrete: an Oracle 8i database) functions as a repository for the objects of a topic map instead of the groupware-based environment. The generation and configuration of a topic map still reside in the groupware-based environment (v. fig. 5). Analogous to the first prototype, documents are created to define database connections, topic types, and association types. Connections to any groupware-based organizational knowledge base can be established independent of their

location; specifically, they can be hosted by different servers. The definition of topics is supported by *topic rules*. Several parameters and even the possibility of using programmatic code – the formula language – enable the exact, intuitive, and easy definition of topics. Similar to these topic definitions, topic associations can be specified. Within *association rules* the involved topics and the relationship between them can be defined again by using several parameters and programmatic code. The creation of topic rules and association rules is supported by software assistants, and context sensitive dialogues that directly access the definitions of topic types and association types, as well as the designs of the knowledge bases, especially the forms. Based on the definitions in the groupware-based configuration and management environment, software agents create and update tables in the relational repository for the objects of a topic map. They make possible the incremental maintenance of the data in the relational repository, i.e. not the complete data is updated each time changes are made; only the actual additions or deletions are carried out in the repository. The software agents and the used SQL statements reside in the groupware-based environment.

Figure 5. Configuration Environment and Front-End of the second Implementation Approach

Like the first prototype, a Java servlet provides the navigation functionality and the front-end design (v. fig. 5). This Java servlet retrieves the topic map data from the relational repository by using SQL statements. All resources needed for the visualization, e.g. the Java servlet, HTML documents, image resources, are stored on a server.

Again, the second implementation approach gains most of the above-mentioned benefits, like applying the concepts of security and access control, replication, and workflow management. Furthermore, the above-

mentioned lack of performance is eliminated because of the usage of a relational database management system as a repository for the objects of a topic map. The time exhausting data lookups are moved from the groupware-based repository to the relational repository.

6. Conclusions

An architecture of a groupware-based knowledge management system based on the ISO standard ISO/IEC 13250 TOPIC MAPS has been presented. The K-Discovery project combines groupware paradigms and topic maps, leading to substantial synergies. These leverage their mutual advantages and therefore provide benefits for both research fields. From our point of view, the K-Discovery project provides a considerable framework to identify knowledge structures in groupware-based environments. In fact, there are already some companies that have shown interest in such a solution.

The work on the first implementation approach has been discontinued due to the above- mentioned shortcomings. However, work on the second implementation approach continues. Prototypes of the core modules are being implemented and parts can already be used for trial purposes. The finalization of the system integration is planned for the near future. In detail the implementation of the Groupware-based Topic Map Engine and the Topic Map Navigator are almost finished. The conception and implementation of the Topic Map Modeler is about to begin.

Moreover, future works will focus on some further conceptions. For example, a concept of roles involved in the topic map publishing process and workflows, that this publishing process support, will be defined.

Eventually, knowledge management applications upon the Groupware-based Topic Map Engine will be developed. Specifically, a solution for the ad hoc analysis of workflows will be implemented.

Bibliography

[Ahmed 2000]

Ahmed, Kal: Topic Maps for repositories; XML Europe 2000, Paris, France, 2000.

[Bornschein-Grass 1995]

Bornschein-Grass, C.: Einsatzformen und betriebswirtschaftliche Gestaltungspotentiale von Groupware als Grundlage von Computer Supported Cooperative Work; Deutscher Unversitäts-Verlag, Wiesbaden, 1995.

[Deutsche Bank 2000]

Deutsche Bank AG: http://public.deutsche-bank.de/deuba/transaction/navigate.nsf/frameset/MPOL-4GLMLV?OpenDocument; Frankfurt, 2000.

[Güldenberg 1998]

Güldenberg, S.: Wissensmanagement und Wissenscontrolling in lernenden Organisationen: ein systemtheoretischer Ansatz, DUV: Wirtschaftswissenschaft, Edition Österreichisches Controller

Institut, Zugl.: Wien, Wirtschaftsuniv., Diss., 1996 2. durchges. Auflage, DUV, Deutscher Universitäts-Verlag GmbH, Wiesbaden, 1998.

[Lotus 1995]

Lotus Development Corporation: Groupware - Communication, Collaboration, Coordination; Cambridge, Massachusetts, USA, November, 1995.

[Lotus 2001]

Lotus Development Corporation: Domino Designer Help [Release 5.06a]; Database, Cambridge, Massachusetts, USA, 2001.

[Nastansky et al. 2000]

Nastansky, Ludwig; Bruse, Thomas; Haberstock, Philipp; Huth, Carsten; Smolnik, Stefan: Büroinformationssysteme, in: Fischer, J.; Herold W.; Dangelmaier, W.; Nastansky. L.; Suhl, L. (Hrsg.): Bausteine der Wirtschaftsinformatik, Erich Schmidt, Berlin, 2000.

[Ontopia 2001]

Ontopia AS: http://www.ontopia.net/solutions/products.html; Oslo, Norway, 2001.

[Pavone 2001]

Pavone AG: Enterprise Office Help [Release 4.6d, Build 320]; Database, Paderborn, 2001.

[Probst/Raub/Romhardt 1999]

Probst, G.; Raub, S.; Romhardt, K.: Wissen managen: Wie Unternehmen ihre wertvollste Ressource optimal nutzen; 3. Aufl., Gabler, Wiesbaden 1999.

[Rath/Pepper 1999]

Rath, Dr. Hans Holger; Pepper, Steve: Topic Maps: Introduction and Allegro; Markup Technologies 99, Philadelphia, USA, 1999.

[Schliwka 1998]

Schliwka, A.: Teamorientiertes, medienübergreifendes Know-how-Management in verteilten Umgebungen; Shaker, Aachen 1998.

[Wagner 1995]

Wagner, M.: Groupware und neues Management: Einsatz geeigneter Softwaresysteme für flexiblere Organisationen; Vieweg, Braunschweig et al. 1995.

Authors

Stefan Smolnik

Address: Warburger Straße 100, D-33098 Paderborn, Germany

E-mail: Stefan.Smolnik@notes.uni-paderborn.de

Phone: +49 – 5251 – 60-3375

Fax: +49 – 5251 – 60-3399

Biography:

Stefan Smolnik – was born in 1970. He received his computer science diploma from the University of Paderborn in 1998 and is currently both a PhD student and research assistant at the Business Computing Department 2 of the Faculty of Business Management, Business Computing and Economics at the University of Paderborn in Germany.

Ludwig Nastansky

Address: Warburger Straße 100, D-33098 Paderborn, Germany

E-mail: Ludwig.Nastansky@notes.uni-paderborn.de

Phone: +49 – 5251 – 60-3368

Fax: +49 – 5251 – 60-3399

Biography:

Ludwig Nastansky – is a professor of Business Computing at the Faculty of Business Management, Business Computing and Economics at the University of Paderborn in Germany. His teaching, research, and consulting focuses are: information & knowledge management, office systems, groupware oriented architectures & systems, groupware-based software development, workflow management, business-to-business office solutions, integrated intranet - extranet - internet system design, document management systems, hypertext- and multimedia-systems, project management and process enacting systems, business process re-engineering, electronic messaging systems, corporate client-server architectures, process-oriented controlling systems, virtual online environments for supporting teaching and learning processes; all activities in context of the Groupware Competence Center (GCC) at the University of Paderborn.