Organization Design as a Groupware-supported Team Process

GroupOrga — Participative and Distributed Organization Design for Office Information and Workflow Management Systems

Dissertation

Submitted to the Faculty of Business Administration, Business Computing, International Business Studies and Economics of the University of Paderborn

by

Dipl.-Wirt. Inform. Marcus Ott

Paderborn 1998

Organization Design as a Groupware-supported Team Process

GroupOrga — Participative and Distributed Organization Design for Office Information and Workflow Management Systems

Dem Fachbereich 5 — Wirtschaftswissenschaften — der Universität-Gesamthochschule Paderborn zur Erlangung des akademischen Grades eines

DOKTORS DER WIRTSCHAFTSWISSENSCHAFTEN

(Dr. rer. pol.)

vorgelegte **Dissertation** von **Dipl.-Wirt. Inform. Marcus Ott**

Paderborn 1998

This thesis is dedicated to Dörte
for all the support and encouragement she has given me.

There are not enough words.

ACKNOWLEDGEMENTS

To Dr. Ludwig Nastansky, Professor of Information Management, for providing the opportunity to do Ph.D. research at the University of Paderborn. Thank you for supporting this research and supervising this thesis.

To my colleagues at the Department for Information Management & Office Systems and the students of the GroupOrga project team for providing the intellectual foundation upon which I drew. Very often you helped me at one time or another to find some of the most obscure references. Special thanks for your fruitful discussions of the ideas and concepts presented in this thesis.

To Yuko Sakamoto for proof-reading, offering advice on style, and commenting on the dissertation. Modern information technology, such as e-mail and the Internet, as well as a good sense of collaboration and teamwork helped bridge the distance between California, USA and Germany.

To my parents for your love and best wishes. You provided me with the driving force and the support needed to complete my task. Thank you.

Finally, my most sincere thanks go to Dörte. Not only were you a great help, but you have stuck with me through all of this and I am deeply grateful. An uncountable number of thanks go to you.

ABSTRACT

This dissertation describes a vision, concept and prototype technology that supports organization design as an ongoing group process. In particular, it is concerned with the design of organizational models for office and workflow management systems.

The paradigm, which shares many of the change process attributes prescribed by the participative management and sociotechnical system design, defines organization design as an organic process potentially involving everyone in the organization and those significantly associated with the organization.

Such an understanding is promoted by a formal enterprise model which covers the different facets of an organization that can be modeled, such as the processes, the information, and the organizational structure. This model, representing the circumstances of an enterprise, tends to be large and complex. It should also be continuously evolving to reflect the dynamic nature of the enterprise.

GroupOrga also describes a prototype technology that supports organization design based on groupware technology. The technology consists of various integrated pieces, such as interactive, graphical tools for generating design descriptions, and multi-user, team-oriented database applications that are replicated between various locations.

CONTENTS

List of Illustrations

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AAA	Autonomous Administrative Areas
AAP	Autonomous Administrative Point
ACDF	Access Control Decision Function
ACI	Access Control Information
ACL	Access Control List
API	Application Programmers Interface
ASN.1	Abstract Syntax Notification One
BDL	Babcock Dienstleistungs-GmbH
BIS	Business Information Systems
BPR	Business Process Reengineering
CA	Certification Authorities
CCITT	International Telegraph and Telephone Consultative Committee
CIM	Computer Integrated Manufacturing
CPI	Continous Process Improvement
CSCW	Computer Supported Cooperative Work
DAP	Directory Access Protocol
DIB	Directory Information Base
DIE	Integrated Development Environment
DISP	Directory Information Shadowing Protocol
DISP	Directory Information Shadowing Protocol
DIT	Directory Information Tree
DN	Distinguished Name
DOP	Directory Operational Binding Protocol
DOP	Directory Operational Binding Management

DSA	Directory System Agents
DSA	Directory Service Agents
DSE	DSA Specific Entry
DSP	Directory System Protocol
DSP	Directory Service Protocol
DTP	Desktop Publishing
DUA	Directory User Agents
EDI	Electronic Data Interchange
EER	Extended Entity Relationship
EKB	Enterprise Knowledge Base
ER	Entity Relationship
GEIMM	GroupOrga Enterprise Information Management Model
GI	Gesellschaft für Informatik
GMD	Gesellschaft für Mathematik und Datenverabeitung mbH
GoM	Guidelines of Modeling, also: Grundsätze ordnungsgemäßer Modellierung
GroupOrga	Groupware-based Organization Design
GUI	Graphical User Interface
HDI	Hirn-Dominanz-Instrument
HICSS	Hawaiian International Conference on Systems Sciences
HOB	Hierarchical Operational Binding
IAA	Inner Administrative Areas
IAP	Inner Administrative Point
IETF	Internet Engineering Task Force
IOS	Integrated Office Systems
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
IT	Information Technology
ITHACA	Integrated Toolkit for Highly Advanced Computer Applications

ITU-T	International Telecommunication Union-Telecommunication Standardization Bureau
JVM	Java Virtual Machine
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LRDM	Logical Relational Design Methodology
LS:DO	LotusScript Data Objects
LSX	LotusScript Extensions
MS	Microsoft
Notes FX	Notes Field Exchange
ODBC	Open Data Base Connectivity
OLE	Object Linking and Embedding
ORM	Organization Resouce Management
PD	Participatory Design
R&D	Research and Development
RADD	Rapid Application Development and Deployment
RDN	Relative Distinguished Name
SNI	Siemens Nixdorf Informationssysteme AG
SOB	Shadow Operational Binding
TQM	Total Quality Management
WAN	Wide Area Network
WET ICE	Workshop on Enabling Technologies, Infrastructures for Collaborative Enterprises
WfM	Workflow Management
WfMC	Workflow Management Coalition
WfMS	Workflow Management System
WI	Wirtschaftsinformatik
WWW	World Wide Web

Chapter 1
Introduction

> *Organization modeling* is concerned with describing the *structural* aspects of an organization. It describes the different parts of an organization, how these parts are related to each other and their properties ([Li/Lochovsky 1996], p. 193).

Organization design or *organization modeling* is the branch of management that addresses problems of inefficiency in organizations. It is the design of the structure of the organization, using the term organization in the widest sense. But it is more than the design of the pattern of positions and functions often described as the *organization structure* and usually recognized in the organization chart; it is also the design of the organizational work relations in the business processes, the organizational members who carry out the work, and the information used in these organizational processes.

Information is vital for an organization to achieve flexibility and integration in its organization design. A successful enterprise is a knowledge-based organization with the infusion of information technology. However, computer and communication technologies will add to the complexity of design processes if they are treated as the cure-all to an organization's problems.

The focus of this dissertation is an innovative and integrated approach to organization design and the information technology which supports this approach.

If organizations are to remain effective, they must change eventually. These changes must be by design, not by chance. Three things are needed to achieve this:

- The basic features of organizational and human behavior must be identified and understood, since these determine what can and cannot be designed. Creating new organization designs or planning re-organization is useless without identifying what is needed.

- The ability to communicate accurately what the organization design means to the workers.
- An awareness of all outstanding factors in the organization and its situation, not just those factors that are explicit or troublesome.

Current demands for a break with functional organizational structures, and the separation from strongly hierarchical forms have resulted in many recommendations for new organizational structures. These proposals all aim at quick response capability and flexibility. This removal of hierarchies is explained by one of the offerings in information technology (IT). Process-supporting IT allows for an organizational structuring that is guided by an organization's processes and procedures. The two characteristics of an IT implementation that reduce the workforce and shift competence and responsibility to lower organizational levels are:

- Vertical and horizontal integration of separated activities
- IT-supported task completion

However, this process orientation in an organization requires structural reorganization. Therefore, a comprehensive concept for the modeling and design of organizations is required. But although this goal is well known, there is a lack of vision and concrete models for future forms of organization design processes. Current research in the field of workflow management and the design of office information and office communication systems focus primarily (and sometimes exclusively) on organizational processes. Thus, these results stem from a process-oriented point of view and the research does not extend to the organizational structures of employees, units, workgroups, and roles.

Hence, the body of research needs to include a combination of considerations on organizational structures using current developments in business process reengineering and workflow management approaches. This project addresses this issue and discusses an innovative concept and framework.

This chapter introduces the scope and objective of the dissertation in section 1.1, presents its embedment in a comprehensive workflow and office product set in section 1.2, and gives the organization of the rest of this research in section 1.3.

1.1 Scope and Research Objective of this Dissertation

The subject of this dissertation is a visionary paradigm for IT-supported and team-based organization design. The vision, which shares many of the change process attributes prescribed by participative management and sociotechnical system design, defines organization design as an evolutionary change process potentially involving everyone in the organization and those who are associated with it. This approach is presented in contrast to the traditional, academic view of organization design activity that involces only high-ranking people in the organizational hierarchy. The project describes a group process for organization design in which the members use a computer-based technology. Moreover, it presents the prototypical organization design system that was developed as a testing environment.

The term *design* is frequently understood as *design-from-scratch*. In this research it should always be understood as *re*-design, since few organization design process are started from scratch. In the field of *Business Process (Re-)Engineering* (BPR), organization design is presented as a powerful means to improve organizations, which is why this research often refers to redesign, not design-from-scratch. However, although this focus is set, there is no reason why the approach presented should not be applied to any design-from-scratch situation. The difference is that there is no given context which has to be considered, and that issues of transition from one design to another are irrelevant.

The research field of organization design also addresses the question of how change processes are accepted. Although change acceptance is important, and one may question if the proposed paradigm itself can be accepted, it is not the primary focus of the project. Rather, the approach is based on the assumption that the organization design process laid out here, which integrates everybody in the process, can *itself* contribute to a growing *acceptance in the redesign*.

Theory on organization design covers two distinct fields: processes and structures. This research deals primarily with the organizational infrastructures and attempts to identify the peculiarities of its design. When necessary, adequate introduction into process management is given and appropriate references are named.

Another important aspect in literature is *organizational learning*, which observes organizational peculiarities on a macro level. The approach in this project can be seen as an organizational learning processes on a micro level, since in this concept, organizational members are explicitly involved in the design and learning process.

As noted above, this study is mainly concerned with possible ways to support the organization design process and to better understand the involvement of the people in an organization undergoing this process. This new approach can serve as a partial solution for a computer-based system that supports work in the problem domain of organization design. Since this project is not predominantly concerned with how to *implement* this new approach, it proposes only one implementation environment. Nonetheless, the proposed concept is not restricted to this particular system implementation, and can be implemented in other technologies, as well.

Using these considerations as a starting point, this research presents the concepts of an organization design framework and process in cooperative environments. The main objectives are:

- Revelation of practical problems and conceptual challenges in traditional approaches to organization design; especially in combination with current developments of office information and workflow management systems.

- Development of a comprehensive enterprise model for the representation of organizational circumstances and design, based on various sources.
- Presentation of theoretical solutions and innovative practical concepts for everyone in an organization to be able to create and design this model.
- Partial implementation of the resulting concepts and models in a prototype environment, the GroupOrga system, as a demonstration of a groupware-based organization design system.
- Description of an evolutionary design process that uses current IT with the aim of continuous, parallel design at various locations and by everybody in an organization.

All parts of the project are based on practical case examples of organization design. In addition, results of empirical studies in different organizations are presented. The GroupOrga system is an integral part of the project and an important outcome of the practical conception and implementation of the dissertation project.

Organization design is an increasingly important subject. However, this study is not a comprehensive thesis of design; it is a practical one. It is based on experience, observations and discussions during the conception and development of various workflow management platforms and solutions.

The research does not recommend a design for an organization for implementing particular instances of workflow management environments. But it does describe the basic features of organizations, and features that are common to certain types of organizations. Every organization is different from other organizations, but each has certain basic features that may be recognized and designed. This must be taken into account whe making decisions about organizational structure and change. Because each organization is unique, the solutions, modeled in the GroupOrga system, are also unique.

Nevertheless, the research indicates what factors have to be considered, and how they interact. In other words, it gives a framework for design processes. The presented enterprise model shows a methodology of design and a way of recognizing and thinking of all the features that are basic to all organizations. It will not show what changes are needed in a specific organization. Organizations are too complex for that. The research presents the main issues by questioning organization design.

1.2 Embedment in Existing Workflow and Office Projects

GroupOrga was born from a dissertation project as a conception of a distributed organization design environment. The basic scientific research, as well as the first system design and implementation was conducted at the Groupware Competence Center at the University of

Paderborn, Germany. In a parallel phase, early results and prototypes were adopted by Pavone Informationssysteme GmbH, a Paderborn software firm and management consultancy. During this time, the preliminary results were developed into concrete products and solutions, which have been marketed nationally and internationally since the spring of 1997. Since then, the GroupOrga prototypes have been combined and integrated with a groupware application environment for workflow management and with an extensive application for distributed office management in teams being developed by Pavone Informationssysteme GmbH.

In addition to the development of products from GroupOrga prototypes, Pavone Informationssysteme GmbH is currently developing a full consultancy service in the field of organization design with groupware technology. These services include consultation, implementation, education, and training, as well as a complete integration of groupware-based organization design environments in existing IT infrastructures.

Due to this background, the project covers diverse aspects. Hence, the focus of this dissertation is a wide presentation of the research project in its entirety. When needed, a detailed explication of conceptual or technological details is presented. It is referred to the substantial, corresponding project documentation and tool manuals ([Liebrand 1995], [Meyer 1995], [Heinz 1995], [Meyer 1996], [Müller 1997], [Heinz 1997], [Hoischen/Otto 1997], [Huth 1998], [Matysczok in prep.], [Brunner in prep.]). In contrast to these references, which form the basis for parts of this research project, this dissertation does not focus on details, but rather on a conceptual level.

1.3 Organization of this Dissertation

This dissertation is organized into two parts: a conceptual and a technical documentation. The purpose of the first part—which is this part—is to present the primary research contribution, which addresses the design of organizations as a team-based approach. This part clarifies terms, discusses the problem field, and proposes a solution, in conceptual and practical respect. The purpose of the second part is to present secondary information on the research contribution. It provides information and technical references on various aspects of the solution.

Chapter 2 begins with a review of the literature to establish the distinguishing characteristics of the problem domains that are of interest in the project. Definitions and explanations of relevant terms are developed.

Chapter 3 reviews traditional theoretical and practical approaches to organization design. Technological criteria is investigated and established for a definition of the problem domain of the project. The purpose is to explain how IT is connected to the organization and how it can be (but is not yet) used to improve the effectiveness of organization design. Next, the limitations of some theoretical approaches are described and the causes for their failures are discussed. A case is made for re-thinking the scope of traditional approaches, as opposed to fixing their problems.

Chapter 4 surveys existing conceptual enterprise models and evaluates the criteria. The comprehensive conceptual data model GEIMM is presented and discussed. The presentation highlights the new concepts and distinguishes them from those borrowed from existing approaches in this field. The conceptual data model is evaluated against the same criteria that is used to evaluate traditional concepts of organization design, which provides theoretical evidence for its usefulness.

Chapter 5 presents the concepts and an architecture for the realization of an organization design application environment. It describes the basic features of the new paradigm for organization design. Distribution in the design process and the design of an enterprise knowledge base, which is the foundation of the prototype system *GroupOrga* is discussed further. The integration of GroupOrga with existing applications for office and workflow management concludes chapter 5.

Chapter 6 presents a meta-process for organization design in teams. This chapter demonstrates how the defined process steps can be mapped with specific tools in the GroupOrga system. Moreover, it assesses the concept in the real world by describing relevant episodes from several case studies of concrete organization design processes.

Chapter 7 summarizes the findings of the GroupOrga project. It also describes new directions in organization design. Current issues in IT and organizational theory are discussed, and future research paths are described.

Supporting material for the project are cited in appendices A to E which are combined in the second part.

Chapter 2
Fundamentals and Terms

This chapter presents the foundations for the GroupOrga project, selecting individual elements from various sources to allow for their critical appreciation and a subsequent problem definition in chapter 3. Moreover, it outlines essential theories and conceptual approaches, as well as their practical realizations.

Section 2.1 offers a clarification of relevant concepts and definitions for specific terminology used throughout the project. Section 2.1.1 focuses on the team as an important element of organizations, sections 2.1.2 and 2.1.3 describes groupware as a supporting technological means for teams and learning organizations. In section 2.1.4, the office as an organizational element, as well as office management systems, their goals, intentions, and restrictions is examined. Section 2.1.5 addresses workflow management and corresponding systems. Concluding, section 2.1.6 describes organizations and the necessity of their design.

Using the theoretical framework established in section 2.1, section 2.2 discusses practical implementations and presents a product for groupware (section 2.2.1), office management (section 2.2.2) and workflow management (section 2.2.3).

2.1 Clarification of Relevant Concepts

Roithmayr ([1996], p. 104) stresses that the field of Business Computing is a young discipline, which is why it does not have a clear framework of accepted terms and explanations. Hasenkamp and Syring also noticed the lack of homogenous vocabulary in this field of research ([1993], p. 406).

Therefore, before going into detailed aspects of the GroupOrga approach of organization design, the terms in this thesis must be clarified and defined. These terms are used frequently in the following chapters and have to be formulated clearly since they are the basis for the logic of the following model and method of organization design.

This section defines the concepts of *teams and teamwork*. It focuses on the current *groupware technology*, comments on *office systems* and *workflow management* and *systems*, and investigates the aspect of *design in organizations*. Other terms are explained or defined when introduced.

2.1.1 The Team and its Work: Teamwork?

Katzenbach and Smith ([1993], p. 70) describe a team as a small group of people whose abilities complement one another and who work for a common goal. The people combined in such a team have a shared objective. An earlier investigation revealed that the trend towards teams in organizations is stimulated by numerous developments in businesses: fast-moving markets, a trend towards lesser hierarchies, team-based performance ratings or reports about role model organizations with massive team-orientation (see [Ott 1997a], p. 91).

Teams can present another form of workforce parallel to the traditional concepts of hierarchy and performance. Teams integrate and promote formal structures and processes. Hierarchical structures and the underlying processes are essential in large organizations, but they are not threatened by teams. In fact, the deployment of teams is a good way to bridge structural borders and to design an organization's core processes. When teams are seen as a complete replacement for hierarchical structures, their real potential is misunderstood.

In contrast to a group, a team cannot independently establish and break itself up because it is integrated in the organizational structure and was created in order to reach a given goal. Consequently, a team is understood as a formal group. In the English-language literature, the terms "team" and "teamwork" are often implicitly connected with the co-working of specialists or employees of higher rank. As a result, "groups" and "groupwork" are generally comprised of workers on the lower levels of hierarchy. The GroupOrga project will not differentiate in this way since the work of teams or groups often spans several hierarchical levels due to the trend of lesser levels of hierarchy. In contrast, *informal groups* are created because some human needs remain unfulfilled at the workplace ([von Rosenstiel 1978], p. 245). Hence, a team is a group that shows organizational characteristics, such as job descriptions or job relationships (e.g. supervisor/employee). This explains the widespread use of the term workgroup.

When assigned to a project, team members are given their own tasks. These tasks are coordinated, directed and completed to achieve a common goal. This is *teamwork*.

2.1.1.1 Organization of Teamwork

Each team can solve the assigned task by itself, with each team member deciding how to complete the tasks. In order to have this freedom, the team must autonomously allocate the sub-tasks, complete the sub-tasks, and integrate the results. In order to meet these requirements, team members must assume a partial autonomy of their freedom of action. Partial autonomy of team members is a substantial characteristic of team-based organizational forms.

Seitz, Galster and Lang [1993] distinguish different forms of organization of work within teams (see Figure 2-1). On the one hand, they identify the individual work of a single person, and on the other hand they identify the collaborative work of many people (*cooperative teams* and *collaboration in organizational groups*). While the former type of teamwork implies a deliberate intensification and regulation of group processes, the latter results in an interplay of its members, which is not directly imposed by the supervising organization.

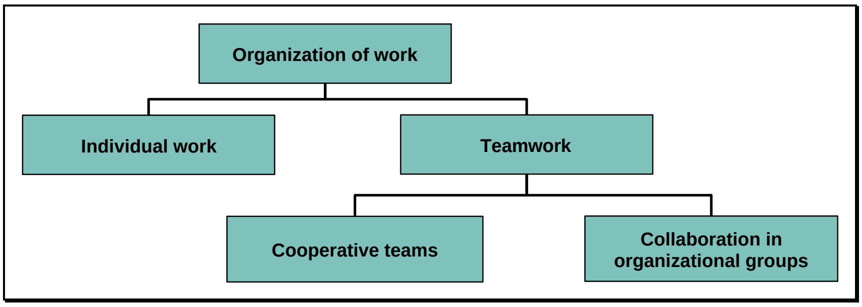


Figure 2-1: Different forms of organization of work within teams

A further differentiation of teamwork can be made when considering the lifetime of the teams performing the teamwork. Forms of teamwork that are limited in time, such as a research project, are characterized by their existence for a certain period only. They exist parallel to actual organizational forms. Typical representatives are *quality circles*, *project groups*, and *task forces* (see [Bungard/Antoni 1993], p. 22). In contrast to these short-term teams, long-term teams are integrated into the existing organizational structures. A representative of this type is the *autonomous workgroup*, which is found in the manufacturing industry. This type of team is a functional unit with 3-10 people within the regular organizational structure. They continuously work together in order to produce a complete product or service (ibid., p. 23).

Exceptional interest for teams and the organization of teamwork grows out of the co-existence of organizational restructuring and the introduction of productive, team-supporting IT such as groupware platforms. The growing readiness for a restructuring of conventional organization can be noticed. Coupled with the introduction of innovative IT, it allows for the realization of computer-supported teamwork ([Wildemann 1995], p. 10).

2.1.1.2 Computer-supported Teamwork

Owing to the rapid progress in technology, the provision of powerful computer networks and distributed information and communication technology can be realized inexpensively even for small units of the organizational structure. While the personal computer has been used for supporting individual work, when combined with network functionality, it can be used to integrate the single person's results in the overall group context. In order to promote this development, group members must have easy access to all information and results within the group, and they must be able to communicate easily. Accordingly, IT moved into a new function in the framework of computer-supported teamwork; next to being a tool (*instrumental aspect*) it grew into the function of a new communication medium (*medium aspect*), which could connect the singular tasks into an electronic workflow.

To reach these requirements, how teams complete work and how information technology can support this work, must be studied (see [Bannon/Schmidt 1997]). Conversely, the effects of information technology on teamwork and on changing organizational structures are another point of investigation.

A particular research field, which subsumes most of the above aspects of IT and which focuses on computer support of cooperative teamwork, is *CSCW* (Computer Supported Cooperative Work) or *Workgroup Computing*. For information on CSCW and Workgroup Computing, refer to [Bannon/Schmidt 1991], [Bowers 1994], [Hasenkamp/Syring/Kirn 1994], [Hummel/Schoder/Strauss 1996], [Nastansky 1993b], [Rogers 1994], or [Turner/Kraut 1992]. For the last decade, this discipline developed computer applications that support communication and shared data-management for team members who work on a shared project. This opens the way to make dispersed information sources available to all the people involved in a distributed process environment and to facilitate individual coordination. *Groupware* supports this functionality through computing, thus requiring special hard- and software. CSCW is the field that studies the use of this technology, whereas groupware is often used to denote the technology that people use to work together. *Groupware* as a special type of computer support for teamwork is discussed in section 2.1.2.

2.1.1.3 Communication, Collaboration, and Coordination for Teamwork

Teamwork in organizations, which is characterized by social and functional relations, is based on the organization members’ ability to communicate with each other, effectively coordinate the work and collaborate in work processes. Although *communication*, *collaboration* and *coordination* are key aspects in the field of CSCW and Workgroup Computing, they are not confined to CSCW research.

As a first and rather simple approach to this intention, communication can be understood as distributing information in an organization, collaboration as sharing information and building a shared understanding, and coordination as assisting individuals and groups in the adjustment of complex tasks involving a rich combination of delegation, sequential tasks, and forwarding.

Communication

"Communication is the transmission of knowledge" ([Lotus Dev. 1995], p. 7). It comprises the processes of transfer or exchange of information between the members of an organization, or between them and their external partners, and the preceding and succeeding tasks.

Using characteristics of communication, a *communication system* can be described as a passive electronic medium for transmitting information between the partners. Several variables can be used to distinguish and classify communication systems, such as time, place and the number of participants in the communication process (see Figure 2-2). Differing types of communication can be discriminated into the same or different place, the same or different time, and one or many contributors to the communication.

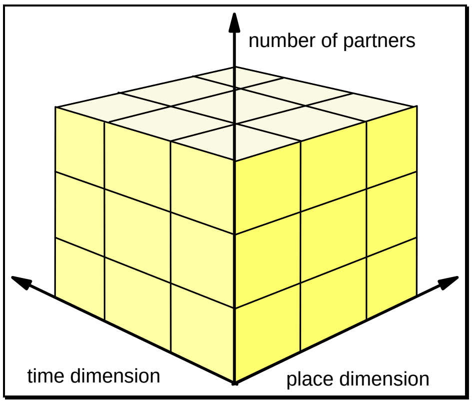


Figure 2-2: Dimensions to distinguish and classify communication systems

Same-time, same-place and one-to-one communication represents a case where communication technology is hardly useful, whereas for the most complex combination—different time, different place and many partners—CSCW technology fits well. When isolating communication from a system view, electronic messaging can be identified as a technology for communication in teams. It is the store-and-forward transport of information among people in groups which supports different-time and different-place information sharing through a "push" model, that is, information is "pushed" from the sender to the recipient.

In this project, communication will be viewed as the cornerstone of any type of teamwork. Only the ability to communicate effectively can be taken as the prerequisite of collaboration and coordination, since each collaboration is based on "coordinated communication".

Collaboration

Dhar and Olson [1989] use the term collaboration "to refer to a goal-oriented process involving contract definition and execution among two or more individuals" (p. 34). Hence, collaborative work comes about, when tasks for completion of a product or service are carried out by several people. The necessary relations for this collaboration are planned and may be predetermined by the product or service characteristics.

Apart from collective work in teams, collaboration can come about in other forms, as well. The term *distributed collaboration* (see [Bannon/Schmidt 1997]) describes the case where the partners do not interact directly. In this indirect model, the participants do not always communicate personally, but use communication systems to interact and to adapt their personal behavior to the common task. In addition, collaboration is not bound to the physical borders of organizations. It is characterized through collaborative behavior as such, which can involve partners in various organizations and locations.

Difficulties in defining the term collaboration are elaborated by Dhar and Olson when they identify three influencing factors to collaborative work: uncertainty, complexity and ambiguity. With *complexity* they describe the problem of mapping the necessary activities in collective work with the resource requirements associated with these activities and the complexity from an individual's perspective to be involved in several projects simultaneously. *Uncertainty* refers to the lack of knowledge about what environmental states will prevail, and to time estimates of project activities that the individual is involved in. *Ambiguity* refers to the fact that the collaborative activities may not be well defined. This may be the case, particularly in the early stages of a collaboration.

For collaboration purposes electronic messaging falls short in many respects: it is not tuned to the needs of many-to-many interaction, it gives no assistance for the tracking of information, and it does not support the maintenance of discussion threads. A shared database approach provides many advantages over a model based on messaging. Primarily, information is pulled as needed, as opposed to information being pushed, as in the case with messaging.

Coordination

Coordination of teamwork aims at the control and guidance of working together. Generally, two types of control can be identified: *self-control* through the members of the team, based on organizational rules and regulations, and *outside control* through supervisors. Both kinds fit to a definition given by Malone and Crowston ([1990], p. 361), who see coordination as "the act of managing interdependencies between activities performed to achieve a goal." The authors provide a list of elsewhere suggested definitions of coordination (p. 366). Thus, coordination is the necessary and conscious arrangement of decentralized actions and decisions of organizational entities. The central aim of coordinating these actions is the optimized fulfillment of organizational goals.

Coordination becomes necessary when a comprehensive task is divided into single work steps whose participants might collide in their interests for scarce resources or because of mutual dependencies regarding amount, quality or completion time of the team-product or team-service. Appropriate coordination is employed as a means to guarantee that the single activities complement one another and contribute to the overall task without a loss in effectiveness due to social or technical conflicts. For instance, coordination can prevent redundant completion of tasks, ensure proper quality standards of output from earlier phases of a process, and endorse completion of preceding tasks on schedule.

Whereas collaboration is relatively passive from the system view, coordination is active, and specifications how activities have to be accomplished by means of the system are given. Discussions on collaboration focus on how groups of people communicate in order to share information and leverage knowledge that helps them perform jobs more efficiently and effectively. Much of this interaction is characterized by its ad hoc nature. Coordination, in contrast, defines particular sequences, structured processes and expected time constraints for people completing a set of tasks. To a great extent, coordination technology has been the domain of workflow automation systems which concentrate on highly structured processes that can be expressed by pre-defined, conditional workflows based on status and restrictions.

Since coordination of collaborating partners can be found in various disciplines such as social sciences, technical fields, and information technology, an approach to interdisciplinary study of coordination was proposed ([Malone/Crowston 1994]). Four *basic components* for coordination purposes are identified: the *goals* of a coordination, its connected *activities*, the *actors* assigned to the activities, and the *interdependencies* between the previously named activities (see [Malone/Crowston 1990], pp. 360f.).

A Layered Approach to Communication, Collaboration and Coordination

Due to the complex semantics of the three terms, there is no widely accepted relationship between these terms. Nevertheless, a layered structure can be identified (see Figure 2-3).

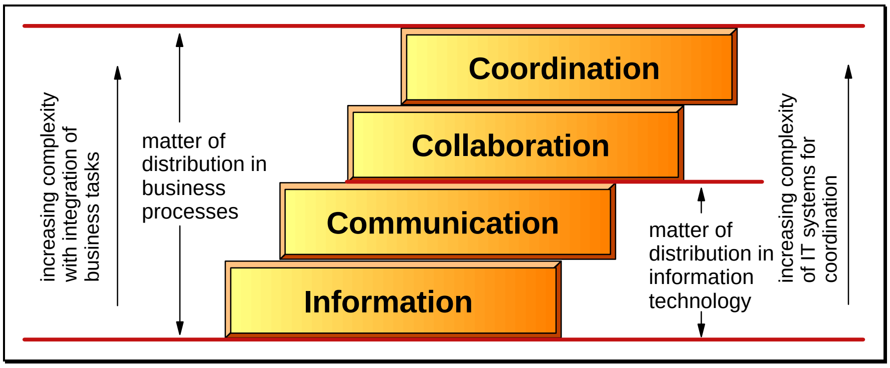


Figure 2-3: Layered structure of communication, collaboration and coordination

In attempting to characterize different teamwork processes more precisely, Malone and Crowston proposed to describe them in terms of successively higher layers, each dependent on the layer below it. The order of their layers is based on the observation that most of the coordination processes need decisions that are carried out by the team. Such considerations concern process selection or the managing of resource and timing constraints based on relevant information sources. Team decision-making requires communication between the members of the team, which, in turn, requires an environment to collaboratively manage and document the processes.

With increasing complexity of the problem domain, as well as the supporting system, the need for coordination increases. This explains the positioning of this term at the top of the framework (see [Malone/Crowston 1990]).

2.1.2 Innovative Groupware Systems

As was discussed in the previous section, any type of application that is used for the support of teamwork requires functionality for the communication of the team members, their collaboration and the coordination of their respective tasks. Accordingly, Ellis, Gibbs and Rein [1991] state: "The goal of groupware is to assist groups in communicating, in collaborating, and in coordinating their activities. Specifically, we define groupware as: computer-based systems that support people engaged in a common task (or goal) and that provide an interface to a shared environment" (p. 40). In this section, it is positioned as a technology, that provides functions for all of these requirements and it is made clear that it comprises both the software and the hardware aspect of information systems. While this section gives a comprehensive summary of the concept of groupware, its technological specification and exemplary application areas in the context of GroupOrga is kept rather brief in order to explain the term *groupware* and its usage.

2.1.2.1 The Term "Groupware"

Experience shows that it takes time before a widely-accepted name for a new technology is found, since it must accurately convey and explain the idea and functionality of the technology to those who have never not heard the term before. Various terms have been formed for teamwork-supporting technology, such as "Computer-Supported Cooperative Work (CSCW)" ([Bannon/Schmidt 1991]), "Groupware" ([Ellis/Gibbs/Rein 1991]), "Workgroup-Computing" ([Nastansky 1993b]), "Group-Decision Support Systems (GDSS)" ([Lewe/Krcmar 1991], p. 345), "Group Communication Support", "Group Collaboration" ([Dhar/Olson 1989]), "Cooperative Computing", "Cooperative Work Systems" ([Malone/Crowston 1990]), and "Computer-Supported Groups (CSG)". According to their respective inventors, each of these terms (and many more) centers around a specific field of technology and each appears to have its own advantage. The terms CSCW and groupware are the most common. CSCW has been popularized by Johansen's publications in this field ([Johansen 1988], [Johansen 1991]). It describes the research field, which has the *role* of information and communication technology at its focal point, whereas groupware stands for the technology itself.

The suffix "-ware" in groupware is dangerous because it often causes one to think of software only. However, the supporting hardware technology and the team processes and their proper management are as important as the programs, routines, and symbolic languages that control the functioning of the hardware and direct its operation. Because this technological aspect is discussed further in section 2.2.1, for now, groupware will be examined more generally as a synopsis of computer-supported activities for teams.

2.1.2.2 Groupware Support for Teamwork

Groupware presents a generic term for specialized, computer-based tools, which center on the collaborative work in teams. Typically these teams are those small or medium sized project groups which have been characterized above. The term itself comprises both software and hardware, as well as the services for and support of group-processes (see [Ellis/Gibbs/Rein 1991], p. 40).

This technology can be used by permanent teams or workgroups that have long-term goals, or it can be used by temporary project teams that have short-term assignments. Such lateral cooperation can be formal or informal, spontaneous or planned, structured or unstructured. Furthermore, the team members can take action at the same place or they can interact via communication technology from different locations, making it unnecessary for the participants to be in the same place.

Although computer systems can be used by several people simultaneously, this *Multi-User-System* technology differs from the concept of groupware technology illustrated here. Multi-User or *Time-Sharing-Systems* focus on groups of people consisting of separate individuals who are serviced sequentially and who have little or no interaction or common tasks.

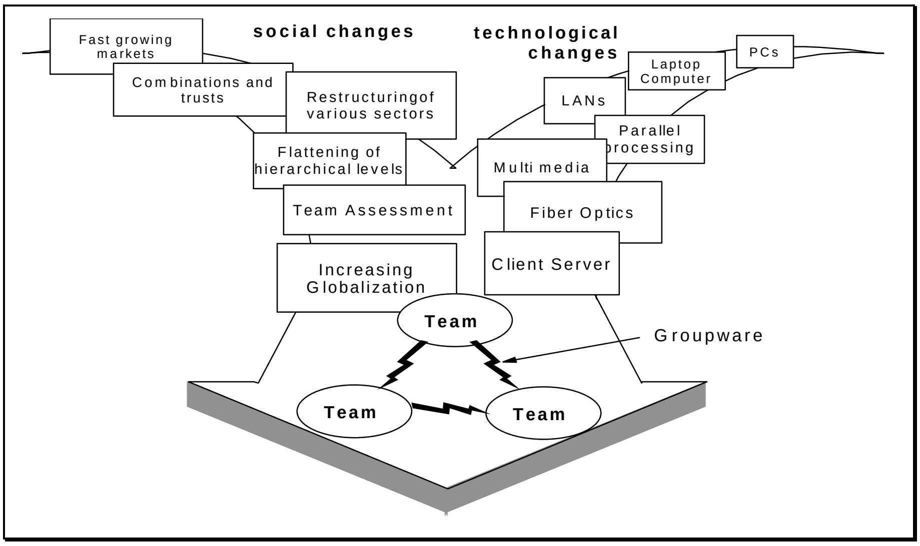


Figure 2-4: Social and technological driving forces for groupware

[Ott 1997a] lists a number of economical and technical trends which are summarized from [Johansen 1991] (p. 91). These trends are seen as driving forces for a rapid development towards lesser levels of hierarchy and an increasing focus on teams and teamwork. However, they should not be seen as a closed explanatory approach, but as spotlights of a current overall business trend. Figure 2-4, an adaptation from [Johansen 1991], shows some of those aspects addressed here, which lead to an increasing deployment of groupware technology. For further discussion on these business trends, refer to the two references mentioned.

2.1.2.3 The Technological Perspective of Groupware

While the preceding sections dealt with the team-support qualities of groupware and the advance of its use in organizations, the main interest here is on the technical specifications and characteristics of it. Three different views arise for groupware, one of which is examined closely here: information management in shared information spaces. The two other distinct application areas, electronic messaging and workflow management, are put aside for now.

The Shared Information Space

All groupware approaches have the common goal of setting up a shared information space for teamwork. This information space is compiled, used, extended and changed by all members of the team. In this shared information space there are multiple ways to collect the information and various tools are provided to jointly generate the information. Since cooperative work is performed in distributed environments with different users who have different goals and viewpoints, the interaction process results in interpersonal discrepancies. Many of these non-technical difficulties have not been encountered in traditional group technology, such as Multi-User-Systems or Time-Sharing-Systems (see [Bannon/Schmidt 1991]).

The users have diverging strategies in solving a problem. Each decision is formed by the individual type of problem solution, which may not be readily accepted by other team members. In order to ease the understanding of a team member's result, each member should have access to all information—the shared information space needs to be *transparent*. From their observations on joint document creation in distributed teams Dourisch and Bellotti [1992] deduce that a transparent information space makes the activities of other group members easier to understand.

Furthermore, decisions are made within a certain framework. Therefore, it is indispensable for the team members to know the relevant framework of a decision process in addition to the actual information or result. Heath and Luff [1993] underline that transparency leads to a better cooperative understanding and is therefore the precondition for building a context for one's own activities. Bowers [1994] concludes similar results from a systematic examination on the use of groupware.

However, reality shows that organizations (and the performing teams) are not perfectly collaborating systems. They are temporary coalitions of individuals with different interests and goals. Correspondingly, information is often used to gain power. On the one hand, this observation supports the demand for information transparency. On the other hand, the divergence of interests and motives weakens the prospect of it becoming reality. The request for a comprehensive and fully transparent data-basis appears to be unrealistic; transparency is restricted so that team members can control the distribution of information (see [Heath/Luff 1993]).

When discussing groupware-based applications that use shared information spaces, Hartmann, Kahler and Wulf [1993] point out that transparency must be kept in mind. Ignoring the manifold intentions of people involved, their various strategies and different work contexts avoids, but does not solve, the problems (p. 65).

Requirements to Groupware Platforms

Abstracting from the technological viewpoint, the aspect of groupware becomes less visionary and much clearer. Groupware is not a new concept. In the past, it consisted of flip-charts for commonly created graphs and charts in group meetings; pinboards for collecting ideas in brainstorming sessions; public bulletin boards and employee newspapers for discussing opinions; and implicit knowledge in the heads of the employees about how a business process should run and why it resolves into an output ([Dier/Lautenbacher 1994]).

However, the technological view of groupware has resulted in confusion and a misjudgment of its great potential. Using tools which support electronic mailing (e-mail), shared access to common information and a better coordination of activities can result in substantial productivity gains. But what remains decisive is the context of such *trivial* applications. The development of a "shared mental model" ([Dier/Lautenbacher 1994], p. 26) is also important, as the above considerations have shown. Technological requirements for shared mental models are discussed below.

Group-authoring is one of these requirements ([Johansen 1988]). Comments in electronic documents, abstracts and text blocks are combined in groupware to create a centralized report which can be accessed by all team members. The technology allows team members to make additions and document revisions, and remembers who made which changes. Team members can makes changes next to previous comments, and can compare the revisions easily. The goal is to improve the speed and the quality of creating team-documents.

Moreover, teams have different support needs, and an **integrated groupware system** that supports these different needs is certainly attractive. Johansen [1988] argues that "comprehensive support is difficult to provide, even if the focus is only on one type of team. Still, this approach to groupware is an important direction that is becoming feasible" (p. 38). However, he finds that this requirement is hard to fulfill, "With today's groupware products, users are likely to find that the specific functionality they achieve within an integrated system is not as powerful as that same functionality in a stand-alone system" (p. 39). The situation described by Johansen characterizes a discrepancy between the values of the power of integration in groupware platforms and specific functional areas. Similarly, Lewe and Krcmar ([1991], p. 346) formulate that groupware should allow for such an integrative approach, in order to minimize media breaks and process interruptions due to incompatible, functional software packages. In their opinion this is achieved by implementing groupware among existing systems or by integrating them into the groupware platform. This requirement for sufficient integration of all functionality is known as "seamlessness".

Since teams have an obvious need for a group memory, particularly if it allows the various members to search it the way they personally prefer, groupware should provide such a **group memory** capability. Search methods are likely to vary between team members and a problem arises in structuring data to be accessible to team members. Therefore, flexible search and indexing structures are necessary. With a technology called *Hypertext*, a non-linear linking and indexing for flexible storage and retrieval options can be implemented in groupware ([Nastansky 1992], p. 10). Similarly, Engelbart points out to facilities to create, transport, store, access and manipulate so called hyperdocuments, which can be linked and back-linked, for easy access to relevant information at the right time ([1990], pp. 151ff.).

A similar requirement asks for **full-text** search functionality, since teams create and use large amounts of information. In the work processes this information may be hard to locate, but full-text search in groupware environments would decrease the effort to find and select semi-structured or free-form text, with more power achieved through more structure ([Lewe/Krcmar 1991], p. 347).

Shared calendaring and scheduling and **project management** are also necessary aspects. These essentials for groupware are derived directly from the idea of teams, i.e. teams have to coordinate their schedules internally and externally. However, implementing group calendaring very often fails due to personal factors, such as the team member who refuses to use an electronic scheduler. Therefore, different approaches to calendaring and scheduling always discuss a combination of private and public calendars. Requiring project management support results from obvious and often pressing needs for task planning and coordination. Groupware plans what needs to be done, tracks progress, and coordinates activities of individual team members. Workflow Management software is one type of software within groupware and will be examined in section 2.1.5.

Characteristics of Groupware Products

So far the *need* for a groupware infrastructure has been established, which is to exploit the integration of the three fields of communication, collaboration and coordination. From this reference point, key infrastructure characteristics are discussed in current literature (see [Nastansky 1993a], [Lotus Dev. 1995], [Lewe/Krcmar 1991], [Johansen 1991], [Ott 1997a], p. 93).

The **object store** or **distributed shared database** that houses the common information and manages the data, regardless of its original source, is considered the core of the groupware infrastructure. This shared database environment is the message store for communication applications, a virtual common workspace for collaborative applications, and a shared database for coordination. Such a database environment is internally consistent across all of these applications, which paves the way for a consistent method of handling information throughout all stages of teamwork. In addition, groupware comprises completely different, **multimedia data types**, compared to those supported by conventional applications. Such complex documents—often called "compound documents" or "semi-structured documents"—are built as the *object container* and should support a wide variety of objects, such as text, numbers, rich text, images, graphics, video, voice, hyperlinks, and embedded applications ([Nastansky 1993a], p. 11). Valuable information can be generated only by this combination of different data types. Most importantly these documents can contain processing intelligence, which dissolves the traditional split between program orientation vs. data orientation.

Distributed databases founded on a distribution model are another key infrastructure element. Point-to-point messaging between the team members can be seen as a necessary (but not as the most important) communication type. More likely, information management takes place by distributed and replicated databases that reside on the work places of mobile or local team members and on servers in local area networks (LAN) and wide area networks (WAN).

This database approach provides the best way to cope with and manage the large amount of document-based data in the team environment. In order to reach a consistent and logical view of this physically distributed information, **database replication** synchronizes the differences in multiple copies of the same object store at geographically dispersed locations [Kawell et al. 1992]. The process of replication appears straightforward; however, several replication technologies, such as bi-directional replication or client replication, can be selected ([Lotus Dev. 1995], pp. 31f.). So far, replication technology has been emphasized as a means for team-to-team communication. Moreover it must be understood as a technology that rates high for the team member using a mobile office. Just recently this aspect has become very important, when for groupware technology an **integration with the WWW** has been achieved, as described in [Ott 1997b] (p. 256).

Since textual objects will remain as the main information container, **text and document processing** are still important for groupware applications. An integrated editor with necessary functionality eases creating and editing of compound documents. Alternatively, external editors can be integrated into the groupware environment if the existing application is ill-equipped with editing functions. At the same time, a characteristic of groupware applications is a **text retrieval function** for the information store. Document management in groupware raises yet another key infrastructure element of groupware: **Workflow Management**. Dier and Lautenbacher [1994] define the automation of document-based processes as the main goal of groupware, as long as it is understood in the context of collaboration and coordination (p. 32). The different types of management and control of workflows will be discussed in section 2.1.5.

Importing and exporting of external data sources is also important for data integration, although much of the information in groupware applications is actually created there. Groupware connects team members and applications in different technological environments, working with varying data formats in other data stores. Therefore, a groupware infrastructure seamlessly imports, shares and leverages the structured data in relational databases or the semi-structured data in desktop software. Object Linking and Embedding (OLE) and OpenDoc are defined as the industry standards for cross-platform data integration and are supported by a groupware application.

Groupware platforms are often capable of supporting the full spectrum of **application development**, ranging from personal end user programming to powerful, professional application development. While the end user will likely rely on customizing and extending given applications through templates and macro languages, the professional developer searches for a highly capable programming language and development tools.

Communication via e-mail has been discussed and it was characterized as a simple means to pass low structured messages from point to point. E-mail has one main advantage over paper messages; it is fast, but it does not allow for sending any kind of intelligence or high-level structure. Within a groupware framework **electronic messaging** must be more extensive. *Object messaging* can transfer more complex information objects between the users in a groupware environment than e-mail. These semi-intelligent objects can be disseminated between the users, as well as between the distributed object stores, and they integrate themselves into the information structure found at the destination (cp. [Nastansky/Ott 1996], pp. 45ff.).

Due to the high demand of availability on the broad range of possible application areas, groupware needs a **secure access mechanism**. However, the security strategy for groupware has to follow preconditions that are different from those for mainframes. For example, there should be no central instance that would guarantee a coherent security management. Instead of this central authority, different structures, such as distributed servers managing their own security technology are implemented in groupware. Distributing selective information, encoding local information, and preventing interception on the networks are the main features of groupware, which uses strong encryption technologies such as the Rivest-Shamir-Adelman (RSA) algorithm (see [Burnett 1996]).

2.1.3 Groupware in Learning Organizations

The concept of *learning organizations* is of implicit and explicit importance for GroupOrga; however, in this context it is examined in connection with IT (i.e. groupware), rather than as a concept of its own. The term was coined by Cyert and March [1963] and conceptualized by Argyris and Schön [1978]. For current discussion on organizational learning, refer to [Albach/ Wildemann 1995], [GfürO 1996], and [Schreyögg/Conrad 1996]. Research in organizational learning has concentrated on individuals, their behavior and their ability to adapt to changes in the organizational environment. This approach is helpful, since the learning of an individual is the first (and according to Oberschulte, the only) step to organizational learning (cp. [Oberschulte 1996], p. 49). Today, IT plays a significant role in this process. Due to its growing importance and its complementary aspect to human capacities, it can significantly contribute to organizational learning ([Krallmann/Boeckhoff/v. Bogdandy 1996], pp. 7ff., [Wildemann 1995], p. 10).

The future success of organizations will be dependent on to what extent informal procedures and organizational structures are supported through IT with formal patterns. J.S. Brown from Xerox Palo Alto Research Center (PARC) emphasizes that most IT systems are based on formal processes, which worsens the situation and impedes the organizational learning process (cited in [Klotz 1995], p. 14). Krallmann, Boeckhoff and von Bogdandy describe groupware as a technology that fosters informal processes by communication and collaboration. They strengthen their position with a case study on groupware use in marketing and sales (pp. 7ff.). In this case study only specific employees knew customer and market information, and the availability of this data depended on their presence in the organization. After implementation of a groupware system, this knowledge was available to everyone.

In [Dier/Lautenbacher 1994] the influence of groupware on organizational learning is described in great length. They identify five single components of learning that make up a learning cycle. All of them have to be run through for learning to occur (pp. 80ff.):

- **Sensitivity.** A better perception of events and changes in the environment can be reached by groupware, since it provides the means to distribute information unhindered by others (superiors for example).
- **Symbolization.** Possible forms for presenting personal ideas, experiences, and observations to others are extended through additional, electronic media. Groupware offers another medium, next to text, graphics and speech.
- **Interaction.** This aspect addresses the actual *distribution* of information. A frequent problem is the absence of relevant communication partners. The "omnipresence of information and communication" (p. 82) achieved through groupware cannot be achieved through any other medium (internal mail for example).
- **Reflection.** It represents the actual learning phase, which generally happens informally as discussions such as in the cafeteria. Again, groupware can encourage this process, and it supports a more factual and objective process.
- **Integration.** When the new insights are manifested in the organization, the learning process has been successful. Groupware can anchor this knowledge in an organization, for example, through an organizational know-how database. This aspect is a main focus of this research, and is addressed in the subsequent chapters.

Similarly Güldenberg and Eschenbach ([1996], p. 7) describe a process of organizational learning. They also refer to an organizational know-how (data)base, with the difference being that their concept does yet not conceptualize it as an electronic medium. Like Dier and Lautenbacher, they depict the learning process as a cycle.

The Schmalenbach-Gesellschaft e.V. states "only 'learning organizations' have the ability to implement necessary changes quicker and more successfully than their competitor" (p. 626, translation by the author). Information technology, like groupware, contributes immensely to this ability ([Schmalenbach 1996], p. 658).

2.1.4 Office Management Systems

In order to apply groupware and teamwork to office management systems in a meaningful fashion, some understanding of the office is necessary. Most descriptive studies have placed great emphasis on apparent office actions, suggesting that offices are the embodiment of these actions. *Office automation* was of great interest to the researcher in the 1980s, and various models and approaches were introduced. This section takes a critical look at the way offices are conceived in the office automation literature and at how this thinking has come true in today's implementations. Additionally it presents an understanding of office management in the context of GroupOrga.

2.1.4.1 Office Perspectives

In attempting to develop an understanding of an office and its operations, the notion of *office* must be examined. This, however, is difficult. For example, Dodswell laments, "It is extraordinarily difficult to provide a concise and clear definition [of office]" ([1983], p. 8). Viewing the office as a place where management work is conducted, or as a set of functions and activities whose output is written and oral, is likely to lead to a rather narrow focus. While the former observation concentrates on geographical constraints, the latter concerns itself only with what people do in offices. Neither view sufficiently takes into consideration the fact that offices are not isolated entities, but rather they are interacting and existing independently within some larger context—the organization.

Bracci and Pernici [1984] were the first to emphasize the complex nature of offices. They summarized a variety of concepts that are embedded in an office management system and they also pointed out the inadequacies of some existing office approaches.

Although the concept of *office* poses ample difficulty in terms of definition, there are a number of dimensions or levels through which an office could be conceived (cp. [Ellis 1983]). For example, offices can have:

- Geographical dimension (physical placement)
- Temporal dimension (work hours)
- Activity dimension (tasks that are performed)
- Structural dimension (employee/supervisor relationships)
- Spatial dimension (area in which people work relative to their co-workers)
- Economic dimension (economic criteria that are goals of the organization and by which workers are assessed)
- Social dimension (the social and psychological reasons that motivate people to work in offices)

This list is not exhaustive, nonetheless it is extracted from Ellis' view of the office to suggest the complexity of understanding an office. Ellis and Nutt [1980] state:

> The *office* is that part of a business that handles the information dealing with operations such as accounting, payroll, and billing. In particular, *office work* consists of information-handling activities such as text editing, forms editing, filing documents, performing simple computations, verifying information, and communicating within the office (p. 28).

Even though this definition is rather simple, it will serve as the basis for further examination of office perspectives.

The aspects listed above can be divided into two different theoretical perspectives of the office: an *analytical* perspective and a *social* perspective. These two views represent two noticeably different notions of what goes on in an office (see [Kling 1980]). The former sees the office as an environment in which the people perform a variety of functions, which are conceptualized in terms of largely formal and structured activities within a given organizational framework. The latter sees the office in terms of mostly unstructured and informal human action with intergroup conflict, with the sovereignty of individuals and social groups considered the dominant issue. The analytical perspective describes it in formal, action oriented models, the social perspective can only describe it in informal, unstructured models.

These two perspectives reflect the general notion of the office that exists in the published literature. As next step, the perspectives will be made operational through office models as basis for applications for office management, which are more specific conceptions of an office.

2.1.4.2 Applications for Office Management

A number of different office models are found in literature to describe office activities, some of which will be examined in chapter 4 and serve as the foundation of the proposed enterprise management model. In the context of this section's discussion of the term *office management system*, a more concrete approach will be taken from the applications point of view.

"An *automated office information system (OIS)* attempts to perform the functions of the ordinary office by means of a computer system. Automation in the office aids the office worker in document preparation, information management, and decision making" ([Zisman 1978] quoted in [Ellis/Nutt 1980], p. 29). Although this definition originates from the beginning of office automation, it does not differ significantly from the current one as a more recent definition from Prinz [1989b] shows:

> Since the co-ordinated exchange and processing of messages is an essential part of office work, most of the models ... originate from this application field. In published literature, such models and their corresponding implementations are usually referred to as *office information systems*, *office procedure systems* or *office automation systems* (p. 128).

The same notion is found in [Desai 1991], "Examples of office tools are electronic mail, document preparation systems, desk calculators, etc. The tools are integrated into an environment, to assist decision making."

A Brief Office Automation Retrospective

The aim of the next passage is to give a brief office automation retrospective, in order to distinguish the innovative steps made in this field (see [Prinz 1989b], pp. 129f and [GartnerGroup 1997a], pp. 5f.).

Before the introduction of computer technology office workers used simple **office support systems** such as typewriters, telephones, and calculators. The first improvement made with the introduction of computers into offices was the activation of text-editors, word processors, formatters, spreadsheets and graphic programs. At first these applications were isolated and had no interaction among the various applications. Consequently the next step was to enable data exchange between them and to provide the office worker with a standardized user interface for all application types. However, all these characteristics supported only the operations of a *single* office worker (see Figure 2-5).

With the advent of **office information systems,** several office workers were supported in their common work. These systems introduced facilities for the exchange of data between office workers, which was first implemented by shared files or database systems. Electronic communication and the exchange of data between office workers on remote computer systems was simplified with e-mail systems, which were later standardized.

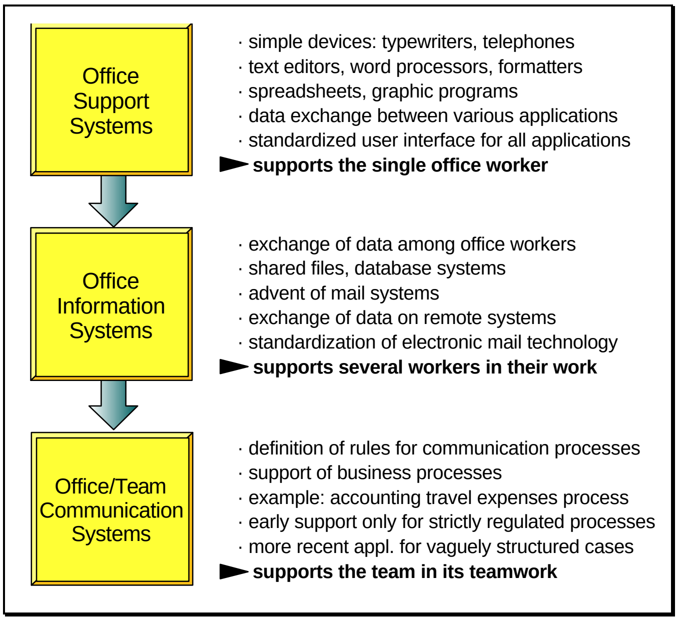


Figure 2-5: Stages in office automation

Using the term **office/team communication systems,** Prinz describes applications which allow the coordination of business processes by means of IT. Here, the definition of rules for the execution of specific business processes is addressed and the examples used are simple, standard business processes. While early models handled only standardized processes, modern applications also support less structured processes.

2.1.4.3 Current Office Management Approaches

The current office management approaches are office information systems that Prinz describes as supporting several office workers in their actions. They provide certain tool-like applications which are used by operating systems supporting simple network operations. These systems, such as Microsoft Office or WordPerfect Office, are considered mature and are offered by a number of software firms. Most of these systems have a long development history and were developed from a functional approach (supporting single, separated functions).

These current systems are based on a client-server architecture, with the personal tools at the workplace (client) and the joint services at the server side. However, this subdivision is theoretical and is chosen to make the respective functionality clearer. In concrete terms, these elements are integrated on the user's desktop without a deliberate distinction between the two.

On the client side, tools such as text-processors, archiving, spreadsheets, mail, graphic programs, desktop publishers (DTP), simple databases, calendars, and calculators, are available (see Figure 2-6). The office worker can make use of the tools as required.

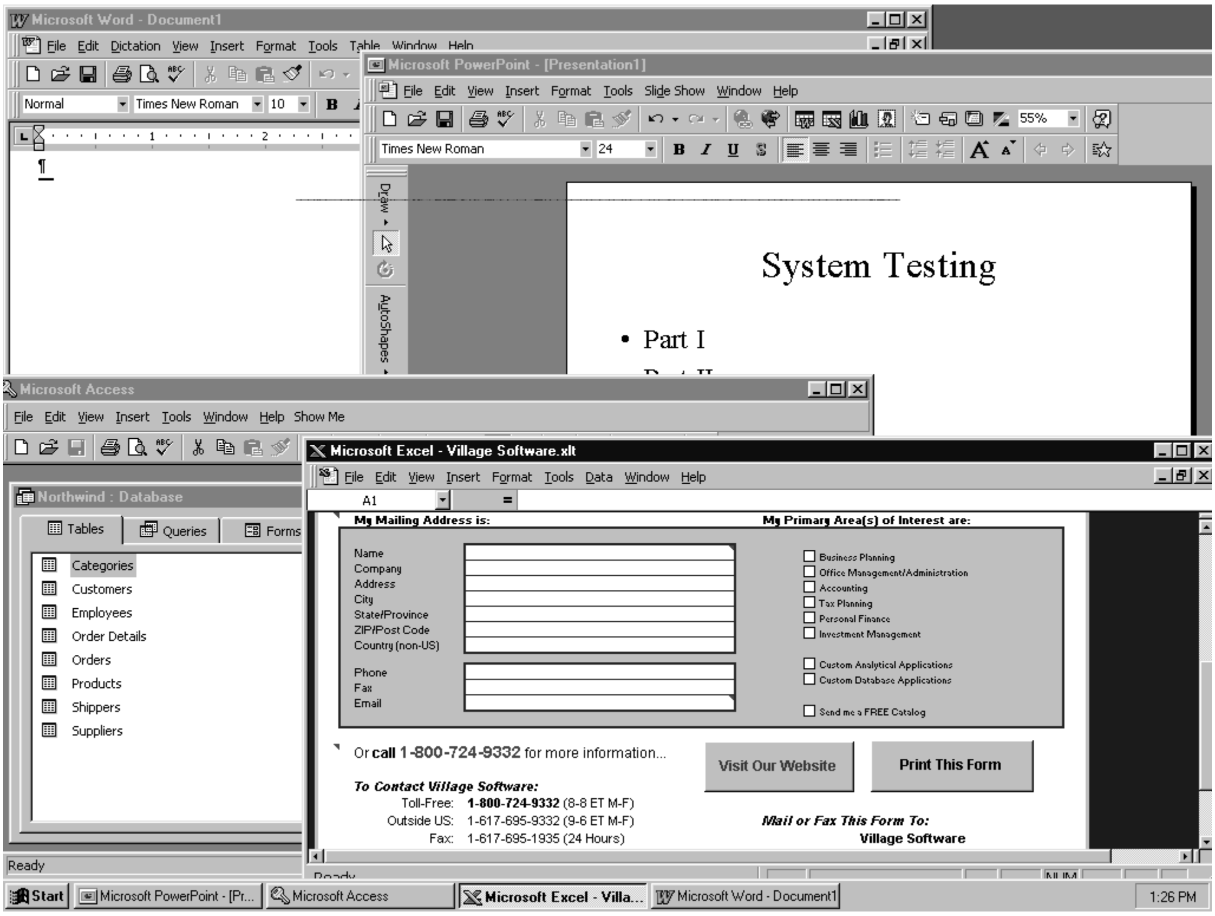


Figure 2-6: Traditional stand-alone office tools in client-server environments

The single tools result in a system with three distinguishing marks:

- ❑ The "look and feel" of their operation is the same
- ❑ Data from one tool can easily be transferred to another
- ❑ Outside applications can be integrated through standardized interfaces

The use of such applications is usually simple and the different tools are integrated in the form of a graphical "desktop" metaphor. Various services are provided by server components in this architecture, among them: connectivity to external and public networks, e-mail routing, central bulletin board databases, security/network/file management and print services.

However, it must be pointed out that these discrete tools rely on a file architecture which is marked by locality and primarily by personal access (cp. [GartnerGroup 1997a], p. 17). Sarin, Abbot and McCarthy [1991] state about current office systems, "... if one looks closer, one notices that the dialog between the worker an the computer system has not changed. The worker typically locates a file in a directory, launches an application against that file, and then browses/edits the file using concepts defined by the application" (p. 213). Like other authors, they ask for office systems that flexibly integrate users in the business process, as well as provide guidance and structure where necessary. In addition, the current office system is aiming at the production of higher quality printouts in less time, rather than at the general provision of information to teams (p. 213).

2.1.4.4 Distributed Office Management

Current office management systems do not support office work that is divided into specialized domains handled by different office workers (where collaboration is needed in order to accomplish the work). Offices are in a continuos state of flux, and the advent of remote work, for example, has removed the geographical boundaries associated with offices. The functions offices carry out can now be distributed across the social and geographic landscape (cp. [Ellis 1983], p. 11).

Some characteristics of this distributed office work (which may be taken as concrete forms of what was discussed under the team aspect in section 2.1.1) must be taken into consideration:

- In a large organization, the individual cannot understand how everything is done. Therefore, individuals need to collaborate.
- Office workers perform their work concurrently. Sometimes they synchronize their activities.
- Tasks are performed using a set of inputs. Office workers produce a set of outputs. These inputs and outputs may be received and transferred to other office workers.
- In cases where office workers encounter missing information, they have to contact some other office workers. In other words, they have to communicate.

Hence, office work that involves the simultaneous participation of several persons is teamwork. An accumulation of activities by autonomous actors who perform locally using only personal data is not.

For the purpose of this discussion, Tsichritzis [1985] and Reichwald [1984] both summarize two problems that arise when trying to support distributed office activities by providing centralized office systems (that is, gathering the knowledge of all office workers into a global and consistent knowledge store, such as a central file server):

(1) As the size and the scope of a firm increases, general knowledge gives way to specialization. Integrating the specialized knowledge may not be difficult, yet converting collaborative office procedures to fit in an integrated environment will not be easy since it requires the *integrator* to have knowledge of all the different kinds of specialization.

(2) In a centralized office system, partially or occasionally inconsistent office procedures, specified by different office workers, are not allowed. However, such inconsistencies cannot be prevented in an environment with multiple players and are allowed as long as they are not too serious.

To overcome these drawbacks of current office management systems, an approach that supports office problem solving in a *logically* distributed environment becomes necessary. In this context, Woo and Lochovsky [1986] contrast *logical* and *physical* distribution:

> In some systems, information is geographically distributed for performance purposes rather than for conceptual need. The term, *logically*, is therefore used to indicate the logical need of organizing information without having to worry about the physical location of the information (p. 185).

A proposed solution in the literature is to leave office knowledge in independent knowledge bases, as they appear in the manual office, and allow them to collaborate when there is need for consensus.

2.1.4.5 Innovative Approaches to Office Management

The existing office automation applications presented in section 2.1.4.3 stem from functional developments of proprietary systems by hardware vendors or third party software firms. A second and younger line of development takes advantage of progressive standardization. On a market of small and mid-range IT based on LAN architectures, open systems that are independent of specific hardware or software are developed. These *open systems* lean toward *platforms* (as opposed to *tools*), which help the user to establish a knowledge of organizational processes ([Sarin/Abbott/McCarthy 1991], p. 214). Such office systems offer the user a framework for combination of various applications through standardized interfaces. The architectural characteristics are LAN-based, object orientation, modularity, and openness for integration of different systems.

Götzer [1995] and others present groupware and Workgroup Computing as the representation of this new type of office management system and as a consequential outcome of a development towards distributed office work. But this simple view appears to be insufficient for the context of GroupOrga research because groupware may serve as the platform for the implementation of an office management system, but it cannot be comprehended as an office management environment of its own.

A prospective concept in question is based on an architecture which, on the one hand, assists office work with flexibility and responsibility of the team member and on the other, provides the necessary structure and conformity for activities in teams. Since office work falls on a continuous spectrum—one end representing structured office work, and the other end representing unstructured office work—a groupware-based framework has to be positioned between traditional data-processing and open, loosely structured group activity. Office management systems need to be rigidly structured by means of distinctly specified workflows, given and fixed information containers or uniform database design, and at the same time they have to allow specification of individual workflows, creation of personal sorting criteria, and ability to accept personal tasks.

In order to achieve this combination of structure and flexibility as the core for the management of office processes, the framework should consist of two basic components: a flexible working entity at the user's desktop and a corresponding structural part provided by a semi-centralized system repository, which would serve as the base for all distributed applications incorporated in the system and for workflow control. Although this requirement appears to be convincing, few approaches have yet to suggest a feasible implementation. Section 2.2.2 introduces such a groupware-based office management application.

2.1.5 Workflow Management Systems

Workflow is concerned with the control and automation of procedures where information, documents or single tasks are passed between participants of business processes according to a defined set of rules. BPR is frequently mentioned in the same breath as Workflow Management (WfM); however, opinions on their similarities and differences are contradictory ([WfMC 1996b], p. 6). This section compares BPR and WfM and introduces Workflow Management Systems (WfMS).

2.1.5.1 Business Process Reengineering or Workflow Management

BPR is a methodical approach that redesigns the business processes and designs a resource allocation and an organizational structure that fits seamlessly with the newly designed processes ([Hammer/Champy 1993], pp. 32f., [Theuvsen 1996], p. 67, [Coulson-Thomas 1994], p. 7). This redesign is meant to create decisive improvements in customer orientation, processing time, reactivity and flexibility of organizations ([Davenport/Short 1990], p. 14, [Harrington 1991], [Wirtz 1996], p. 1024).

The term *reengineering* was first used by Hammer [1990], while almost simultaneously Davenport and Short [1990] published a study on the similar topic of *Business (Re-)Engineering*. Hammer and Champy [1993] added a focus on processes with their definition, "Reengineering is the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service and speed" (p. 32). The essential mark of BPR is the process orientation of organization design in contrast to traditional approaches which rely only on structural characteristics, such as function, product or region. In other words it is no longer *process follows structure*, but *structure follows process*.

BPR is not a single and nonrecurring method that improves business processes overnight, but rather an extended concept that has the ongoing fundamental reorganization of business activities as the goal. Wirtz ([1996], p. 1024) extracts three essential characteristics from Hammer and Champy's [1993] and Davenport's [1993] recommendations:

- Process-oriented view of all business activities
- A new orientation of business activities stretching across traditional functions
- Focus on customer-oriented net product

Figure 2-7, an adaptation from Davenport and Short ([1990], p. 14), shows how closely process-focus and cross-functional orientation are connected.

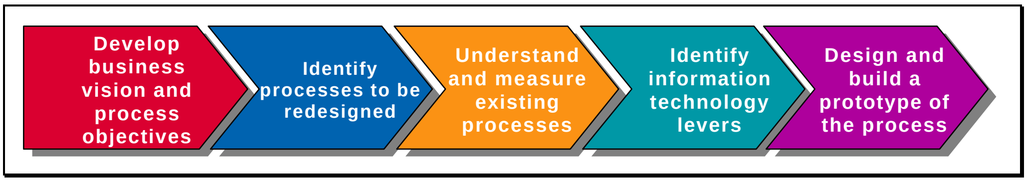


Figure 2-7: Reengineering process steps

The enormous debate around BPR and its popularity in organizations has been driven mainly by (north-American) consulting firms (see [Nippa 1995], [Österle 1995], p. 13), while the reaction from scientists has been hesitant ([Theuvsen 1996], p. 66). Wirtz suggests that this is due to those consulting firms having shaped most of the BPR concepts. After all, he argues, the leading representatives of BPR, such as James Champy (CSC Index) and Thomas H. Davenport (Ernst & Young) are connected with such consultants (p. 1026). Recent scientific studies of BPR is found in [Gaitanides/Scholz/Vrohlings 1994], [Picot/Franck 1995] or [Kieser 1996]. Nevertheless, all of these investigations conclude that "there is some potential in this concept to increase efficiency of organizations" ([Kieser 1996], p. 185).

BPR is often directly associated with Workflow Management, which is concerned with the assessment, analysis, design, definition, modeling and operational implementation of the business processes of an organization. In spite of this understanding, a distinction that Georgakopoulos, Hornick and Sheth [1995] and the WfMC [1996b] also make will be adopted in GroupOrga—BPR activities should be understood as a general restructuring method for business procedures and not all of them result in workflow implementations—notwithstanding that workflow technology is often an appropriate means to support these activities. Conversely, not all workflow implementations necessarily form part of a BPR exercise, yet some of them only cover a small fraction of what should be done in an organization to adequately reengineer its processes and structures.

2.1.5.2 Workflow Management and Supporting Systems

Workflow is "the flow or progress of work done by a company, an industry, a department, or a person" ([Houghton Mifflin Company 1994]). In the English language, the term workflow has no direct connection to any business field, neither office nor manufacturing. In recent publications, the term refers to the flow of information through an organization as it is processed, distributed, edited, and compiled, rather than to the flow of materials through any type of production chains (see [Gable 1991], p. 1). In this project, the focus of the flow will be restricted to information in offices.

In this context, the basis of workflow is the notion that business processes are actually sets of tasks done in a prescribed order that incorporate information from various sources (see [Gable 1991], p. 1). Hence, workflow is the succession of single tasks that are necessary for the completion of a business process to achieve an overall business goal ([Krickl 1995], p. 30). Often workflow is implicitly used in connection with IT systems, and—while it may also be manually organized—in context of this research it is seen as a form to provide computerized and networked support for the procedural automation of work. Workflow is "the computerized facilitation or automation of a business process, in whole or part" ([WfMC 1996b], p. 6).

A *Workflow Management System* (WfMS) provides computerized assistance in the processing of workflows. It manages and controls the sequence of work activities and the appropriate supply of human and IT resources to the various activity steps (cp. [Georgakopoulos/Hornick/Sheth 1995]). In other words, the WfMS provides the right user, at the right time, with the right tools and data for completion of a dedicated task within a workflow. "Right" reflects the realization of the respective business process ([Leymann 1997], p. 82). McCready [1992] gives a basic definition of workflow management systems:

> Workflow software is the tool which empowers individuals and groups of individuals in both structured and unstructured environments to automatically manage a series of recurrent or nonrecurrent events in a way that achieves the business objectives of the company. Simultaneously, workflow software should allow feedback to managers, ensuring them the opportunity and ability to extend or modify those business processes as the business environment changes (p. 3).

More precisely, a WfMS is a system that "completely defines, manages and executes 'workflows' through the execution of software whose order of execution is driven by a computer representation of the workflow logic" ([WfMC 1996b], p. 6). For such a system, Hasenkamp [1987] knows two different types of *workflow*, which he calls a *workflow type* and a *workflow instance*. A workflow type characterizes the abstract succession of tasks, which is defined only once to describe a certain class of business processes. Workflow instances are derived from a workflow type and indicate concrete processes. For example, the instance "Travel expenses Mr. Smith/July" is derived from the general workflow type "Monthly travel expenses".

Such systems may be implemented in a number of ways, using information and communication technologies and operating in an environment from small workgroups to enterprise-wide scenarios. Rarely has an area of computer science been flooded with system implementations, as WfM has been in recent years. Rather than introducing another WfMS—neither for local nor for wide area business processes—researchers desire a consolidation of efforts in this field ([Jablonski 1997], p. 72).

The discrete functions of workflow management software can be split into five areas ([Krickl 1995], p. 31, [Nastansky/Hilpert 1995], p. 31, [Deiters/Gruhn/Striemer 1995], p. 460, [Morschheuser/Raufer/Wargitsch 1996], p. 5):

- Analysis or simulation tools for the process designer
- Modeling, design or system definition techniques
- Workflow control or steering
- Workflow and activity enactment
- Monitoring and ex-post assessment

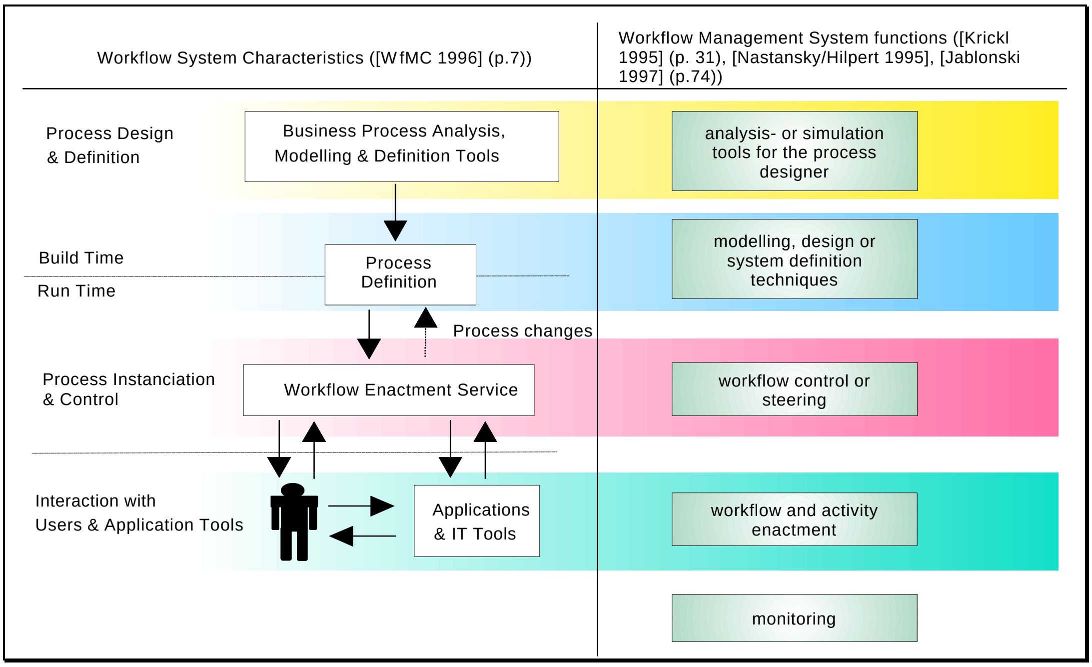


Figure 2-8: Five discrete functions of workflow management software

At a higher level, all WfMSs provide support in three functional areas ([WfMC 1996b], p. 7):

- Build-time functions, which are concerned with the analysis, definition and modeling of workflow processes
- Run-time control functions, which deal with the management of workflows in an operational environment and the succession of various activities

- Run-time interactions with workflow participants, which are the human users and IT applications that process of activity steps

Figure 2-8 compares the workflow system characteristics of the Workflow Management Coalition (WfMC) to discrete functions identified in the literature. Noticeably, *monitoring* and *organizational interfaces* are not explicitly subsumed in the characteristics defined.

The build-time functions are those which support the creation of a computerized definition of the processes in a business. Using analysis, simulation and modeling techniques, a business process is transferred from the real world into a computer definition. Wendel [1996] distinguishes this process definition and modeling phase, that is, the designing of "the computerized representation of a process that includes the manual definition and workflow definition" ([WfMC 1996b], p. 8), in four types, as shown in Table 2-1.

Modeling type	Description
Form-oriented modeling	Focuses on the document, which may be presented in the form of "electronic circulation folders" ([Karbe/Ramsperger/Weiss 1990]). Such folders contain all relevant data, as well as its own routing information ([Karbe 1994]).
Process-oriented modeling	Focuses on the process and tasks, sequences and activity steps. Based on the current activity step, the WfMS decides which documents are needed and provides them to the user. After completion of activities, it automatically determines the next user and again the necessary documents (see [Zisman 1977], [Deiters/Gruhn/Striemer 1995]).
Conversation-oriented modeling	Focuses on the communication which coordinates process activities. Using the language/action perspective, this approach defines the communication necessary for realization of a workflow as *conversation*. In order to carry out activities, these systems provide various conversation types ([Winograd 1995], [Kreifelts et al. 1991]).
Information-sharing modeling	Focuses on a shared information space (cp. section 2.1.2.3) which coordinates the activities by continuously updating status information during the workflow enactment. In this approach the explicit design of processes is superfluous since the mere change and evaluation of status information portrays the flow of information ([Hasenkamp/Syring 1993], p. 35, [Ott/Nastansky 1997c], p. 29).

Table 2-1: Four different types of the process definition and modeling phase in WfMS

Hence, the creation of computerized process definitions may occur in various ways, such as through formal languages, graphical editors, or textual descriptions. This creation is not yet standardized; however, the result of this modeling is considered to be the major element for standardization in the field of workflow management ([WfMC 1996b], p. 8). Based on this definition, the run-time engine creates and manages operational instances of a process, schedules activities, invokes appropriate resources, and assigns work items to users. The detailed description of organization design elements for the allotment of work items to workflow participants is the major concern of the GroupOrga concept. The ultimate completion of activities remains with the human user, or is rarely carried out through automatic IT operations. Often, the user works with a particular IT tool or an application program. These run-time interactions may vary from system to system, depending on the system's overall paradigm (cp. Table 2-1 and [Lawrence 1997]).

Several authors have summed up these separate functions into a *workflow life-cycle* which explains the management of business processes in WfMS as a cyclic procedure. Such a continual ex-post analysis and redesign process ensures constant improvement in the business processes and prevents the computerized models from deviating from real life (see [Heilmann 1996], p. 149, [Nastansky/Hilpert 1995], p. 31, [Krallmann/Derszteler 1996], p. 34, [Deiters/Gruhn/Striemer 1995], p. 460).

Currently, many WfMSs are on the market, one of which is described in section 2.2.3 and taken as the reference workflow system for GroupOrga implementations. Recently, another survey of WfMS was published ([Weiß/Krcmar 1996]), adding to the list of existing surveys (cp. [B.BIT 1992], [BIFOA 1993], [Mummert 1996], [Mummert 1997], [Sodan 1994]).

2.1.5.3 The Workflow Management Coalition Reference Model

In an everyday office scenario the flow of work involves the transfer of tasks and activities between different workflow participants. In addition, they may use different workflow products in order to have specific parts of the process be supported on different operating systems or sub-networks. Products which suit the stage of the process best may be used, but no common specification of workflow management systems exists yet ([Weiß/Krcmar 1996], p. 503).

In order to solve this dilemma, the *Workflow Management Coalition* was founded in 1993. WfMC members include WfMS producers, consulting firms, and research institutions from more than 100 organizations. Its goal is to establish standards that enable the transfer of workflow control between composite WfM applications using various workflow products operating together as a single WfMS. A central project of the WfMC is the compilation of a framework, the *WfMC Reference Model*, that is made up of five interfaces (see [Sauter/Morger 1996]). Results of the WfMC are the passing of the Reference Model ([WfMC 1996b]), shown in Figure 2-9, and a workflow glossary with more than 50 definitions of workflow-related terms ([WfMC 1996a]).

Five applications or tools were discussed in section 2.1.5.2. The corresponding interfaces, which have yet to be defined, are supposed to allow interchangeability between components of different WfMS. The different forms of interchange are already clearly specified on seven levels. A simple form is that of homogeneous use of the same WfMS between two partners, and a complex form is the coexistence of these two partners, using absolutely different systems ([Weiß/Krcmar 1996], p. 507, [WfMC 1996d] and [WfMC 1996c]).

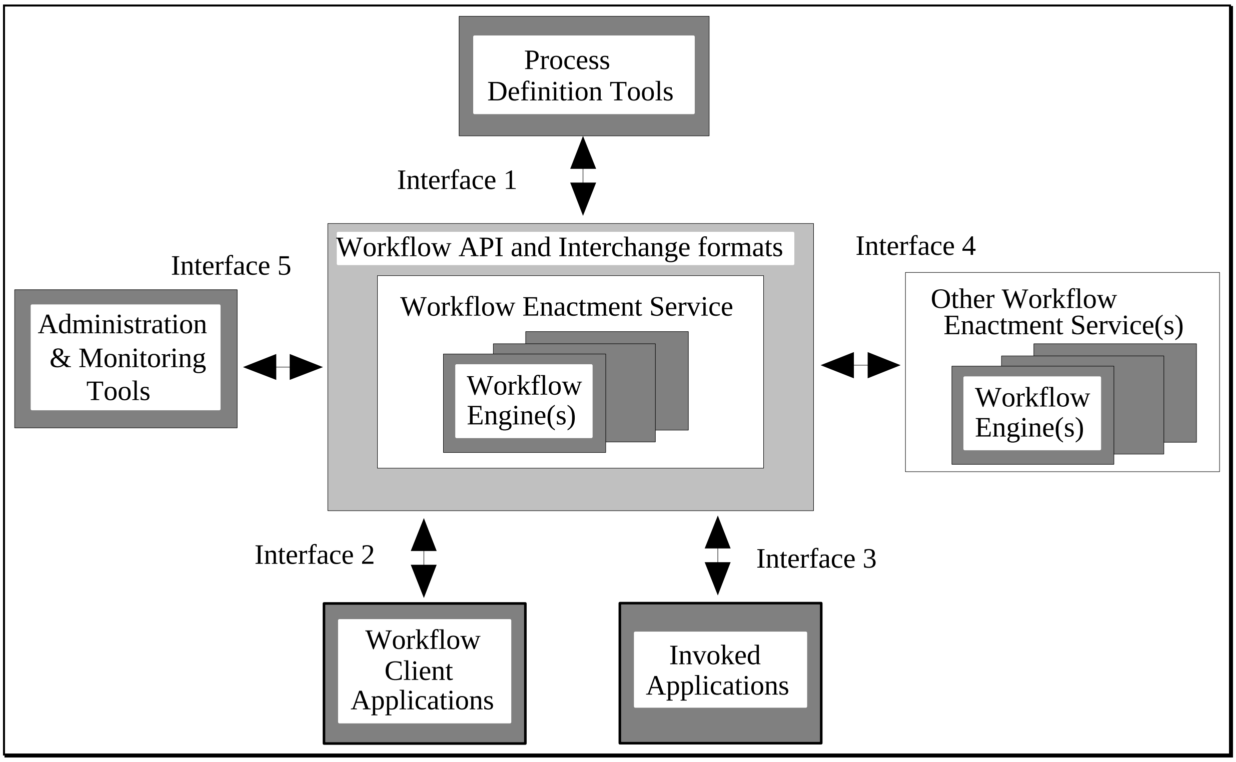


Figure 2-9: The WfMC Reference Model and the five WfM interfaces

Jablonski [1995a] (p. 80) and GartnerGroup [1997b] (p. 3) fault the weak concreteness of the Reference Model, since all specifications exist only in a very coarse framework, and, for example, no detailed information is available on the transportation of tasks and activities. Therefore, the model can be matched to most WfM products in the marketplace, thereby having less relevance toward a common workflow understanding. A tighter definition and some sort of hallmark combined with a WfMC certification is desirable (see [Weiß/Krcmar 1996], p. 507).

2.1.5.4 An Important Concern: Flexibility in WfMS

In recent years, WfMS have fallen short of most of the user's expectations. Except for a few success stories, WfM is considered a "nice technology, but it doesn't allow us the flexibility to handle the many exceptions, and to really get our work done expeditiously" ([Ellis/Keddara/ Rozenberg 1995], p. 12). Major criticism points at the systems being too rigid and directive, thus restricting rather than assisting the users.

In answer to this criticism, workflow literature deals with the aspect of *flexibility* in WfMSs. But since WfMSs are meant to guide and control the coordination of various process participants, with the aim of reaching a mutual goal in a given time with prescribed standards, the demand for flexible (often called *ad-hoc*) changes appears to be a contradiction in itself ([Hagemeyer et al. 1997], p. 179). Nonetheless, two different categories of flexibility in WfMS can be identified ([Rathgeb 1996], p. 56).

On the one hand, the concrete flow of a workflow instance cannot always be planned in advance. At most it can be modeled only vaguely. Only at run-time can the final decision be made on how to carry out the concrete process. Deiters et al. [1996] distinguish process classes due to their impossibility to be pre-planned. In this context they understand *pre-planning* as sufficiently defining details of a workflow in advance. For example, the authors call processes in which parts are well structured *semi-structured* processes. For such processes they suggest a belated modeling called *late modeling* at run-time.

On the other hand, exceptions cannot be foreseen. Long term exceptions are, for example, changed customer needs, drifting market conditions or the formalization of informal organizational structures in a business process (see [Herrmann 1995]). A long list of short term exceptions are found in [Galler 1995] and [Rathgeb 1996], both of whom provide a classification of the different cases. The former distinguishes the concrete reason for the exception (determined by the system, the data or the user). The latter investigates what workflow characteristic (overall timing, single activities or the general workflow processing) is violated by the exception. Table 2-2 combines reasons for exceptions collected by both authors and classifies them.

Not surprisingly the upper left field of Table 2-2 remains empty. In general, a workflow in a particular WfMS is modeled and then simulated, so any timing inconsistencies in its general logic are eliminated beforehand.

Exception ...	**... determined through the WfMS**	**... determined through workflow data**	**... determined through the WfMS user**
... violates timing restrictions		❑ out-dated data	❑ processing cannot be completed in given time ❑ user wants to interrupt work
... blocks single activities	❑ passing the task on to the wrong workflow member	❑ wrong or faulty data ❑ misleading data	❑ user needs to ask for advice ❑ user deliberately wants to make changes
... is of general significance for complete process	❑ occurrence of technical problems ❑ compatibility problems with hardware or software	❑ missing data ❑ loss of data	❑ absence of responsible user ❑ user refuses the task ❑ withdrawal of workflow processing through initiator

Table 2-2: Reasons for exceptions and exception handling in WfMS

Regardless of the exception, to be useful a WfMS has to support these necessary ad-hoc changes, as well as highly structured workflow types. In the literature, a number of approaches that classify business processes are found ([Rathgeb 1994], pp. 52f., [Hasenkamp/Syring 1994], p. 19). In an earlier publication, Reichwald and Picot [1987] discriminate tasks by the characteristics *complexity*, *information required*, *cooperation partner* and *pattern of solution*. In [Picot/Rohrbach 1995] this typology is enlarged by separating *singular processes*, *pertinent processes* and *routine processes*.

Nastansky and Hilpert [1994] name such three process types *ad-hoc* and *task force workflow*, *semi-structured workflow* and *standard workflow*. Kirn and Unland [1994] identify three similar types as *unstructured*, *semi-structured* and *structured*.

Ad hoc WF	autonomous Workgroup	Semi-structured Workflow			Standard WF
	open team task	a) *open* team task within standard WF	b) *controlled* team task within standard WF	c) *ad hoc* modification of standard WF	
e-mail, store-and-forward	one-step *next agent*, shared DB	combination of pre-determined and open tasks within a single workflow	partially unspecified elements within pre-determined workflow	completely pre-determined workflow and exception	generally preset *next agent,* shared DB
- urgent - short-lived - exceptional - confidential	- shared access - common task	- completely open as well as standardized tasks - intersection of both	- pre-set *number* of group members takes part - regardless of the sequence	- highly recurrent - pre-determined - *easy-to-apply* ad hoc modification / re-routing	- highly recurrent - well structured - pre-defined
e.g. new type of request	e.g. co-authoring of publication	e.g. co-editing of annual report	e.g. counter-signature	e.g. consumer credit app. (*particular* customer request)	e.g. consumer credit application

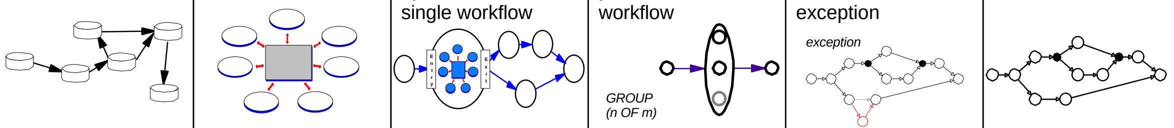


Figure 2-10: The Workflow Continuum

In their *Workflow Continuum* (see Figure 2-10), Nastansky and Hilpert subdivide the semi-structured workflows into *open team task within standard workflows*, *controlled team task within standard workflows* and *ad-hoc modifications of standard workflows*.

Hagemeyer et al. [1997] examine three conceptual approaches for the realization of semi-structured workflows, each approach supporting late modeling and exception handling:

- **Modeling at run-time.** This approach lets the workflow participant design parts of the workflow at run-time. Only when a certain incident takes place does additional modeling become necessary. In this case, the whole process (except the semi-structured part) is modeled in advance. The to-be-defined part is represented by a block box, or in the case of the Workflow Continuum, as an open team task (see [Deiters/Gruhn/Striemer 1995]).

- **Adaptation of a process through variants.** Instead of modeling in a top-down procedure, the provision with a number of variants for a specific workflow type can also provide flexibility. When faced with a decision or exception, the user is offered a selection of similar, yet slightly different alternatives. This approach requires no modeling knowledge from the user (see [Allweyer 1995]).

- **Step-by-Step modeling.** This concept supports the coordination of activities between process members which become necessary when an ad-hoc modification has to take place. It highlights the essential aspects of the workflow to be changed, such as the activities that are directly affected by the change. This approach assists in modeling a block-box or open team task situation consistently (cp. [Just 1996]).

Despite these comprehensive mechanisms for exception handling, Leymann [1997] argues that it presents many difficulties from a transaction point of view. He investigates nested, distributed and global transactions. Furthermore, he examines recovery and roll-back functionality. He concludes that most of today's WfMSs are very flexible, but do not fulfill transactional requirements. However, in this project's standpoint, a *human approach* to workflow management is sufficient for the area of office processes.

2.1.6 Designing Organizations: Organization and Design

In the following section aspects of organization and designing organizations will be addressed in terms of their relevance to the content of this research. Mainly organizational terms will be examined, since terms such as *organization*, *organizational unit*, *organizational structure*, and *organization design* are used differently throughout CSCW literature.

A Common Understanding of "Organization"

To understand *organization*, one must understand not only *the action of organizing*, but also *what is to be organized*. Morgan [1986] lists a number of traditional perspectives of an organization; perspectives that conclude that an organization is a machine, an information processing brain, a culture, or a political system. Each perspective leads to a different interpretation, which results in different ways of guiding an organization. This means that for an analysis of an organizational situation, the choice of perspective influences the results of the analysis. For example, choosing an office perspective leads to a certain set of results. Likewise, choosing an IT perspective leads to another set of results. Therefore, when aiming at office *and* IT results, one must choose a perspective that encompasses aspects from both areas.

2.1.6.1 Terminology

An enterprise can be structured in different ways. It can be structured independently for each single occasion in an ad-hoc approach, i.e. by improvisation. On the other hand the planning can be done generally and in advance for repeating cases in form of pre-planned action. According to [Bleicher 1991] the latter form is called *organization*. However, he points out that *organization* has that many different connotations, both in colloquial language and science. Accordingly, he describes *organization* as a goal-oriented, planned, and structured action that brings order into a system (p. 34).

From the manifold descriptions and definitions of the term in literature, two main conceptual meanings can be distinguished (see [Bleicher 1991], p. 35). In short, the two terms may be paraphrased as "the business enterprise *is* an organization" (institutional term), and "the business enterprise *has* an organization" (instrumental term).

The institutional definition uses *organization* as a generic term for institutions of all kinds, such as business enterprises, public authorities, schools, federations, churches, hospitals, and the military. In this context, organizations are socio-technical systems of interacting elements, which behavior is aimed at reaching a specific goal through a number of rules and regulations (see [Grochla 1975]).

The instrumental definition uses *organization* as a specific term, one that comprises the totality of all explicit structural and procedural regulations. Those regulations are not an end in itself, but their purpose is to influence the behavior of all members, in order to fulfill a given task in an ordered and rational way (cp. [Bleicher 1991]).

This institutional use of organization is mainly found in sociology and in organizational sociology, that is, in analyses of behavioral science, while the instrumental use is found in business management and management sciences.

Damm [1994] points out that a focusing on one of the two terms for investigating a particular problem also has implications on the form of the investigation itself (p. 10). This notion is supported by Grochla, who argues that accepting one of the definitions includes accepting the implied understanding of the model of an organization and its surrounding reality (p. 2). Bearing these remarks in mind, this study will use the instrumental definition of organization.

The instrumental stream distinguishes the procedural regulations and the structuring of an organization. Hence, two distinct areas are identified, the process organization and the structural organization. Although no equivalent terms to the German *Ablauforganisation* and *Aufbauorganisation* exist in English, a similar differentiation can be found with all authors concerned with organization (see literature mentioned in section 2.1.5.1). In this context, the process organization is concerned with the spatial and chronological structuring of activities—the *how*, *when* and *what* of organizational activities. In contrast, the structural organization, is concerned with arranging subunits, distributing tasks, and issuing directives—*who* is doing it *where*, and *who* is directing *whom* (cp. [Eisenführ 1993], p. 4).

In order to describe such a structure or to analyze different alternatives of structural organization, characteristics that distinguish one organization from another must be defined. Such characteristics are called *dimensions* or *parameters* of an organizational structure. Eisenführ distinguishes five different structural parameters (pp. 4f.):

- **Formation of posts.** A *post* is a worker's job description. Division of labor, a characteristic of every large organization, can be marked with low or high specialization of an individual. In low specialization, posts are very similar, while in high specialization, post are very different.

- **Formation of (sub-)units.** Posts are combined into units (or divisions), subunits into higher divisions and so on. Hence, the units are the structural elements above the posts. A characteristic for units is its size.

- **Coordination mechanisms.** Each unit has to be coordinated somehow. According to section 2.1.1.3, coordination is the agreement between individuals to reach a common goal. To do this, various mechanisms exist.
- **Centralization.** This dimension explains to what degree decisions are made *at the top level* of an organization or are delegated *to the bottom levels*.
- **Formalization.** This parameter describes the official, mainly written definition of the tasks of posts and units, as well as the organizational processes. The *degree of formalization* measures the extent to which formalized regulations control the events in an organization, in contrast to informal regulations.

2.1.6.2 Organization Design

Grochla [1982] defines *organization design* as a "means (instrument) for setting up an organizational structure" (p. 3, translation by the author). Although mentioning only one organizational aspect, the structure, in his definition, he distinguishes between the design of organizational structures and the design of organizational processes (pp. 23ff.). A systematic design of organizational processes (for example, the division of labor in those processes) is considered to be important. As the more important element, though, he focuses on the structuring of posts and regulations.

With respect to organizational structures and processes, Baligh and Burton [1981] distinguish the term *description* from *design*, and they conclude that design is the logical inverse of description (pp. 255f.). While the description states what the current structure of an organization is, design states how it ought to be. They argue that a good description is necessary for an effective design; however, a description is not sufficient for improving an organization's situation in any way (p. 256).

Design refers to the organizational and technical structuring of worksystems ([Rathgeb 1996], p. 49). It is "a complex *problem solving process*, which consists of different *subproblems*. These problems are solved by specific activities which are *opportunistic*, i.e., they strongly interact and build on each others' results" ([Streitz 1992], p. 12). Streitz continues, "Design usually is a *social process* that involves a group of individuals. Therefore, facilities which support cooperation should be incorporated in an ... environment" (p. 12).

Derived from these definitions of design in organizations, GroupOrga's notion of organization design will be a structured, active process of modeling the institutions of an organization, with the aim to reach an optimal form of the organization. This comprises all activities that are carried out, in order to allow for a systematic and gradual approach to realize the changes. Hence, organization design comprises the act of organizing, which, in GroupOrga will be seen in focus of the interdependence between organization and IT (cp. [Hoppen 1992], p. 11).

In reference to [Kieser 1993], Klotz [1993] argues that organization design is meaningful and effective only when goals for the design process have been formulated. General organizational goals must be present to develop into concrete and operational goals later on. Hence, he outlines two general goals for organization design (pp. 30f.):

- **A performance goal.** Aiming at the productive fulfillment of all production and business processes in an enterprise.
- **A humanistic-social goal.** Concentrating on the satisfaction of all personal needs of organizational members and their motivation.

In other words, each organization design process follows collective and individual goals (see [Kieser 1993]). While the first reflects the goals of the group of persons who are directly connected with an organization's performance (management, employees, stockholders, labor unionists, consultants, governmental authorities, subcontractors and customers), the second constitutes the goals of the single person.

2.1.6.3 More Recent Forms of Organization Design

Two different and more recent approaches will be delineated:

- Process orientation in organization design
- Self-organization as active, planned process in contrast to design of structures only

In a traditional organization, structure and work are organized functionally. In fact, there is only hierarchy and no concept of *process*. Advocates of **process oriented organization design** consider this odd. They think this runs counter to leveraging processes for competitive advantage ([Snow/Miles/Coleman 1992]). In the process oriented design approach, structural underpinnings are necessary for the business processes. However, its aim is an ongoing adoption of structural characteristics to the business processes ([Davenport/Short 1990], p. 23, [Hammer/Champy 1993], pp. 66, 77ff.).

In Scheer's [1995] view of process orientation, an organization's structure is developed from the structure of the business processes, and the hierarchical responsibilities correspond to procedural obligations. Each manager of an organizational unit takes responsibility for specific sub-processes, as well (pp. 8f.). Hammer and Champy even demand a complete replacement of conventional organizational structures:

> Once it [i.e., the organization] is restructured, process teams—groups of people working together to perform an entire process—turn out to be the logical way to organize the people who perform the work. Process teams don't contain *representatives* from all the functional departments involved. Rather, process teams *replace* the old departmental structure (p. 66).

With the help of three case studies, Chapple and Sayles [1981] put "Work Flow as the Basis for Organization Design". Similarly Lehner [1997] argues that discussions about proper placement of organizational activities into units are only temporarily settled, and that organizational charts only conceal what goes on inside the organization.

Self-organization is another popular theme in current studies of human social activity, enterprises, and IT. The term *self-organization* is a label for phenomena which appear to determine their own form and process(es), especially in the analysis and re-engineering of organizations (see [Whitaker 1997]).

The idea is that organizations are not passive and rigid units, but that their configuration and their behavior evolve during the course of their operation. The precise paths of their evolution are largely determined by the organizations themselves. This is most apparent when the members of an organization actively plan and realize its subsequent form. Whitaker notices that recent trends in organizational auto-determination range from BPR to continuous process improvement (CPI), and total quality management (TQM) to participatory design (PD).

Due to the speed of environmental changes, organizations will be forced to develop a capability of continuous reorganization in order to adapt to new situations. Hence, Nadler defines self-organization as the development of mechanisms in organizations to learn from their successes and failures and the ability to redesign themselves due to an analysis of these discoveries. Today, self-organization is an IT supported process with computer tools for organization design that assist design teams with their work ([Nadler 1994], p. 18).

Self-organization produces adaptive organizations, ones that dynamically modify and design their organizational structure, functions, and behavior to fit its evolving external environments. The concept of self-organization is an upshot of studies on autopoiesis, a biology term coined by Maturana and Varela. In biology, autopoiesis deals with ideal cells and biological organisms. This concept has been extended to organizations and societies. Autopoietic systems are self-renewing, self-repairing, unity maintaining, and self-perpetuating ([Kirsch/zu Knyphausen 1991], pp. 78f.). In an self-organizing organization, the manager functions as a catalyst rather than as a director or controller. Management's function is to stimulate the growth of systems and decision processes that operate throughout the organization and to attain goals and objectives ([Whitaker 1997] and [Staehle 1991], p. 61 and p. 893). In contrast, an allopoietic organization is not self-renewing, although it may be self-perpetuating. Structure, function, and behavior are imposed from above, and the result is typically a static, mechanistic, and rigid hierarchy. Decisions and processes are directed toward reinforcing existing organizational structure rather than stimulating new structures ([Whitaker 1995]).

In conclusion, turbulent organizational environments preclude rigid, formal, and enduring organizational structures. Organization design processes that include active, planned activities that only define the organizational structure may be replaced by other design approaches. Due to inherent organizational factors and innovative IT approaches, the result may be alternative organization designs. The next chapter will focus more deeply on the requirements mentioned only briefly in this chapter. The next section will outline the practical foundations of the following chapters.

2.2 Practical Foundations of the GroupOrga Project

Managing a software development project requires the integration of numerous project teams. Generally, these teams use a common information and communication technology. In order to conceptualize and implement the computer-supported organization design environment of GroupOrga, the system is integrated into an existing IT framework and is based on previous developments in technological areas. Therefore, in the following sections the practical foundations and parallel developments of GroupOrga will be introduced and explained. While the necessary methodical groundwork regarding teams, groupware, office systems and workflow management systems was given in section 2.1, this section will present Lotus Notes as a groupware platform for teamwork, and it will refer to the Lotus Notes-based applications GroupOffice (for office management) and GroupFlow (for workflow management)

2.2.1 The Groupware Platform Lotus Notes for Teamwork

Few products sufficiently support the listed requirements for groupware. Lotus Notes is among these, and has been the market leader since the beginning of the 1990s. Over the years, Lotus Notes has strongly influenced and defined the groupware market; however, for many experts and users, it is still difficult to grasp ([Richards 1997], p. 2).

> Lotus Notes® is a distributed client/server platform that allows you to develop applications containing data to be shared by groups of users across a network. It is comprised of a set of document databases that reside on top of a messaging infrastructure. Leveraging the distributed storage and messaging features, the Integrated Development Environment (IDE) provided by Notes™ enables Rapid Application Development and Deployment (RADD) of strategic enterprise-wide business applications ([IBM 1996], p. 3).

In addition to this definition the Lotus Notes groupware platform will be briefly described here by the basic features that are of relevance in the project context and for a general comprehension of the product. Lotus Notes will further also be referred to as *Notes*.

Notes is a collection of databases that contain documents. A document is any kind of *rich text* data, such as text, graphics, video, or audio objects or (cp. [Calabria/Plumley 1997]). The document databases are semi-structured records with forms, fields, notes, subforms, views, navigators, and agents. For detailed descriptions of theses features it is referred to [Richards 1997] and [IBM 1996].

Since Lotus Notes is a groupware platform, *shared access to information* is one of the features. Provided that sufficient access rights are given, users can access a document in a database, even if that document is being accessed by another user at the same time. This lets one user read a report while the other edits it.

Notes is a *network-based, client-server architecture*, which means that one or more servers provide data and programs, while the client computers provide with access to the servers. The server sends the required data via LAN, WAN or dial-up connection to the client.

The *replication mechanism* is an important feature of Lotus Notes. It assists information distribution and synchronization. The spatially distributed databases mentioned in the preceding paragraph do not have to be constantly connected with each other. Therefore, dynamic "copies" or *replicas*, of an original database are created and distributed. The users work in these replica-databases at different locations at different times. Later, the changes and additions are synchronized through the replication mechanism. In this process, the replica-databases exchange only the changes that have occurred since the last replication process.

Additionally, the replication mechanism allows for a detailed specification of which documents will be replicated between which locations. In other words, due to access rights or other definable criteria, some documents may be replicated (exchanged) between replicas, while others may not. This explains why replicas may not be exact copies of each other. Technical details of this replication mechanism can be found in [Kawell et al. 1992], while operation and practical usability can be found in [Nastansky/Otten/Drira 1994], [IBM 1996] and [Richards 1997].

With its network-based architecture and the replication mechanism, Notes is equipped with a *messaging infrastructure*. Information is not only stored in or retrieved from databases in a bi-directional exchange between user and application, but it can also be routed between users and databases.

Support for mobile users relies on most mechanisms mentioned so far. Mobile computers use replica copies of databases and connect to a server through modems. Information can be reached and updated in the same way as a network computer ([Richards 1997], pp. 382ff.).

Notes allows for effective protection of information through a number of *security features* on various levels. In order to secure data in distributed systems and its transport on networks, Notes makes use of encryption in the RSA standard, named after its inventors, Rivest, Shamir and Adelman (cp. [Burnett 1996], [RSA 1997] and [Lotus Dev. 1994]). *Access Control Lists* (ACLs), which allow only users who are explicitly named in these lists to access a Notes database provide security on servers and local workstations. Various levels of access to a database ranging from *no access* to *manager access*, can be specified.

Notes allows *access to external data* (non-Notes data) that are stored on the workstation, on the LAN or on a mainframe. Such data can be accessed several ways: OLE, Notes Field Exchange (Notes FX), @-commands from the Notes macro language, Open Data Base Connectivity (ODBC) drivers to relational databases, Notes C++ API, Lotus Script, LotusScript Data Objects (LS:DO) and LotusScript Extensions (LSX) to relational and transaction databases ([IBM 1996], pp. 393ff and pp. 171ff.).

In conclusion, Notes is a group communication, collaboration and coordination system that is used to share information. The system supports groups of people working on shared sets of documents and is intended for use on a computer network where the database servers may only be rarely connected ([Kawell et al. 1992], p. 226).

2.2.2 The Office Management System "GroupOffice"

GroupOffice, a Lotus Notes-based office management system developed at the University of Paderborn, fulfills the required specifications for office management systems given in section 2.1.4.5. GroupOffice is a comprehensive collaborative system for the more informal type of communication and collaboration in office teams, but it is described only with respect to the GroupOrga approach. Although it provides powerful aids for various office tasks, most of these are only mentioned here. Extensive descriptions of functionality and use of GroupOffice can be found in [Ott/Nastansky 1997c].

Section 2.1.4.5 proposed a general **architecture** which combined structure and flexibility. Figure 2-13 shows how the GroupOffice implementation has made this concept reality. Each user works with a personal, yet replicated correspondence and workflow application (a Lotus Notes application) on local or distributed workplaces, while next to this flexible component there exists a system repository containing the structural design entities in form of a replicated Lotus Notes template database.

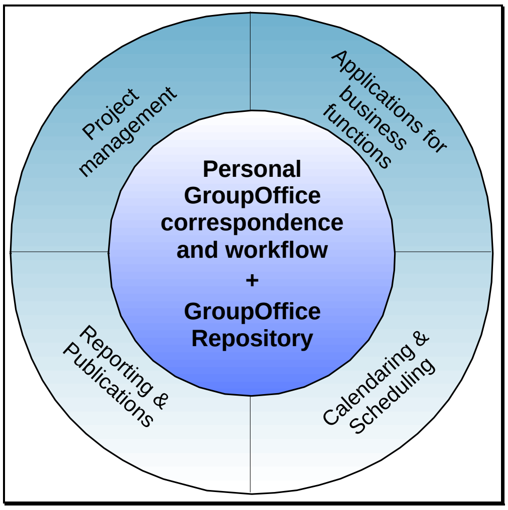


Figure 2-13: Correspondence and workflow in cooperation with system repository

This architecture allows for an object-oriented approach and resembles a separation of design elements from the operative applications into the repository in order reduce redesign efforts and the design complexity of single applications within the architecture. Examples of such design elements are keyword lists, personal letterheads, text blocks, or standard-chapter objects ([Ott/Nastansky 1997a], p. 248).

In addition to its architectural and groupware features, GroupOffice provides most of the **traditional office functions**: text processing, fax creation, directory management, mailing lists, telephone directory, filing, archiving, simple DTP and calculating, access to databases, information retrieval, and so on ([Nastansky/Ott 1996], pp. 43ff.).

Due to the GroupOrga project's direct attention to workflow-support, the **GroupOffice workflow features** will be examined in detail. The management of business processes with GroupOffice takes place in a for loosely structured, informal workflow environment. In contrast to workflow engines for control and automation of highly structured processes, GroupOffice leaves full control to the user. Each office document is equipped with a *Team & Workflow Section* that lets the user specify the next actor and the next task. Hence, in the terminology of the Workflow Management Coalition, each GroupOffice document represents a *Work Item*, which is "the representation of the work to be processed (by a workflow participant) in the context of an activity within a process instance" ([WfMC 1996a], p. 17). The next *Workflow Participant*, "a resource which performs the work represented by a workflow activity instance" ([WfMC 1996a], p. 16) is specified by the current user, rather than having this predefined in a workflow engine. Several Notes *views* provide the workflow participant with the *Worklist*, which is "a list of work items associated with a given workflow participant (or in some cases with a group of workflow participants who share a common worklist)" ([WfMC 1996a], p. 18).

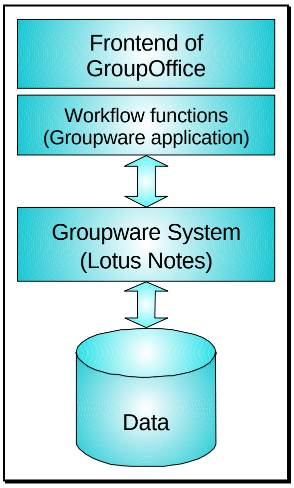


Figure 2-14: Vertical integration of workflow and groupware

From this description, GroupOffice can be characterized as a collaboration environment with initial workflow features for informal workflows. Striemer [1997], strongly demands such an integration of workflow management and groupware for the same reasons; however, he complains that such an architecture is still up in the air (p. C512.04). Striemer presents two concepts of coupling workflow management with groupware and introduces *vertical* and *horizontal integration* (p. C512.06). According to his classification, the GroupOffice implementation represents a vertical integration (see Figure 2-14) with workflow added to groupware functionality.

Alternatively, Striemer requests an integration of workflow and groupware, which supports both strongly structured and semi-structured processes. In this scenario the modeling takes place only in the workflow management system, and loosely structured tasks are carried out in a groupware application. This horizontal integration is presented in the following section.

2.2.3 "GroupFlow" for Structured Workflow Management Tasks

According to Marshak [1995], the major restriction of Lotus Notes groupware is that there is no workflow context in which to easily map out and build sequential procedures. GroupFlow was designed and implemented at the University of Paderborn to provide the workflow development context for building Notes-based processes. The workflow modeler is separate from Notes, but it generates workflow descriptions that are stored, managed and run within Notes databases. This section is a summary of previously published papers, such as [Hilpert

1994], [Nastansky/Hilpert 1994], [Ott 1994], [Nastansky et al. 1995], and [Nastansky/Hilpert 1996].

The GroupFlow workflow management system aims at assisting each of the three basic workflow types identified in section 2.1.5.4 (see Figure 2-10): ad-hoc and task force workflows, semi-structured workflows, and standard workflows. GroupFlow is designed for the middle range of workflow applications. It uses Notes as the runtime and workflow data repository. The external GroupFlow Modeler and other components (written in C++) operate outside Lotus Notes against the Notes Application Programmers Interface (API).

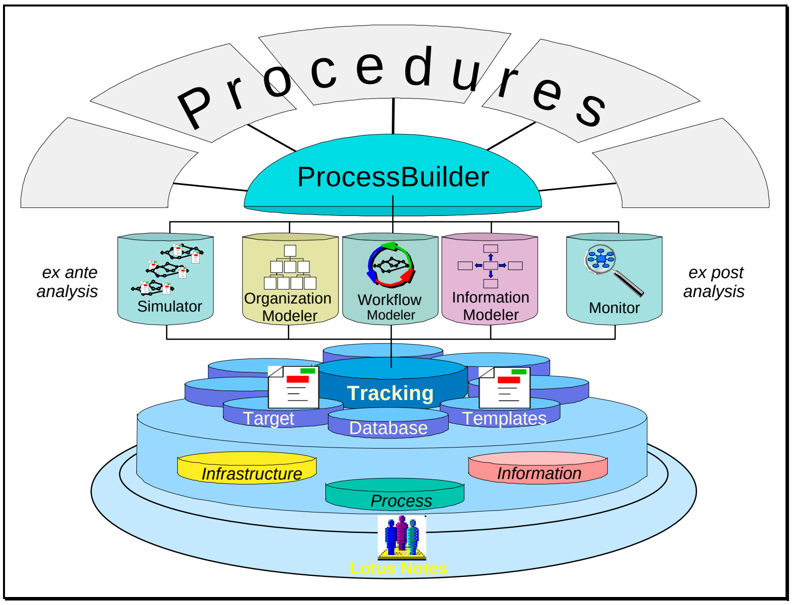


Figure 2-15: The GroupFlow system framework

The architecture of the GroupFlow system framework (see Figure 2-15), consists of three major components. This three-layer architecture complies with WfMC's *Product Implementation Model* as part of the Reference Model ([WfMC 1996b], pp. 11f.):

(1) The *back-end* components manage the structural information of the workflow as well as the messaging and synchronization activities. They include the distributed workflow structure repository and the replication and workflow routing engine. The workflow repository reflects the entities that are relevant to business process design and management. Based on a very simple *enterprise model*, the repository contains structural information about the dynamics of the various business processes, the general organizational structure, and internal application design specifications.

(2) The *GroupFlow target application(s)* include the entities defining the authentic application functions of the business processes that are enabled by the workflow system. Although most of the processing within the system is performed based on form, view and macro/script templates from Notes, they can be customized to organization specific requirements.

(3) The *tool environment,* a set of independent interactive graphical tools, gives the user a variety of workflow related functions, such as workflow modeling, redesigning, analyzing, and simulating. The GroupFlow workflow modeler, the key component in this toolset, supports both top-down as well as bottom-up design, dynamic clustering, update, redesign, and simulation of workflows (cp. [Ott 1994]). Through the GroupOrga OrganizationModeler organizational layout can also be graphically modeled and structured. Any data defining the graphically modeled specifications are stored in Lotus Notes database objects, and can be exchanged across distributed locations using the replication mechanism.

According to the steps in the workflow life-cycle (see section 2.1.5.2), in order to actually use the *GroupFlow* system, deployment phases must be performed. These phases are highly interdependent. In an actual installation, typically the user will toggle between the phases:

(1) Analysis and workflow *concept* design

(2) *Implementation* of the workflow system

(3) *Application* of the workflow system

In phase 1, the business processes of an organization are analyzed and (re-)constructed before the actual implementation of the workflow system. Marshak [1995] gives a detailed description of how to use the various GroupFlow tools—the modeler and its simulation capabilities—for this task (pp. 10ff.). This phase includes the design and testing of a graphical workflow model and the related organizational structure layout.

In order to set up a workflow application, in phase 2, the graphical design of the workflow is automatically transformed into operable workflow definitions. This graphical model (consisting of nodes and edges, embedded properties and attributes) is logically transferred and stored as operable routing specifications in the workflow repository.

In phase 3, users access their work via application databases. Each application database interacts with the workflow repository. These applications are used to activate the document object routing and are designed to be driven by the specifications entered in the workflow repository. The users initiate workflow instances based on the specification and design performed in the former phases. Tasks can be forwarded to the next workflow participant in many ways. Nastansky and Hilpert [1996] describe the standard routing of predefined workflows.

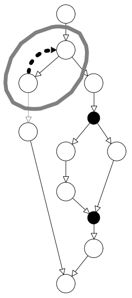
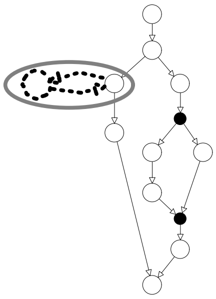
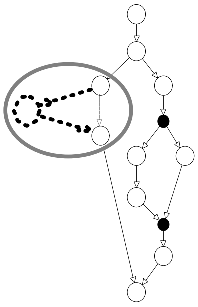
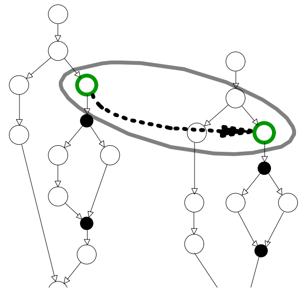
Check-back with previous actor	Inquiry or partial delegation	Complete task delegation	Change of the workflow type
			

Table 2-3: Flexible late modeling and exception handling in the GroupFlow runtime

In addition, the GroupFlow run-time allows for flexible late modeling (ad-hoc modifications and adoption through variants) and exception handling (see section 2.1.5.4) in four different types: *check-back* with previous actor; *inquiry* or question to someone else *(partial delegation)*; *complete task delegation* which invokes a detour in routing path; or complete *change of the workflow type*, where a document is transplanted to a different task and workflow. The *cancel job* option is not depicted in Table 2-3. Here a cancellation of a workflow can be requested and it is up to the supervisor whether to allow a cancellation request.

The participation in a GroupFlow process (phase 3) takes place in Lotus Notes application databases. Standard views (the worklists) are broken down into categories with several categories in each view. These views comprise the list of work items associated with a given workflow participant (or in some cases with a group of workflow participants who may share a common worklist).

Both GroupFlow and GroupOffice are products of Pavone Informationssysteme GmbH in Germany. Recently, in cooperation with the University of Paderborn, the two products were combined in order to provide a horizontally integrated product covering the complete scale of workflow management. Hence, Striemer's [1997] concept of coupling workflow management with groupware, *horizontal integration* (p. C512.07 and section 2.2.2) was realized with ESPRESSO.

Chapter 3
Problem Definition

This chapter offers a situational analysis on the field of traditional organization design in the context of workflow and office management and describes the problem definition for the project.

First, section 3.1 introduces a typical enterprise setting that is the model for the project's research and describes its organizational context. In section 3.2, the technology-based need for the modeling of organizational structures is discussed, and relevant reasons are given for the conceptualization and implementation of an organizational modeling environment. It is shown that an integration of workflow IT and organizational must be reached and that such modeling must take place in a distributed environment with computer-based tools. In section 3.3, process oriented organization design is counterproposed to the traditional organization design approaches, which are examined in section 3.4. Here, literature on organization theories is analyzed and specific problems of these practices are examined. In conclusion, section 3.5 compares these current theoretical approaches with today's tools for organization design. A condensed market investigation shows that most of the drawbacks that have been found in the theories are reflected in the tool environments, as well.

3.1 Situation of the Organization

Chapter 2 showed that new concepts for supporting office and workflow management cannot be applied successfully without a clear understanding of the office environment into which they are placed. in order to narrow down the domain where the results of this research can be useful, this section uses statements from sections 2.1.4 and 2.1.6 to present the understanding of an organization's surroundings and the perspective of the office inside.

The Organizational Context

The situational conditions of an organization are understood mostly as restrictions to all activities to accomplish an objective. Kieser [1993] explains these situational conditions as restrictions that can be neither manipulated from within an organization design process, nor be used as organizational goals to be achieved.

Although these situational conditions can be changed or adapted through non-organizational actions, such as finding new strategies for the organization, for this research, they are deterministic.

The question of which characteristics should be used to describe an organization's situation has been answered in many ways. In English language literature, the term *contingency approach* is the term of choice. Its German name, *Situativer Ansatz*, was coined by Staehle, although the terms *Kontingenzansatz* or *Kontingenztheorie* are also used. While section 3.4.1 focuses more deeply on this theoretical background, this section uses the contingency approach to explain the project's groundwork. Generally, there are four dominant dimensions for describing an organizational context:

- The *environment of an organization* describes the part of the environment that is directly relevant to the organization. It has impact on the organization's structuring and partitioning. For example, when the environment is less dynamic, the levels are generally more hierarchical. This results in specialization and a need for coordination inside an organization ([Kieser 1993], p. 172).
- The *production program* addresses the nature and quantity of goods and services produced or offered by an organization. Producing a larger number of varying goods has an important impact on the organization's structure—this requires more specialization, which, in turn, requires more coordination of tasks ([Frese 1992]).
- The *size* of an organization influences the number of organizational units and their structuring in the hierarchy. Large organizations are more specialized than smaller ones, and they have a greater need for coordination ([Kieser 1993], pp. 169f.).
- The *technology dimension* of an organization can be divided into *production technology* and *information technology*. In office systems and workflow support, IT is the more important aspect, since it focuses on the management of data. The processes of managing data must be seen as a prerequisite for controlling the material processes. Different technologies in this field have dissimilar effects on specialization and coordination, for example, due to their different approaches to delegation of decisions ([Klotz 1993]).

The organizational situation assumed here will relate to the four dimensions mentioned above.

The growing complexity of the *environment of an organization* is currently postulated as the reason for many changes in organizational structures. However, detailed reasons for these statements are rarely found. More likely, one will find lists of observable facts, such as reduction in product life-cycles, increasing competition, internationalization of markets, and higher demands on quality (e.g. [Schmalenbach 1996], pp. 626f.). Many of these facts are not the result of growing complexity (as the number of elements in a system and the number of connections between those elements), but rather dynamics of the environment. That is, current developments do not imply a growing complexity, but only a shift of well-known procedures. On the other hand, empirical studies show that the number of organizational elements does in fact increase. Hence, although a growing complexity is not always valid, in this approach, a constant change in an organization's environment is a valid premise. This section shows which environmental factors are considered for the design of organizations.

This research restricts its view to the parts of organizations that deal with the information processes. The *production program* (the nature and quantity of products or services offered) of an organization, is of average complexity. In other words, the reference organization fulfills average information management processes of traditional functional areas such as: marketing, sales, human resources, controlling, and organization. These processes are of average size, so reasonable coordination is necessary. Information management processes in more complex types of organizational functions, such as research and development (R&D), may also be covered by the concept presented here, but are not the main focus.

Since this project puts team collaboration and coordination in the foreground, measures for *organizational size* are based on the number of posts or employees, rather than on financial or market characteristics, such as turnover or market share. The production program is of average size, which has implications on the characteristic addressed here. The example organization is also of average size; thus, the concept presented here is not aimed at small organizations. This is justified since small organizations have fewer coordination requirements. For example, personal coordination is required more than technocratic coordination due to spatial concentration. For this study, the lower limit of an *average* size organization is 60 clerical employees.

IT was discussed in chapter 2 with the office technologies mentioned there. But the IT used in organizations always lags behind the technological development. The following description of technology in the reference organization must therefore be taken as a model. The status of current technologies in organizations cannot be described here, due to the large differences from organization to organization. This is not necessary, however, since the concept presented here is a new, additional paradigm in the field of office and workflow management. In contrast to the single-user workplace with the standard personal computer, for this project's new concept, the user works with integrated office management systems with organization-wide accessibility. The system supports various organizational and office functions, such as e-mail, information retrieval, calendaring and scheduling. This in-house integration is coupled with public networks to allow worldwide information retrieval and network access. The foundation for this information infrastructure is database technology (in contrast to file architecture).

The organizational environment and production program have a great influence on the decision whether the concepts developed here can be implemented in a specific organization. Hence, these two aspects are examined with another way of classifying the environment. The concept of *causal texture* describes the environment, and helps to determine the conditions for exchanges between the organization and the environment. Emery and Trist [1965] were the first to define four types of causal textures of environments that affect organizations. Some of these were discussed in more recent publications ([Kuutti 1993], [Whitaker 1996] or [Agostini et al. 1996]). The following remarks draw upon these later resources.

The four *ideal* types may exist simultaneously in any real world situation, although their frequency of occurrence may vary. With each of the four types, the degree of complexity of the organization's environment increases.

The simplest type is the *undisturbed environment* where the organization uses trial and error to find the best way to operate. Under these conditions, an organization (such as a small retail clothing store) can exist and adapt as a small unit in one field.

In a second, more complicated, yet *stable environment*, other organizations (such as public organizations) exist in the same environment. However, they do not directly affect each other. In this environment, centralized planning and decision making can lead to improved results.

The third type of environment is *constantly disturbed* by competition and dynamic situations that are due to the unknown actions of competitors. While the undisturbed and stable environment describe a static condition where change is rare and planning is straightforward, the actions of competitors in a constantly disturbed environment must be reviewed. Goals must be evaluated and flexibly must be adjusted because of competitors' actions. A typical organization operating under these conditions is one that has competition in a complex market.

The fourth and most complex type is the *turbulent environment* which adds the influence of a rapidly changing environment where the rules and technologies of today become obsolete tomorrow. The trends that contribute to this include growth and decentralization, increasing complexity and competition.

Regardless of the type of environment, organization modeling has been used to construct models of organizations for the purpose of predicting and estimating the impact of change within an organization brought about by changes in the external environment. The following sections show that while a static enterprise model may have been sufficient when organizations where facing placid environments, a dynamic enterprise model is more appropriate and necessary for the disturbed, turbulent environments. Hence, the concept of this project concentrates on these latter two types of organizational environment.

In conclusion, the assumed organizational context for the research project can be sketched as follows (see Table 3-1):

Type of organiza-tional environment	Disturbed environment with competition.	Disturbed and constantly changing environment with competition.
Production program	Restricted to the information processes of average complexity in traditional functional areas. Reasonable coordination is necessary.	
Size of the Organization	Limited to medium and large size organizations (60 or more clerical employees).	
Information technology	Powerful IT infrastructure, especially organization-wide access to information networks and to integrated databases.	

Table 3-1: The assumed office perspective for GroupOrga

Chapter 2 introduced two extremes of office perspectives, similar to Burrell and Morgan's [1979] division into *objectivism* vs. *subjectivism*. The two perspectives of the office within an organization differ in their focus. For the analytical (or objective) perspective, the focus is on analysis. For the social (or subjective) perspective it is understanding. The perspective of the office in the present case of this project is objective, yet it follows an analytical perspective. In its view the office is largely deterministic and more or less observable. It follows a quantitative research paradigm, not a qualitative one, and it seeks to analyze office operations and functions by breaking them down into their constituent parts.

It is worthwhile noting that most office models found in literature follow an analytical perspective. That is, their view of the office is *activities*, *semantics*, and *functions* (see section 4.2). In fact, the social perspective notes that it is not possible to develop a formal, static model of the office, since its underlying assumption, that offices are not deterministic, negates the possibility of a static, structural model. To fill this gap, this approach tries to implement a dynamic, self-organized approach to organization and office modeling.

As for the three Cs, communication, collaboration and coordination, which was a large part of chapter 2, the concepts developed in this project will be best placed in a *different space, different time* form of *many-to-many* information sharing. Thus, this project focuses on asynchronous collaboration in organizations (see [Rathgeb 1994], p. 49).

There are many publications that discuss new organizational forms. Drucker [1988], for example, describes flat organizations as orchestras or hospitals, due to their lack of middle management levels. Other authors deal with team-based organizations ([Tapscott/Caston 1993]), network organizations ([Sydow 1993]), or virtual organizations ([Davidow/Malone 1992], [Mertens/Faisst 1995], [Moad 1994]). But in these discussions on new organizational forms, the authors use different terms for similar ideas, or they use similar terms for different ideas. This lack of a classification or structuring for the huge number of newly arising organizational forms is noted in [Schwarzer/Zerbe/Krcmar 1995].

The innovative framework presented here does not propose another organizational form, nor does it start from such a visionary point of view. The organizational context and office perspective in which this project's approach can be described as an innovative organization dealing with semi-complex information processes supported by IT. The following sections illustrate why a different approach to organization modeling is necessary. This necessity is derived from current IT (outlined in chapter 2) and is also debated from an organizational and management perspective.

3.2 Technology-based Need for Modeling Organizational Structures

Section 3.1 has already thrown a light on how recent authors have introduced the network, flat hierarchies or virtuality as the organizational form of the future. In industry, multilevel hierarchies have to give way to clusters of business units coordinated by market mechanisms. According to Snow, Miles and Coleman [1992], there are basic characteristics of these new organizations and the forces that have shaped them. These authors, as well as Bradley, Hausmann and Nolan ([1993], pp. 33f.) list globalization, foreign competitors, technological change, outsourcing, accelerated innovation, and deregulation as some of the forces that have shaped new organizations.

The realization of new organizational structures requires the restructuring of information and communication technology. Some organizational elements grow superfluous while others are newly established and have to be integrated into the information infrastructure. Conversely, applying of IT, resulting in the realization of benefits, leads management to demand more sophisticated technology. This cycle has shaped the emergence of both new technologies and new organizational structures. One way to constantly improve the information flow in changing organizations is the use of CSCW technology. It provides services which can be the reason for a restructuring or which can support a restructuring effort as accompanying means.

- ❑ Integration of Workflow IT and Organizational Structure
- ❑ Focus on Flexible Organizational Subsystems for WfM
- ❑ A Data Model for WfM and Office Management Systems
- ❑ Modeling Organizational Structures through Distributed Application Platforms
- ❑ Tool Support for the Organization Design Process
- ❑ Information Technology Reshapes Organizations

Table 3-2: Key points in favor of organizational modeling

Client-server-based architectures for IT in combination with CSCW systems seem to be best suited for these projects. This is based on the perspective outlined in the chapter 2 that CSCW systems seamlessly support teamwork in organizations. WfMS are often referred to as the cure-all environment from the selection of CSCW systems available. In sections 3.2.1 to 3.2.6 a number of problematic aspects, which arise when unreflectively applying WfMS as a means for restructuring procedures in organizations, are discussed. Table 3-2 lists the key points of this discussion, which are the focus of the sections 3.2.1 to 3.2.6.

3.2.1 Integration of Workflow IT and Organizational Structure

WfMSs are widely accepted as a technology that improves productivity, process flexibility, and quality, and reduces turnaround time. However, not only the technology for supporting the processes is crucial for improvements, the users who have to work with it and the organization they work in are crucial, as well. In other words, the question of *who* has to perform a certain part of a process has to be tackled. Moreover, a smooth interplay of the two is considered the most important factor ([Scholz-Reiter/Bastian 1995]). Due to an increase in automation through WfMS, the organizational focus gains importance, and in order to improve the effectiveness of information processes, the organizational potential must also be exploited. A growing concentration on organizational structure in WfM is, next to automation of processes, an important element for IT in offices. However, this point is ignored when understanding WfM as a pure *process* technology, rather than as an integrated solution.

When envisioning new structures in offices and organizations, the correct and up-to-date planning of personal, information and organization resources is of great concern for advancements in business processes. In this context, information about organizational elements, such as units (departments), posts, hierarchy, workgroups, staff deployment, and staff knowledge, must be transparently documented and managed. With computer-based modeling, organizations can easily visualize work steps and processes, information and communication objects, and organizational structures.

So far, the development of IT for process management has touched the area of organizations (especially its structures) only accidentally and in separated approaches ([Nunamaker/George/Valacich 1989]). Comprehensive knowledge about how to simultaneously manage the workflow system design process and the organization design process is missing in research ([Schwarzer/Zerbe/Krcmar 1995], p. 1). This is considered problematic, since on the one hand, there are systems being developed that do not exploit the full potential of today's IT for new organizational structures, and on the other hand, not all possible forms of organizational structures can be tested and evaluated as basis for new CSCW environments such as WfMS.

Moreover, a deeper understanding and integrated design of the two fields is desirable, since IT developers and the people who are responsible for organizational structures are both considered *change agents* in restructuring processes. The new structures designed on both sides—the IT systems and the organizational structures—are not neutral objects, but they interact with and they depend on each other. So far, the theoretical understanding of and the connection between the two fields is not fully developed ([Malone/Crowston 1994]).

Similarly, Schwarzer, Zerbe and Krcmar ([1995], p. 4) also point out that few authors deal with the connection between new organizational forms and IT implementation. Approaches that focus on organization and management with only rudimental technological expertise, and approaches that focus on IT can be found, but they have no impact on organizational subsystems. Although business process management supported by IT is interdisciplinary, research relating to or involving the two academic disciplines is rare ([Hoppen 1992], pp. 5f.).

An investigation of WfMSs and process modeling software ([Mummert 1996], [Mummert 1997], [Kirn 1995], [Bach/Brecht/Österle 1995]) reveals that none ([Sheth/Rusinkiewicz 1993], [Dayal/Hsu/Ladin 1991], [Breitbart et al. 1993]) or few approaches in WfM pursue the design and documentation of structural organization. Nevertheless, its necessity as a mandatory task for WfM is acknowledged broadly ([McCarthy/Sarin 1993]). Mostly rather simple role concepts are intended for this, which may have the advantage in that *ProcessOwner*, *ProcessManager*, or *CaseTeam* can be assigned; however, they are not powerful enough for a comprehensive and flexible connection of workflows and structural elements (or resources) of an organization.

3.2.2 Focus on Flexible Organizational Subsystems for WfM

Any work (handled in workflow systems) is situated activity. In particular, work is situated in an organizational context (as described above), that includes anything relevant and necessary to achieve the goals of an organization. This implies that regulated communication and cooperation between members of an organization relies on organizational structures. Consequently, applications systems for workflow management need information on the organization in order to fulfill their tasks according to the underlying structure. Among others Picot and Maier ([1993], p. 8) and Scheer, Nüttgens and Zimmermann [1995] point out that WfM has focused only on the process aspect; the next step is a simultaneous concentration on processes *and* structures.

But the organizational context is complex and dynamic, reflecting the complex and dynamic nature of cooperative work. The "Arbeitskreis Organisation" of the Schmalenbach Gesellschaft has identified characteristics for the *new organizational context* which require substantially changed formal organizations, demanding highly flexible structures ([Schmalenbach 1996]). For example, Ellis, Keddara and Rozenberg [1995] state that "organizations must frequently make structural changes, such as: adding an new employee, adjusting for a new tax law, filling in for a manager on vacation" (p. 11).

In other words, for WfM, there may be an organizational structure with membership unstable and patterns of interaction that change dynamically in order to face the requirements and constraints of the situations ([Bannon/Schmidt 1997]). Because the *formal organization* is subject to changes and undergoes these changes with different speeds and different visibility, similar to the aspect of late modeling for processes in WfMS (see section 2.1.5.4), the WfMS must allow for late modeling in organizational structures as well.

On the surface, this added complexity could mean giving up an organization's structural elements in WfMS. This strategy is apparently followed by many WfMS who do not have an organizational database or any other component that deals with structural entities. However, following the integration goals from section 3.2.1, this would be a mistake. In addition, business reengineering concepts state that improvements are more successful when processes are completely restructured. If organizational structure is left out and stays untouched in a restructuring, the possible benefit of new processes will be diminished. A comprehensive view on processes and structures must be maintained.

3.2.3 A Data Model for WfM and Office Management Systems

Sections 3.2.1 and 3.2.2 have described that if an organization is to compete in the future through the use of WfMS, it must use systems with the capability of flexibility to support it. The systems must be able to continuously monitor the market, quickly respond through new structures, and quickly modify business processes. But achieving this flexibility requires more integration of functions within the organization than has ever been achieved. Ortner ([1991a], pp. 424 ff.) points out that integration between process management and organization management contradicts decades of management science teachings that in order to cope with complexity, organizations have to be split into manageable pieces, each piece having minimal interaction with others. However, this decomposition in information systems, as well as in the organizations, impedes the free flow of workflow information.

To achieve integration, different systems in an organization must ‘understand’ each other. Therefore, the requirement exists for a shared representation in which the organization can be expressed. Fox uses the term *ontology* to express this need of a shared view of parts of an organization that are agreed upon by people engaged in collaborative action ([Fox/Grüninger 1997]).

Today, in order to reach an adaptation of an office system to the organizational structure, *user databases* register the workers that are allowed to use a particular application. Information from this database is used for identifying users in order to control access. Users may also be assigned functional roles (like *administrator*) that imply special access rights. But different components of an office system use different locally administered user databases, which differ greatly from each other in content and structure. Rupietta [1994] argues that these databases meet technical requirements, not organizational needs. He says that a mapping between technical and organizational views is required in the form of *enterprise organization*, rather than in the form of individual, local user databases for specific system resources.

Although current WfMSs tried to reach this enterprise view (instead of focusing on technical access rights only), their office and organization databases were independently created.

Consequently, they do not share the same representations of *organization*. Schwarzer, Zerbe and Krcmar ([1995], p. 4) stress that numerous practical descriptions have each focused on single cases, rather than choosing a general approach. This has led to different representations of similar organizational structures, and the inability of these systems to share information. Hence, these systems denote the same entity but use different names. That is, though each system may represent the same concept (e.g. unit), they have different names (e.g. department vs. (sub-)division). The authors opt for abstract modeling (of organizations) as a solution to this problem.

In addition, the representations used in today's system models lack adequate definitions of what the terms mean. In other words, although terms are used, they are not defined. This leads to inconsistent use of interpretations and uses of terms. Marshak [1995], for example, moans that the lexicon in the WfM discussion is "enigmatic", using a special case as example.

3.2.4 Modeling Organizational Structures on Distributed Platforms

Organizational subsystems within WfMS support the documentation of organizational contexts, the modeling (build time) and instantiation (run-time) of organizational entities in workflows. For this, enterprise relevant structural information is captured. If it was possible to accumulate this knowledge, for example about the *entire* structure of an organization, then this would be a highly valuable information source for learning organizations (see [Senge 1990]). Thereby everyone is involved when "data is gathered, analyzed, and then interpreted, creating knowledge. Knowledge is disseminated, aggregated, evaluated, and decisions are made" ([Lotus Dev. 1996], p. 1).

Currently, organization design in WfMS is in its infancy, supporting either single-authoring or multi-authoring without any supervised regulations (that is, anybody can work on anything). Hence, actively managing distributed co-design is a crucial issue for enterprise-wide organization modeling that still wants appropriate investigation ([Rupietta 1994], p. 122). To increase efficiency (do the things right) and effectiveness (do the right things), distributed organization modeling should happen in a predefined and coordinated way. Later, this requirement is examined from the theoretical point of view. Here, the technical need for distributing the organizational repositories is strengthened.

The effectiveness of any person involved in the organization design process depends upon the ability to maintain awareness of the current organizational context ([Agostini et al. 1996]). Two attributes characterize this awareness: *visibility* and *transparency*. Visibility refers to the fact that the organization structure is visible and accessible to a worker, whenever and wherever modifications have to be made. Transparency is the attribute explaining that this organizational structure is always ready in normal working situations (at times when no modifications have to be made).

In order to maintain both attributes of an organizational repository, the component in charge of handling organizational context information should be set up in a distributed way. The proposed solution is to leave the structural knowledge in independent and distributed knowledge bases and allow them to cooperate when there is need for conformity. This way modifications in the organization structure (as documented in the repository) may be carried out as they appear in the real organization at different locations. Thus, techniques for distribution of organization databases across networks (for example, distribution according to real organization structure) have to be considered.

3.2.5 Tool Support for the Organization Design Process

This section reveals simple observations about (organizational) design work. First, design work often involves the collaboration of a design team. This means that the work of design is, in part, organized in the interactions between team members. Second, when the result of the design is to be documented in organization repositories for WfMS, designers have to use tools in their design work. For example, they could use organization chart diagram tools for structuring and organizing the design work or they use database tools to list and manage person, unit or user access information in user databases.

But often there is a conflict between the fact that design work is done collaboratively and the nature of currently available tools that may be used to support design. Most of the tools do not systematically take account of the collaborative organization of their work. For example, organization design methodologies are group-independent, and thus do not take into account the numbers involved in the design process. Consequently, when applying tools to a modularized design process, the tools say nothing about how different individuals or groups within a team can collaborate when the substructures are developed concurrently. Yet, as section 3.2.6 shows, communication, collaboration and cooperation between different members of the organization is essential for a successful design process.

Jirotka, Gilbert, and Luff [1992] believe that these contradictions have their origin in the way in which any design process, including the organization design, is seen *in the abstract* as opposed to a real world process. This has separated the design process from the features of working and organizational context in which it really takes place.

Another important concern for organization structure design tools is that of the user interface. Grudin [1990] notes the tendency of our terminology assuming that everything is in reference to the computer. He suggests *computer interface* instead, also referring to a time when computers will reach beyond individual users to support groups and organizations. His position is that with the advent of groupware and systems to support teams and organizations, the focus of the computer interface will extend into the social and work environment. Until recently, documentation and management of organization structures in computer systems (such as WfM or office systems) had reached a state of interface development characterized

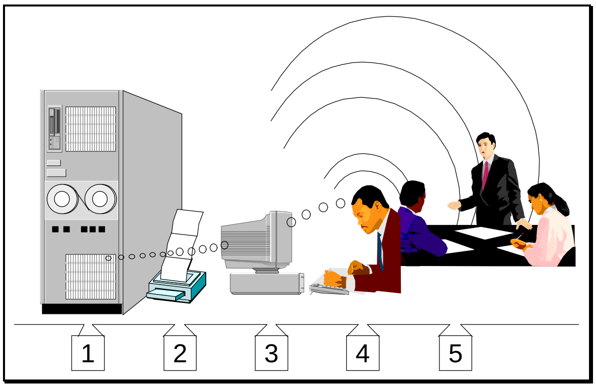


Figure 3-1: Five stages of computer interfaces

by high-level programming languages and environments (②). Currently, the computer interface is the display, keyboard and mouse (③). However, using these graphical platforms, research is increasingly focusing on the end users' conversational dialogues with applications (④) and from there to the support of organizations and the social and work environment (⑤) (see Figure 3-1, [Grudin 1990], p. 262).

In spite of the emerging (and converging) of graphical user interface styles, for end users who deal with organization design these computer interfaces are of little interest if the only form of display is that of an organization chart. Lohse et al. [1994] present a classification of visual representations developed from an exploratory research of different forms of graphics. The result of this research reveals that although subjects believe that network charts convey a lot of information, this form of graphical representation is considered unattractive. Differentiation made in the research describes network charts as showing relation among components. Correspondences among the components are shown by lines or arrows (for example, flow charts, organizational charts and data models).

3.2.6 Information Technology Reshapes Organizations

Section 3.1 outlined that changing organizational structures are anticipated. Among the changes discussed widely are coordination-intensive structures, which some management theorists call *networked organizations*, or more picturesquely an *adhocracy*. Although this structure makes heavy use of rapidly shifting project teams and highly decentralized networks, and consequently is extremely coordination intensive, an opposite point is made here.

New electronic media, groupware technology with its characteristics such as e-mail, video conferencing, and bulletin board systems for example, gives organizations options to work more effectively ([Lucas/Baroudi 1994], p. 11ff.). Unpredictable lateral communication, which was impossible due to its too high complexity, is made possible by the use of IT. Computer networks, for example, can be used to find and coordinate people with diverse knowledge and skills from many parts of an organization. The use of high-level information objects, such as compound documents combined with intelligent object messaging (not only mere electronic messaging) *imposes* new organizational structures, in order to take advantage of all the available technology.

Hence, the process towards networked organizations is driven by Malone and Rockart's two different forces shown in Figure 3-2. On the one hand, adhocracies come into existence due to competition, globalization and other market driven factors ([Sauter/Mühlherr/Teufel 1994],

p. 518). On the other hand, the existence of high-level IT requires more flexibility in organizations, which again makes adhocracies more common. So, it is not only "*market forces* demand → *new structure* requires → *information technology*", but more importantly it is "*information technology* allows for → *new organizational structures*".

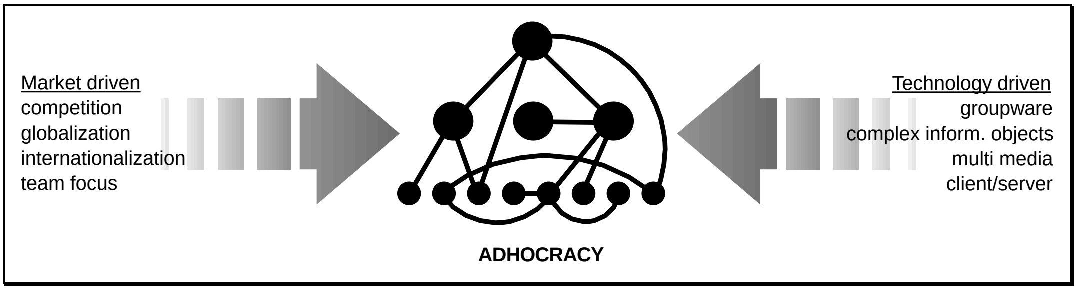


Figure 3-2: Driving forces towards network organizations

For example, decentralization of design work is one aspect which can be considered. Due to security and safety reasons, all organization design procedures are centralized at the organization's headquarters, although, by their nature, they could be decentralized, as well. Introducing a client-server system allows this to be carried out at distributed places, since control can always be gained through the central server. The result is that IT allows for a controllable decentralization ([Lucas/Baroudi 1994], p. 11). Secondly, IT results in a change in functional structuring. Introduction of process management systems shifts organization structures from a functional division of work towards a process-oriented division of work. Integrated clerical work supersedes functional work, which again changes the organization's structure.

3.3 Process-oriented Organization Design as a Goal

The two different aspects of organization, process organization and structural organization (section 2.1.6.1) have long been discussed in theory and practice. However, they have had different importance over time. In practice, the daily work is in the foreground, which emphasizes process organization, while structural organization is (and has been) a task for higher management only. This contrasts sharply with well-established traditions of planning an organization from the top down. In organization theory, the procedural aspect was largely overlooked, while most interest was on organization structures ([Gaitanides 1983], introduction). Only recent BPR discussions and increasing competency changed this situation, so that more often—in the lines of Chandler [1962]—the phrase "Structure follows Process" can be found.

Today's deficits in office organization are due to an over-orientation on one of the two organizational aspects. The office demands a division of labor, rather than isolated workplaces. Consistently, a procedural organization of structure has to take precedence over traditional forms. Current literature identifies bottlenecks and weak points in office work—the amount of time it takes to get a new office worker up to speed, many media breaks, long processing and waiting times for jobs—which are due to a lack of focus on processes and an overrating of organizational structure ([Picot/Maier 1993]). Critics state that current discussion on BPR appears to overdo the turning away from organization structures.

This means that on the one hand, the design of processes can only take place with a certain groundwork of structure in place, while, on the other hand, initial structures can only be defined with a basic knowledge about procedures. In other words, when designing only one aspect, the possibility of designing an adjusted organization is denied and little is left for designing the organizational aspect.

When intellectual abstraction into process and structure is reflected in concrete action, realization of lean and flexible organizations is in danger through this two-stage procedure.

Process-oriented organization design

Hence, without continuous and flexible adaptability of organizational structure in process management systems, the requirements for the process-oriented organization design outlined in the previous paragraphs cannot be met. A fixed structure of an organization, clearly predefined positions, departments and so on, influences the business processes, since necessary relations and structures cannot be changed.

It is crucial that an organization's structure is at least partly aligned to its processes. The processes must be adjusted to structure, and the structure must be adapted to the processes. In conclusion, a process-oriented organization design has both procedural and structural instruments. In the context of this research, this is understood as an orientation of structures on processes, and vice versa.

3.4 Analysis of Traditional Methods of Organization Design

The traditional, academic approach to structural organization design assumes that the design process is a single person's responsibility. For example, it is a manager's or *the organizer's* task to drive, lead and carry out the design of an organization's structure. In order to meet the flexibility requirements of WfMS, the question of "Why structural organization design?" is important, but asking "Why organization design by a group?" is even more challenging.

This section examines today's organization design processes, identifies their weaknesses, and names requirements for these processes to be more effective (Table 3-3, [Rein 1992]). In this section, it is done from an organization theory point of view, in contrast to section 3.2, which did it from a technical point of view. It explains why structural design should be understood as a group process and lists the attributes of such a process.

Traditional organization design	❑ relies on one person's expertise and view (an "organizer") ❑ is based on long-standing, formal methods ❑ focuses on formal roles and structures ❑ ignores existing, everyday business processes which may change
Effective organization design process	❑ relies on multiple views in order to cover the whole problem ❑ is an evolutionary and never-ending process that involves all members ❑ includes formal and informal roles and structures ❑ explicitly includes the day-to-day business processes
Organization design as a groupdriven process	❑ supports solving the problem's complexity due to group communication ❑ allows internal members and external partners to get involved ❑ supports the idea of ongoing process due to multiple process drivers ❑ will be supported through future computer technology

Table 3-3: Weaknesses of traditional organization design and new approaches

Section 3.4.1 begins with a look at traditional approaches. The organization development and the contingency approach is examined further and its advantages and disadvantages are discussed.

3.4.1 A Look at Organization Theories Literature

Organization theory is a way of thinking about, looking at and analyzing an organization. A theory is an attempt to find regularities and patterns in the way organizations are designed and the way they behave. It describes the general patterns and insights into the functioning of the organization. In contrast to physical and biological sciences, which have a well-defined body of knowledge and research involving facts and formulas, organization theory is the study of social systems. Thus, organization theory works with less precise relationships than other sciences. Galbraith [1973] offers a very simple explanation of the purpose of organization theories. He assumes that if an organization were left alone with the forces already set in motion, over time it would move from a state A_1 to a state A_2 (see Figure 3-3). If people in the organization are dissatisfied with the current state, they might intervene and divert the path to state B_2 or C_2. His explanation of organization theories is that they are concerned with these planned interventions.

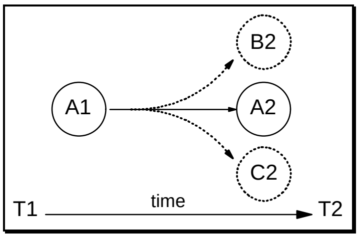


Figure 3-3: Galbraith's view of theory as explanatory means

The study of organizations requires a decision concerning the level of analysis. Organizations are composed of individuals, who are grouped into work units or departments, and segregated by plant or division. Each plant of an organization interacts with other plants and with the external environment. The environment is composed of many organizations all interacting.

In order to study the organization, different levels of analysis are selected by different organization theories. This project aims at changes throughout the system. Therefore, the interest in this context is on the organization as a unit—levels that are higher than the individual and lower than the societal network (the gray-shaded parts of Figure 3-4). With this focus in mind, some opposing perspectives in organization change literature are considered.

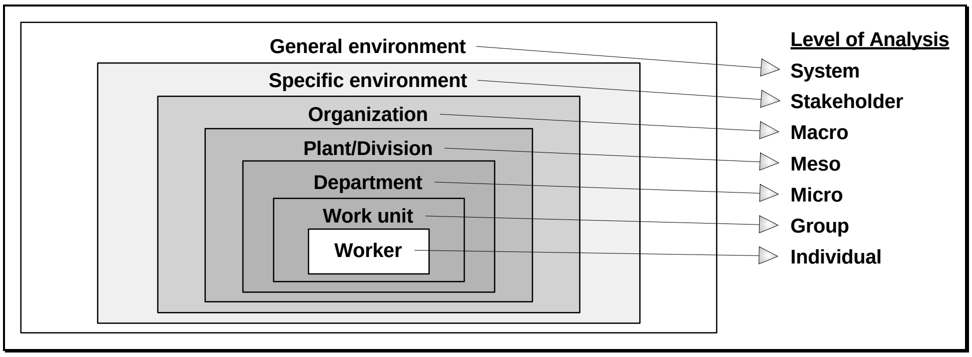


Figure 3-4: Different levels of organizational analysis in organization theories

Organization development or planned-change tradition ([Huse 1980], [Staehle 1991], pp. 846ff.) is the first major stream that is touched upon before dealing with organization theory itself. Organization development puts the members of an organization in the foreground and postulates a self-supported approach driven by employees, rather than by externals or organizers. A main focus of all distinctions of organization development has been on the description, use, and evaluation of specific interventions such as team building, survey feedback, and process consultation. Another focus has been on describing change strategies such as transition management. Yet another branch focuses on the impact of new structures like participative management groups and autonomous work groups. Research and practice in the organization development tradition have generally not been targeted at system change. The usual focus of research and action are the lower levels of analysis.

Kieser ([1993], pp. 113ff.) shows that this stream is highly interested in influencing the practice of organizational change. It includes many normative, *best practice* frameworks that argue for changing organizations in predefined ways in order to increase organizational effectiveness. Another main reservation concerns the relationship between theory and practice. The research upon which normative prescriptions are based is of little help to practitioners who, as Kieser says, want to be supported in complex changes.

The second major stream is that of **organization theory tradition**. Contemporary organization theory is split into opposing perspectives, many of which have been introduced only recently. These perspectives are based on contradictory assumptions about human nature and organizational phenomena. They differ on many points, such as the level of analysis, the emphasis on managerial choice vs. environmental determinism, and the focus on change vs. stability. Notwithstanding recent attempts to give an overview of these perspectives ([Kieser 1993]), organization theory currently offers more choices than a unified framework. Similar to Kieser, four major organization theory approaches are identified here.

First, some theories embrace environmental determinism. They apply theories of biological evolution to organizations and argue that the environment selects entire groups of organizations for survival or extinction due to their greater form ([Kieser 1993], pp. 243ff.). The second set of organization theories emphasizes the behavior in organizations. In this microlevel approach (see Figure 3-4), the level of analysis is the individual or a very small group of individuals. The third set of theories studies all possible appearances of organizations. Most notably it focuses on organizational culture and ignores personal matters. The fourth angle of organization theory is concerned with organizational structuring and thus is the most promising for purposes of this research. It accentuates managerial choice in adapting the organization's design to environmental demands or in altering these demands. Next to contingency theory, which is a well-known representative of the fourth approach, various other perspectives exist. The *resource dependence perspective* states that organizations manage uncertainty created by their dependence on the environment for resources. The *institutionalization school* argues that organizations adopt structures that are seen as legitimate by key environmental actors in order to maintain access to key resources. *Transaction cost theory* ([Williamson 1985]) explains organizational structure in terms of managerial attempts to realize economic efficiencies within the firm. *Open-systems theories* ([Katz/Kahn 1978]) and *structural contingency theories* stress the appropriateness of different organization characteristics in different environmental conditions.

Although, any generalization about such a diverse set of theories is bound to be false, a general evaluation leads to the conclusion that organization theory perspectives are limited as a means for guiding active organization design. Organization theory explains specific dimensions of organizational structure, process or strategy. Contingency theory, for instance, may help to describe and classify an organization's environment (see section 3.1) and how an organization's structure needs to change in a changed environment, but it has little to say about *how to effect* change. Partly because of its theoretical emphasis on *macro* issues, methodology tends to ignore the individual in the organization. Managerial changes in organization structure, carried out by single people in high-level hierarchies, are the main focus.

Recent theoretical approaches to organization design comprise business reengineering, fractal and modular factories, network organizations, virtual organizations, and atomized organizations. Drumm [1996] gives a critical overview of these concepts, grouping them in a "paradigm of new decentralization" stream.

Drumm points out to resemblance in all theories. They present a vision for new organizational structures, are characterized by flat hierarchies, focus on learning organizations, and encourage cooperation in groups or teams (see Table 3-4). However, he lists shortcomings, as well. The paradigm of new decentralization often has no theoretical foundation and is based on an inadequate and idealistic picture of employees. Moreover, as Drumm criticizes, it does not give suggestions for concrete support.

1.	Object, customer and process orientation when forming posts and departments
2.	Flat hierarchies that give much control to a few executives
3.	Higher complexity of tasks in decentralized units, reduction of specialization and tendency to task performance as a whole
4.	Increasing variability and change of tasks over time, learning organization
5.	Cooperation in groups/teams and between teams
6.	Integration of posts and departments by means of communication networks supported by PCs, increased communications needs
7.	Autonomy and self-reliance between posts and departments (self-coordination)
8.	Self-organization of tasks for posts and departments
9.	Self-control for organizational units and their employees
10.	Reduction of interfaces in departments/groups and between posts/employees
11.	High autonomy of departments/groups and their posts/employees
12.	Complementary central steering for decentralized units, at least on a strategic level, central outcome-oriented control of decentralized units

Table 3-4: Characteristics of approaches in the "paradigm of new decentralization" stream

In conclusion, those theoretical approaches of organization design all have their shortcomings, some of which are discussed further in the upcoming sections. Most importantly, few practices give special emphasis to the individual as the most important person for organizational structuring (although organization development does this in part) and few specify how to implement its requirements in today's IT systems.

3.4.2 Problems with Current Organization Design Practices

Section 3.4.1 introduced several theoretical explanations of organization design. Section 3.4.2 identifies and displays four main characteristics of traditional approaches. When these approaches are put into practice, their characteristics result in a procedure for designing structures that would make the redesign ineffective. These traditional concepts:

- Rely on a single person's expertise and line of action (usually a manager or *the organizer*)
- Are based on formalized, *best practice* methods
- Concentrate on previously documented *formal* organizational roles and structures (in contrast to existing structures)
- Ignore business processes

Although each traditional procedure does not show all four deficiencies, several of these problems can be spotted. Each problem is discussed in sections 3.4.2.1 to 3.4.2.4. A groupware-based solution is suggested later.

3.4.2.1 Design Perspective of a Single Person

Because "structural organization was and is often still considered 'a matter for the boss' and happens, against academic better judgment, not rarely ad personam" ([Heilmann 1996], p. 156), in most traditional approaches, organizational (re-)design relies on the expertise of one person. A manager or the organizer is the primary designer. From a technical standpoint, section 3.2.5 showed how current IT practices do reinforce this tendency by not being designed for group use.

Heilmann's objection that this habit takes place *against academic better judgment* is proven by various researchers from this field. Bußler and Jablonski ([1994], p. 82) talk about the *separation of duty*. They point out that different duties, like process modeling vs. organization design, are carried out by different staff. For these authors, a separation of design tasks to multiple people is mandatory. Unterstein ([1994], part 6/8.1, p. 1) says that the development of a complete model of organizations requires entire knowledge of all fields of organizational activity. This knowledge can hardly be assembled by a single person. For Esswein ([1993], p. 554), organization design is a management task which can be found on *all* levels of an organization. It follows that the design of general conditions for organization is passed on to subordinated levels. Ficks ([1986], p. 622) states that in literature it is unclear who has to carry out organization design tasks; however, in the same publication he displays a case history where an approach that had one dynamic leader to restructure the organization in a formal process failed ([Bender/Ficks/Bender 1986]).

Sometimes the organizer is an external professional, who has been hired as a consultant to carry out or fulfill the command and report to management. In either case, new design depends on perspectives of a single person. Organizations are complex systems with a multitude of employees, each with an opinion and perspective of one's own.

Some approaches, especially those that are used by large consulting firms, are aware of this limitation. In order to draw a complete picture of the current state, they compensate by using analysis techniques to collect information from all members in the organization. Based on this information, design is still undertaken by a single person and this person's limited organizational knowledge.

Heilmann's observation of design being "the boss's task" reveals another drawback which has yet to be addressed satisfactorily: to rely on a single organizer creates a situation with a significant weakness. When this one person is not available (due to illness, dismissal, voluntary leave, etc.), the organization may be in a temporary state of helplessness. Moreover, it may then be difficult to find a replacement since professional managers who can quickly come up to speed are rare.

3.4.2.2 Application of Formalized Approaches

Traditional approaches very often cling to cookbook-like practices, which suggest that they could be used in every problem case of organization design. Often such formula solutions can be found in the field of structure analysis, in theories of employee motivation or in modeling procedural work ([Fikes 1982]). Generally, the designer is offered a number of possible model structures and manuals or design guides, from which the organizer can choose a suitable structure for a specific problem. The guidelines consist of a number of detailed criteria lists. However, concrete adoption of the guidelines to real situations is rare.

Wittlage [1995] and Kilmann [1989] have a such *methodical procedure* on a small scale. Whole management books on design guidelines have been published. The difficulty of this approach seems to be a fine line between using an example to illustrate a methodical procedure, while at the same time not interpreting this example as a formula. Interestingly, more recent approaches which deal with flexible organization entities, such as [Esswein 1993], still stick to well-known, yet old and inflexible forms of organization design. His paper, for example, refers to a rigidly formulated 1962 analysis approach from Kosiol. Well-known examples of other formula organization design approaches can be found in the German REFA-Verband (Verband für Arbeitsorganisation und Betriebsorganisation e.V., [REFA 1984], pp. 51ff.).

Some inadequacies of such a *program execution model*, as Fikes [1982] calls it, must be seen. Since organization variety is immense even within a specific business, formula approaches are dangerous since they are inflexible. A formula approach does not account for the variability in the way tasks are accomplished.

3.4.2.3 Formally Documented Structures vs. Informal Reality

Widely accepted, organization design is a formal network that departmentalizes, coordinates, decentralizes, and formalizes roles, tasks, and activities ([Nadler/Tushman 1994], p. 49). The organizational chart is often seen as representation of the organization's design ([Nadler/Tushman 1994], p. 51). Still, this restricted view of only the formal structures of an organization has many problems. Most (if not all) organizations do not function in the way they are laid out on paper. Their procedures and processes orient themselves on informal structures and architectures which evolve between the members of the organization.

In order to support task completion in business processes, sometimes these informal structures complement formal measures by adding new structure to existing structure. They can also come into existence as a contrast to existing formal structure with the aim of protecting ongoing business processes from structural shortcomings which could hinder their successful completion. Since most organization design approaches aim at formal and documented structures only, they rest on a very narrow and static perspective ([Morabito 1995], p. 123). An organization's model that does not account for the difficulties related to working with informally specified tasks, functions and procedures makes a specification of office work unfeasible. Examples of organization design practices that focus only on hierarchies are in [Heinzl/König 1993] (p. 17), [Esswein 1993], and [Hoffmann 1989].

On the other hand, although organization design comprises far more than the formal hierarchical structures, these may be taken as a starting point. The hierarchical structure serves as a form-giving framework for a transformation of informal design variables into observable design variables. It is the basic setting for a development of other architectural elements of organization design, which is discussed later. Nadler and Tushman [1994] see a congruence between the two; they point out that it is not a question of one or the other. In order to decide, they pose a question of how well informal structures fit the formally documented structures. Section 3.4.2.4 shows how this fit can be enlarged.

3.4.2.4 Ignoring Processes

A distinction between the (hierarchical) structure of an organization and the processes taking place inside the organization leads to an artificial separation of the social apparatus *organization*. Such an organization design concentrates on the hierarchies and structures and ignores all process-oriented questioning, which are directly connected with the functioning of the organization ([Picot/Maier 1993]).

In an organization, what can be considered static *structure* and dynamic *process* depends on the time intervals observed. Specific procedures in an organization may be understood as process. When these processes reoccur regularly, one can think of patterns in the processes, and when these patterns remain stable for a long period of time, these patterns may be understood as structures in an organization. In addition, structure depends on the viewpoint. An external viewer might detect a completely different departmental structure in an organization as a department member who is directly involved in the daily business processes. Hence, structural changes alone do not justify adapting to changing market forces. Structure without processes is only half of the design and is not a complete solution.

3.4.3 Requirements of an Effective Organization Design Process

Each of the four problems discussed in section 3.4.2 suggest an area for potential improvements in current, traditional design practices for organizational structures. Such an effective approach to structural design should:

- Be based on multiple personal perspectives, to be equal to the design problem's complexity
- Be an evolutionary (not rule-based) procedure
- Be orientated to informal organizational roles and structures
- Explicitly include relevant business processes in the design attempt

Building on section 3.4.2, these four attributes are outlined in greater detail, since each can be understood as one basic requirement for the team-oriented design process proposed here. After presenting conceptual frameworks and technical aids for the design process, these requirements are developed into an evolutionary, groupware supported design process in chapter 6.

In the sections 3.4.3.1 through 3.4.3.4 arguments on how effective organization design allows for the central idea of *adaptive organizations* are given. "An adaptive organization is an organization that dynamically modifies its internal structure, function, and behaviour so as to maintain congruence or fit with its dynamically evolving external environments" ([de Greene 1986], p. 481). In [Schmalenbach 1996], it is clear that in the past, design was characterized by long periods of relative stability and consecutive short periods of radical restructuring (p. 653). However, by using self-renewal practices, organizations evolve with and adapt to changing environments. In chapter 5, a set of groupware-based applications and tools in connection with a comprehensive enterprise model (chapter 4) is presented to assist participants of this self-renewal process.

3.4.3.1 Using Multiple Personal Perspectives

Organizations consist of many people, each with their own perspective on organizational problems and processes. Thiétart and Forgues ([1997], p. 121) discuss the dynamic interdependencies between multiple actors in an organization. They believe those actors have an immense effect on organizational situation (including its structure). At first, this means that human beings have to be seen as the subjects which have know-how in an organization. However, one actor's know-how alone is not enough. It is the combination of the actor's know-how that results in a comprehensive information pool ([Roithmayr 1996], p. 116). Ellis and Nutt [1980] stress that the perspectives of actors in organizations vary between the levels (for example, high level managers vs. clerks). They say that a clerk may want to see only a view its portion of the work. Although assumption is correct, compared with all other employees a particular clerk would know this particular limited view of the organization best.

Ortner [1991b] centers his investigation around the development of an overall enterprise model. He also demands the participation of every member of an organization in the modeling process, since he expects a common lexicon for the model. This can only evolve if everybody collaborates in this form-giving procedure (p. 273). A common lexicon in the model can improve communication processes by new IT.

An empirical case study, which was carried out in 34 manufacturing organizations that underwent business reengineering efforts ([Hadamitzky 1995]), revealed that success in organizational learning processes is based on the inclusion of all concerned employees of all hierarchies. Hadamitzky interprets this statistical result with a considerably higher penetration of new organizational concepts when employees are actively involved, instead of being passive observers. Though organizational learning is a second step to organization design, his findings should be taken into consideration for this multiple perspective approach.

Malik [1993] examined organizational learning and contrasted two types of management theories: *construction/technomorphic* and *evolution/systemic*. Construction/technomorphic management is based on the idea that an organization can be compared to a machine: it has precisely defined parts and a plan of how these parts have to interact. Function, reliability and efficiency rely on the proper functioning of the parts (p. 63). Evolution/systemic management is based on different groundwork: it is a self-organized, spontaneous regulation which can be compared to living organisms (pp. 64ff.). Malik has revealed seven dominating and corresponding characteristics for both management types, as shown in Table 3-5.

"Construction/technomorphic" type	**"Evolution/systemic" management type**
Management...	Management...
1. is guiding individuals.	1. is the design and leading of complex systems.
2. is leading by few.	2. is leading by everybody.
3. is the task of few.	3. is the task of everybody.
4. is direct impact.	4. is indirect impact.
5. is focused on optimization.	5. is focused on supervision.
6. has sufficient information.	6. never has sufficient information.
7. has the main aim of maximizing the output.	7. has the aim of sustaining the system's life.

Table 3-5: Characteristics of "construction" vs. "evolution" management type

Three of the above dichotic statements support the viewpoint taken here. Management (including organization design) is everybody's task in an organization and everybody has a leading position. Moreover, the responsible people in management levels never have enough information available to optimally complete the design task on their own. According to Malik, every organizational member, regardless of how high or low in the organizational hierarchy, fulfills a certain kind of planning and leading activity. Whenever employees direct or guide other colleagues to a beneficial result in their work, they take a leading position for a certain period of time. In addition, they have to plan and coordinate their own work so that leading and planning is not restricted to the highest levels in the hierarchy, but is everybody's task (pp. 75ff.). Similarly, Malik knows that management never has sufficient knowledge, so the information which a central design process may be based on is always inadequate. He considers it remarkable how much traditionally discussed design methods propose a prognostic approach to organization design (p. 83).

Therefore, an effective, technology supported approach must integrate everybody who decides to actively take part in the design. This also includes the parties which are significantly connected to the organization's prosperity, such as stockholders, labor unionists, consultants, governmental executives, subcontractors and customers (see [de Greene 1986], p. 484 and Figure 3-5). Casonato from GartnerGroup stated in a key-note-speech: "Your innovative systems are not designed for internal users only, because the same system may be the one which you may want to use to interact with your partners and external enterprise members"

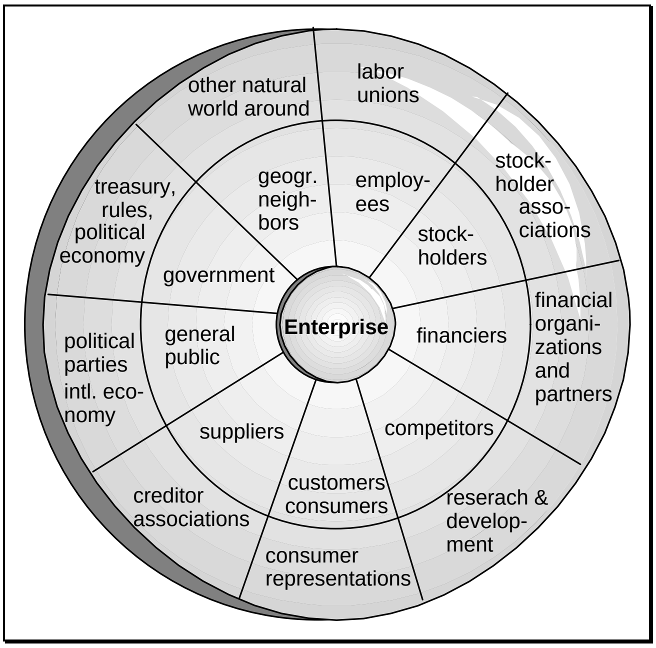


Figure 3-5: Parties significantly connected with an organization's prosperousness

([GartnerGroup 1997a], p. 19). Moreover, current organizational forms cannot afford to stick to a central, and hence error-prone and time-intensive design approach. For competition reasons, they have to react quickly which is difficult in centralized layouts. This approach to organization design, one that involves all members of an organization in the design process, partly answers Roithmayr's question on how human know-how in organizations can be retrieved, archived and transferred to other human know-how owners ([Roithmayr 1996], p. 105). Roithmayr examines the notion of know-how in organizations and know-how owned by the organization's employees. As for organizational structure know-how, a database and enterprise model like the one proposed in this approach can be a means for retrieving, archiving and preserving human and organizational know-how.

3.4.3.2 Organization Design as Evolutionary Approach

The new organization design approach should be an evolutionary (or flexible) approach, resulting in *organic types of organizations* ([Galbraith 1973]). Organizations vary immensely, even within one industrial branch or sphere of activity, and hence the approach must allow for easy self-modification of structures, in order to adapt to ever-changing external, market circumstances. The theory of autopoiesis and self-organization (section 2.1.6.3) is closely connected with this characteristic. The internal structure of organizations is in permanent variation and constantly adapts to changing conditions. In the short run, this evolutionary change is almost imperceptible, while in the long run it results in optimally adjusted structures. Development of organizations happens over time. In other words, organizations learn how to cope with their environment every minute, and they have to adapt their internal communication and coordination.

Although this learning process is often mentioned and described in general terms, only few studies have explained it in detail. Organizational learning must be viewed as a series of adaptations at the individual or workgroup level. These minimal adaptations carried out by individuals grow into evolutionary design changes over time.

The employees' views are most important, because only with their support will the approach turn into a self-correcting and continuous change. Hence, no interference from the top level or from outside is necessary. The employees themselves figure out improvements and ways to realize them.

This step-by-step or evolutionary change process must be seen as valid means for continual structural change processes, which span hierarchies and internal functions. In contrast to sudden reorganization attempts, an evolutionary approach enlarges the employees' acceptance for change and their awareness. Hence, structural and personal development processes are no longer separated from each other. Tool support in this phase of reorganization should always be available, so that modifications and adaptations can be carried at any time. With growing speed in change processes, organizations are forced to constantly restructure. Mechanisms which enable organizations (the employees) to document successes and failures, learn from them and react to them with immediate self-driven reorganization are necessary. Nadler foresaw in 1994 that technology supported tools for organization design would help in quick and responsive creation of new design teams, instantly changed workgroups, adapted organizational roles, and so on ([Nadler 1994], p. 18).

3.4.3.3 Focus on Informal Organizational Roles and Structures

Informal roles and structures should be a focus of the structural design process. Informal roles emerge, prove their necessity, and gradually develop into formal roles and structures. Such a formalization of informal elements is a series of gradual, successive stages. The systematic progression during this team-oriented design process is based on the explicit description of what informal elements exist and how they interact with other informal and formal organizational elements.

According to Knolmayer and Herbst [1993], *business rules* exist in every organization. These rules for execution of specific tasks within an organization explain or narrow down all allowed activities to reach the required business goals. They are established from ethnic or cultural norms, legal prerequisites, and intra-organizational regulations. Knolmayer and Herbst stress that the business rules address internal administrative processes. Most importantly, they argue that business rules are rarely explicitly formulated (for example, in organizational handbooks). Instead rules are implicit and are part of the employees know-how ([Roithmayr 1996] and section 3.4.3.1).

An effective organization design process should increase the congruence between opposite factors in an organization, such as informal and formal organization. *Congruence* must be understood as the measure of similarity between the requirements, the goals and the layout of the two components. In other words, congruence indicates how well the two components fit each other. In terms of formal and informal organizational structure, an innovative approach should allow for explicit documentation of informal roles and structures, so that they can be used for further reference. Nadler and Tushman [1994] see a possible congruence between the two components. They point out that it is not a question of one or the other, because often informal structures supplement formal ones by filling design gaps and inconsistencies. Hence, these informal elements work well in an organization and should be turned into formal ones, since they are necessary and useful.

When changing from manual to computer-based information systems, some of the informal rules are implemented in application programs and database applications. However, these rules may change over time and newer informal structures may evolve. Accessible information (for example, through innovative information systems) has an impact on behavior, and formal and informal changes in the information system influence the organization ([Lehner et al. 1991]). Consequently, each participant in the computer-based information system (in contrast to external system designers) should have the opportunity to take part in the design and to formalize the informal structures.

3.4.3.4 Explicit Consideration of Business Processes

In a new approach for structural design, business processes must be explicitly included in the design process, since they are the ultimate factor for efficiency of an organization and hence the attention of all redesign procedures. Section 3.3 introduced process-oriented organization design which should be implemented in an effective organization design process. More arguments are highlighted here, adding to those from section 3.3.

"Structure follows Strategy" is Chandler's often-quoted thesis, which postulates a connection between structure and strategy ([Chandler 1962]). Several authors have recently modified it to "Structure follows Process". Chapter 2 showed that reorganization along the lines of BPR and WfM driven by this statement does not result in the desired outcome. Rather, a structural redesign process should go along with "Structure *parallel* to Process". By *not* separating vertical and horizontal design measures (structures and processes), two main effects are reached. First, typical hierarchical barriers, which may impede the dissolving of hierarchies or decentralization, are demolished right away during the design process. Second, this parallel approach allows for a simultaneous optimization of structure *and* processes. By improving processes, the necessity for coordination and control is reduced, which allows for lesser hierarchies and a flattened structure. Concurrently, a restructuring of departments and hierarchies frees the processes from routine and makes concentration onto the remaining and important process-interfaces easier.

Davenport [1993] summarizes, "Firms and organizations today tend to be structured in a way that works against the success of their new process designs. Most organizational structures are based either on function or product, with little or no process orientation". Agreeing with the requirement of parallel between structure and process design, he qualifies his statement, "we do not recommend that processes become the only basis for organizational structure...but only when firms adopt more process-based organizational structures will processes be managed in congruence with other aspects of the organization".

3.4.4 The Group Process of Organization Design

In conclusion, this section summarizes why organization design should be viewed as a *group* process. It describes the characteristics of such a process if it is to be effective. The four main motives why structural design should be implemented as group activity and responsibility are:

- The problem is complex
- Members of an organization and other concerned parties (stockholders, labor unionists, consultants, subcontractors, customers, and governmental authorities) have vested interest in solving organizational problems
- Organization design is an ongoing procedure
- Networked computers are standard and provide the technological means to support organization design as a distributed process carried out by dispersed, large teams

3.4.4.1 Complex Problem

The number of tasks and activities in an organization is extremely large. This makes organizations very complex systems. The difficulty in understanding the problem domain "organization" is not just because of the size of the domain. It is difficult because it has many concepts which must be organized and related to each other. It is also because of the complex nature of the relationships between these concepts, such as units, workgroups, actors, and roles. In particular, organization design as a complex problem is complicated by three factors:

- Organization models are partly composed of complex objects, which again are composed of other objects and of the relationships between them
- Organizational objects may exist at multiple (hierarchical) levels of granularity
- Many of these objects overlap, for example, an organizational object (a unit) may share some or all of its component objects (agents) with another object (a workgroup)

Such complex organizations must be precisely understood, so that for an understanding of all these interconnected tasks the common, specialized knowledge of all people engaged in an organization is required.

Davidow and Malone [1992], the pioneers of *virtual organizations*, debate that a reduction of hierarchy levels greatly increases necessary personal control. Decreasing hierarchies down to three our four levels results in a future control span of 50 to 70 (sometimes 200) subordinates. Team design, assignment of project membership to employees, knowledge documentation and role assignment is traditionally a single manager's duty. But an increase in amount of direct subordinates requires other than traditional management methods. Rather, responsibility has to be shifted to groups and teams, since one person cannot manage the large number of transactions well enough to make effective decisions on their organization in future. Galbraith [1977] assumes that the people who carry out the practical work are the people who have the knowledge about organizational peculiarities. He states that "an organization cannot be designed without the people who are to operate within it. Participation of members is needed not only for acceptance of the new design but also to generate the new design which must take account of the many unique features of any specific organization". So, to improve productivity, quality and performance, the people who work in an organization should actively be involved ([Drucker 1991]).

All the above authors stress that computer-based tools can be a means for dealing with these peculiarities of complex organizations and flat hierarchies. In a group process of organization design, this complex problem can be addressed by using distributed IT.

However, this is not to propose that organization design should now be carried out by a small group of *experts*, replacing the *one* organizer used to be responsible for it. Such an approach would not necessarily bring about a significantly improved solution to the problem, since the one person who used to be responsible has now been replaced by a group of people *acting* as one. The approach proposed here is more far-reaching and encourages everybody in the organization to participate as a potential designer. First, traditional approaches were based on the assumption that senior management could design the total organizational structure. Certain business functions were centralized and others were decentralized to divisional or lower business units. In contrast, this project proposes that only key, high level infrastructures should be explicitly designed by senior management. Day-to-day decisions and operations related to getting work done are too dynamic and depend on fast response to diverse customer requirements. So, these decisions would be best left to knowledgeable workers in self-designed networks.

3.4.4.2 Organization’s Members show Interest in Problems

In one way or another, everyone is a member of an organization. When an organization fails to meet expectations, people find ways to disassociate themselves from it. When the membership is in a workplace, people will find a new task, new employer or a new organization. For many people, the success of their organization also means (financial or personal) self-satisfaction. When this is not the case, sullenness and confusion may result. Hackman and Oldham [1976] name three critical psychological conditions for maintaining contentment in the workplace: noticeable importance of the work, noticeable responsibility of the work, and knowledge of the overall outcome of one's own contribution to the work. In this scenario, it is sensible to involve the people with strong interests in the organization's success in the design of organizational structures. Hansen [1991] believes that "empowered people—and with good leadership, empowered groups—will have not only the ability, but also the desire to participate in the decision process." Although Davidow and Malone [1992] believe that some employees may not want to accept such new responsibilities and power, generally, if something matters to people, they usually put in the extra effort. Hansen puts it this way, "When people understand the vision, or larger task, of an enterprise and are given the right information, the resources, and the responsibility, they will do the right thing."

A design process that has all the interested parties participating in it has a better chance to produce qualitatively high solutions and to accepted. "... the ultimate organization designers are those who have to make the design work" ([Galbraith 1977]). Employee involvement has been called a variety of things by different researchers, such as *distributive ownership*, *empowerment of others* and *user participation*.

In order to reach this positive outcome of integrating everybody, electronic means have to be available to share information amongst those who show interest. "Information freely shared with empowered people who are motivated to make decisions will naturally distribute the decision-making process throughout the entire organization" ([Hansen 1991]). An organization model reflected in an organization database (section 3.2.3) can be used as an electronic communication environment, which will allow everyone *inside and outside* the organization to see how they are contributing to the realization of the overall goals of the company. Armed with such a tool, every member can help to develop an enterprise structure which is capable of quickly adapting to its current operational environment.

3.4.4.3 Organization Design as Continual Process

Often organization design is referred to as *a thing*. However, this is misleading. Organization design is not a not recurring matter, but a continuous process. It is an activity that is never completed, because every organization must continually restructure itself in order to reach its goals in changing environments. Consequently, organizations have to change when the environment changes, much of which it does not control. Moreover, they have to be understood as social elements of their environment and each change has implications for the organizational structure. New technology, revised regulations, inflation or other economic reasons, and additional competitors, are factors which cannot be addressed with old mechanisms.

These reflections suggest that organizations should see their design as an uninterrupted process from within. This process of designing an organization's structure may proceed in a repeatable order. Small changes in the structure produce small changes in the costs and returns to these structures, and continually moving from one structure to another, "in the neighborhood" may lead towards an optimum structure for that very situation.

A continual structural design has advantages:

- Little uncertainty about the success of continual steps, since the period between planning and putting the change into action is generally much shorter

- Active support through employees, since continual processes correspond to the human way of thinking

The notion of continuous design of organizational structures goes beyond that of reengineering. Table 3-6 shows a comparison of both approaches, undertaken in [Kaminske/Fürmann 1995]. Continuous design is a long-term organizational process which aims at the temporary stabilization of redesigned structures, while reengineering, in contrast, is a radical short-term measure. Since reengineering is carried out by a few designers (and their small team), participation of employees is suppressed. Moreover, the success of radical reengineering is measured in quantum leaps, while continuos design is measured in small steps.

	Reengineering	**Continuous design**
Organizational frame	Project for improvements	Organizational structure
Measures for improvement	Quantum leaps	Kaizen
Breadth of result	Fast results	Long-term success
Strength	Initiation	Realization and stabilization

Table 3-6: Comparison of reengineering organizational structures vs. continuous design

While the continuos design process has been examined on a micro level (in other words, from a single organization's viewpoint), for reasons of completeness, the macro level ([Schmalenbach 1996]) should also be mentioned here. The authors state that recently, in (German) organizations, much has been achieved through organization design due to urgent calls for action. However, they point out that current design processes are not finished and that the development of changed organizational structures is a permanent task. Thus, on both micro and macro levels, it is everybody's responsibility to participate in the design process. Since organizational structure is not a static constant, there is no time for a single person to analyze, propose, and implement a new solution. Hence, organization design is *dynamic* and *on-line*. The design process is continuous, resulting in open organizational structures, which allow for quick response to environmental changes.

3.4.4.4 Extensive Distribution of Networked Computers

Current trends in IT show that mainframe architectures are no longer appropriate for flexible and fast-moving information needs. The era of networked personal computers in client-server architectures is setting new standards ([Spiegel 1996a], [Spiegel 1996b]). Not only are more and more computers connected by means of worldwide networks, they are getting smaller and portable. In spite of great speed in technological development, for a long time networked computers were used only for simple data exchange and connectivity. The primary means of communication between planners (during groupwork and during coordination of subtasks when designing the organizational structure) are traditional means: internal mail, telephone, circulation folders, and person to person. Since CSCW and innovative Groupware platforms were introduced, workers have exploited its potential to coordinate projects and teamwork. While the CSCW approach has been widely implemented for the support of business processes and procedures through multiple WfMS (chapter 2), it has yet to be considered for structural design in organizations. The availability of distributed computer systems with the characteristics sketched in chapter 2 allow for an employment which has not been thought about before: use for structural organization design projects.

By supporting the design and model-creation of organizational structures within networked computer-based tools, the quality and usability of the models can be improved and the design process may be simpler. In addition, the use of computer tools in networks encourages an engineering-like design process ([Lippold/Hilgenfeldt/v. Kortzfleisch 1993], p. 1). Models, which are managed on computers can be modified more easily. This allows the user to create several variants of a model or to develop new models that are based on successful existing ones. Managing models on IT also allows for their systematic and economical distribution on the network. Hence, networked computers provide the means to realize a group process of organization design, since without them, for everybody to have easy access to the organization model, process definitions or resource lists would be almost impossible.

However, heterogeneity is an undisputed fact in today's distributed systems. Therefore, a successful support of organization design through networked computers requires a standardization and adjustment of cooperation mechanisms. The best way to reach this goal remains unclear. Lengthy international standardization processes (for example, Open Distributed Processing (OPD) by ISO or X.500) stand against pragmatic arrangements between industrial organizations (for instance Common Object Request Broker Architecture (CORBA) by Object Management Group (OMG)). Despite these formal processes, defacto standardization through developers can still result in standards which are not technically neutral and independent of special hardware or protocols. Hence, in this research, an internationally standardized approach, such as X.500 standardization for distributed directory structures, is examined and supported.

3.5 Analysis of Existing Tools for BPR and Organization Design

This section introduces the current discussion on the usefulness of software environments for organization design. A GartnerGroup study introduces an overview of available BPR tools which do (or do not) include structural organization design capabilities. Next, the features of four selected tools are presented.

The tasks and activities of employees who are responsible for organization design (the organizers) are supported and influenced by various software environments. A common description of these tasks does not exist and is probably impossible to generate. The most important attributes were summarized in sections 2.1.6.2 and 2.1.6.3. In addition, office organization, hardware organization and process organization are also listed in literature as tasks for today's organizers. Vossbein [1990] suggests different viewpoints in order to examine traditional organizational tasks:

- Realization of management tasks with software support (decision making)
- Support of organization design with software (substituting traditional technologies such as pen and paper with software)
- Effect of software tools on forms of organizations (centralization vs. decentralization)

The second viewpoint above is the basis for this investigation. Using general organizational tasks as a base, sections 3.5.1 and 3.5.2 assume a homogeneous job description. Available software tools can be classified into various organization design fields: structural organization design, process design, strategic planning, project management, documentation and archiving, and training.

This investigation concentrates on tools that support structural organization design. Section 3.5.2 compares four selected tools against the technical and organizational requirements listed in sections 3.2 and 3.4 and gives a systematic overview of these tools.

3.5.1 Market Analysis

Current literature and tools for structural design are mostly confined to general organizational principles and graphical *presentation* of existing organizational structures. An important goal of the tools should be the active support of the design process. However, most efforts are limited to presentational tasks (for example, graphical display of hierarchies and processes).

Several market analyses document a growing concern in organization design tools. The GartnerGroup, an American research company, did a study on BPR tools in June 1996 ([Lindo 1996]). A similar market analysis on software tools for BPR was conducted at the Hochschule St.Gallen, Switzerland in 1995 ([Bach/Brecht/Österle 1995]). A study on design support systems for *structural* design elements was done in 1993 and published in 1995 ([v. Kortzfleisch 1995]). Tiemeyer and Chrobok [1996], as well as Lehner [1991] tested numerous tools for organization design, also with a focus on structural design capabilities. These studies repeatedly examined tools such as ARIS-Toolset from IDS Prof. Scheer GmbH, BONAPART from UBIS GmbH, Aeneis from ipro Tool GmbH, and Orgline from ALLDATA SDV GmbH. All of the above tools (and others named in the following section) were closely examined. Descriptions of the tools listed in brackets are in chapter B in the additional technical documentation.

Classification of Tools

In general, the existing tools can be divided in four groups:

- Tools for mere presentation

- Tools for design and analysis
- Tool for process optimization
- Complex tools with integrated process functionality

Presentation and drawing tools are the simplest tools for organization design. The aim of these tools is to display organizational structures properly and easily (VISIO, Chartist). Analytical tools concentrate on careful coverage, presentation and analysis of the organization. Processes and structures are evaluated using time and cost factors (Ablauf-Profi, Proplan, ProAS\Process). Process optimization tools optimize previously documented processes (and partly also structures). These tools simulate given situations and propose alternative processes (CAIPLAN-process, INCOME, MOSAIK, PRISMA, Process Charter, SDW-Tools). Complex tools belonging support all areas of organization design: design, analysis, simulation and operation. Moreover, they cover organization specific areas, such as processes, structure, and resources (AENIS, ARIS-Toolset, BONAPART, Nautilus, ORGLINE, and ProzeßMonitor).

In their latest market analysis from 1996, the GartnerGroup divides the existing tools differently (see Figure 3-6 from [Lindo 1996]).

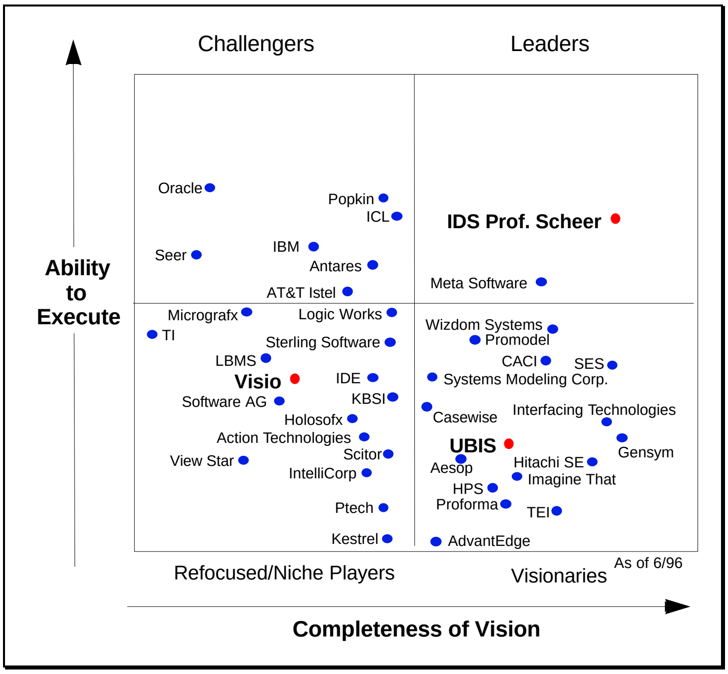


Figure 3-6: Market study on BPR-tools available in 1996

Based on findings from these market studies and own experience, four tools have been chosen for discussion: ARIS-Toolset, BONAPART, Nautilus, and VISIO. ARIS-Toolset, BONAPART, and Nautilus are complex tools, and VISIO is a presentation tool. Each application represents a different segment of the GartnerGroup portfolio (see Figure 3-6, [Lindo 1996], p. 47). Nautilus is new on the market (released autumn 1996) and is included because of its then popularity. For a complete investigation of these tools, refer to chapter B in the additional technical documentation and [Hoischen/Otto 1997].

ARIS-Toolset 3.1 (from IDS Prof. Scheer GmbH, Saarbrücken) is considered the market leader for BPR tools. It has the largest functionality, because it models information systems as well as business processes. Due to its close connection with SAP R/3 and because it provides predefined SAP R/3 reference models, it has the widest spread on the German market. These SAP R/3 reference models may be taken as a guide for SAP R/3 software implementation and setup—a process which is extremely complex.

BONAPART 2.0 (from UBIS GmbH, Berlin) is often named simultaneously with ARIS. The goal of this tool is to make information and communication technology, and new organization designs easier to implement. Although it is complex and training intensive, the literature recommends BONAPART for BPR ([Computerwoche 1995], p. 18).

Nautilus 1.2 (beta) (from integra ISA GmbH, Bielefeld), was introduced in September 1996. Business processes can be modeled based on a comprehensive reference database ("CW-Kompass") that has homogeneous business terminology. The content of this database was developed through a study on numerous BPR plans and projects in selected companies. From this starting point, standard terminology is identified, collected and constantly updated.

VISIO 4.0 (from VISIO GmbH, Munich) comprises no comprehensive BPR functionality, but rather drawing and presentation skills. The GartnerGroup describes VISIO as a niche player ([Lindo 1996], p. 47), since it allows for graphical documentation of business processes, but has neither simulation or analysis capabilities, nor any possibility to connect data objects. Its ease-of-use, low price (compared with ARIS-Toolset or BONAPART) and interface to the full-size BPR tools justifies its inclusion in this examination.

The four tools chosen here (and in chapter B in the additional technical documentation) are mainly Business *Process* Reengineering tools. Nevertheless, the tools were tested based on their structural design capabilities (rather than process modeling).

To evaluate the tools, the benefit analysis technique was used. This project showed that this investigation technique cannot assess the social changes and paradigm shifts that occur when implementing new technologies and concepts, such as groupware, nor can it grasp new types of synergetic cooperation between people. However, benefit analysis can measure and quantify easily observable characteristics in the examined tools and it can classify them according to a pre-defined set of criteria. Hence, this technique is suitable for making preliminary decisions.

Section 3.5.2 evaluates the four tools and presents the results of the benefit analysis. For more detailed analysis refer to chapter B in the additional technical documentation. For a comprehensive introduction into the benefit analysis, which is a specific form of scoring method, refer to [Domsch/Reinecke 1989] and [Scheller 1974].

3.5.2 Assessment and Comparison of Selected Tools

The four presented BPR tools were evaluated according to a list of criteria. This section discusses these criteria, then presents the results of the evaluation.

First, seven categories, covering different aspects of BPR projects, are defined. Because this analysis was conducted with KPMG Unternehmensberatung GmbH, structural organization was not the only focus (see Figure 3-7).

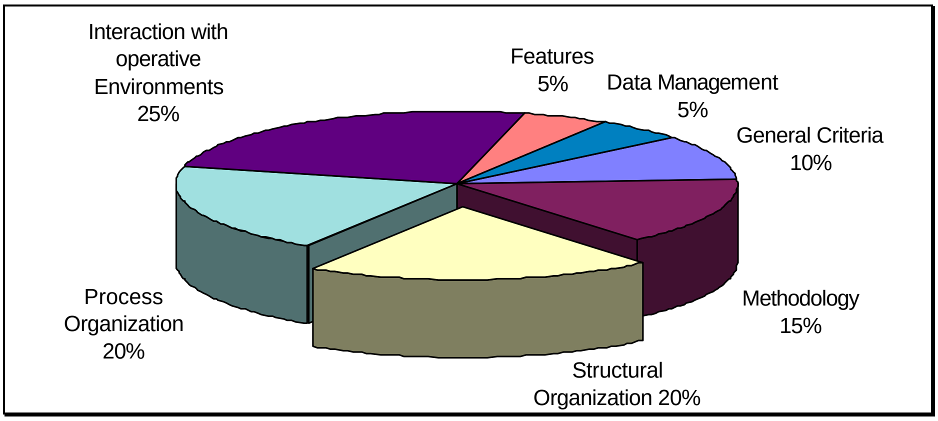


Figure 3-7: Categories of the market analysis and their respective weight

While many criteria in most categories are "standard requirements", especially in the structural organization category, "visionary requirements" may also be found. Examples of visionary requirements are: an organizational subsystem within the BPR-tool (see 3.2.2), an organizational data model (see 3.2.3), support of distributed modeling (see 3.2.4), direct integration or data exchange between BPR-tool and structural organizational data (see 3.2.1), and integrated tool support (see 3.2.5). Since visionary requirements often overlap with standard requirements, there is no explicit separation of the two. However, in order to assess the tool's future prospects, well-known criteria are weighed less, while visionary ones are weighed comparably more.

Four of the seven categories and the pertinent criteria are listed and commented here. The fifth, structural organization, is discussed in section 3.5.2.1.

General Criteria

- multi-platform operation
- simultaneous design through multi-user environments using common data
- availability of repositories and libraries

Methodology

- academic methodology and meta-process model

Interaction with Operative Environments

- can interface with other applications in the BPR tool-family, such as WfM platforms
- can use existing models of organizational structures

Data Management

- stores structural organizational data in tool and platform independent databases
- easy access to databases due to open systems and standards

3.5.2.1 Criteria for Structural Organization

This category (see Figure 3-7) is the most important one in this research. It was weighed considerably high (one fifths of all criteria). Once it is determined that the BPR tool has an organizational subsystem and structural information, the tool is evaluated according to a list of criteria which is listed below. For more information on subcategories, weighing, and criteria numbering refer to Table B-9 in chapter B.

The tools must satisfy these requirements:

Organizational objects (subcategory 31)

- Is the organizational data model generally available and are well-known organizational entities (person, unit, position) available through the model (311)?
- Do pre-defined link-rules for the tool's entities exist (312)?
- Can self-defined model entities (role, skill, location) be established for adapting to enterprise specific situations (313)?

Hence, this subcategory evaluates whether the BPR-tools meet the requirements of today's innovative organizational structures and their data models.

Relations between the entities (subcategory 32)

- Are system-defined relation-types are available (321)?
- Can new relations may be defined and used (322)?
- Are self-defined relation-types checked for consistency in the model (323)?
- Can descriptions for relations be hidden (from the graphical user interface independently) (324)?

Creation of organizational charts and *user-friendliness (subcategory 33)*

- Can organizational charts be automatically generated from process descriptions (331)?
- Does a change in the database invoke a dynamic adaptation in the chart (332)?

- Does a change in the chart invoke a dynamic adaptation in the database (333)?
- Can different model entities be distinguished by their intuitive, graphical representation (334)?

Thus, the direct integration or data exchange between BPR-tool and structural organizational data is in question.

Description of the model's entities (subcategory 34)

Can freely definable text be designated to every entity the model (341)?

Is a description of the entities generated automatically (342)?

Skill management (subcategory 35)

- Does the tool's model have freely definable know-how and skill-entities (351)?
- Can relationships between skill-entities and external documents in order to better define the skill (352)?
- Can input from employees easily be incorporated in the database (353)?
- Are employee qualifications transparent (can available skills and the skill owners be separated) (354)?

Basic data model (subcategory 36)

- Are reference models available (361)?
- Is a predefined meta-model for organizational structures provided (362)?
- Can this meta-model be adapted by the user (363)?

Resource management (subcategory 37)

- Can resources be assigned to organizational units and other organizational entities (371)?
- Can humans and material assets be assigned as resources (372 and 373)?

Access control and *security management (subcategory 38)*

- Does the tool have access control schemes and an organizational model that provides for the visionary concept of distributed organization design (see section 3.2.4) (381)?

Task management (subcategory 39)

- Can tasks be assigned to individuals *and* abstract structural entities (groups, roles, units) (391)?

Organizational analysis (subcategory 310)

- Can structural organization be analyzed and evaluated (3101)?
- How complex is the evaluation and the criteria used in the evaluation (3102)?
- Can design recommendations for process-oriented structural design be generated (3103)?
- Are table used in the results of the analysis (3104)?
- Are graphics used in the results of the analysis (3105)?

3.5.2.2 Assessment for Structural Organization

Although this section explains the results of the benefit analysis, only some of the (numerical) values are compared to each other. The complete results are in chapter B in the additional technical documentation.

The overall analysis of ARIS-Toolset 3.1, BONAPART 2.0, Nautilus 1.2 (beta) and VISIO 4.0 shows that according to the subjective choice of criteria and subsequent evaluation, ARIS-Toolset is the best BPR tool scoring 403 out of 600 points. This is 67%. ARIS-Toolset is followed by BONAPART with 52% and Nautilus with 51%. VISIO is fourth with 18% (see Figure 3-8).

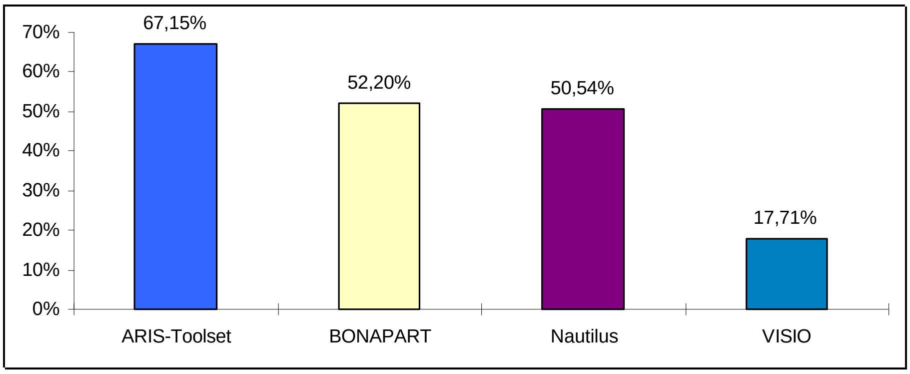


Figure 3-8: Result of overall analysis

Before going into detail, it must be stressed again, that VISIO has a niche position in the BPR tool market. Compared to the other three tools, VISIO has a different type of use, mainly diagramming and conforming to ISO 9000 documentation. Thus, its low score (18%) is due to the fact that some tested features, such as simulation and analysis, do not exist at all. A similarly disappointing result for VISIO is noticed in the *structural organization* category, as well. The following analysis discusses this category closely.

Three of the four products have many structural modeling features or elements. ARIS-Toolset provides a large set of system or pre-defined structural entities—many more than BONAPART, Nautilus, and VISIO. However, only BONAPART, Nautilus, and VISIO can describe user-defined organizational elements in the model. This is a drawback for ARIS-

Toolset modeling, in terms of flexible and changeable structural organization design. Similarly, free definition and instantiation of relationships (edges) between the structural elements can only be done in BONAPART, Nautilus, and VISIO.

None of the tools met the requirement of generating models of necessary organizational structures from given process definitions. VISIO came close—it can generate an organizational chart from a Microsoft Excel spreadsheet or a Lotus Notes database. Through this technique, changes in the organizational structure can be carried out easily via graphical or textual user interfaces.

Concerning the intuitive graphical representation of structural information, Nautilus was the best, and BONAPART was the worst. BONAPART's presentation of large organizational charts is poorly arranged and unclear due to simplified and poor use of icons..

In the category of skill management, again Nautilus was the winner. This product explicitly allows skills or qualifications to be defined for each employee or each organizational unit in form of free-text information and through corresponding keyword lists. The other tools have little functionality of this kind. Only Nautilus and VISION can link to external documents for a description of qualifications.

ARIS-Toolset, BONAPART , and Nautilus have a meta-model of organizational structure. ARIS-Toolset has the most extensive meta-model, while BONAPART has the least. ARIS-Toolset has the most entities in the meta-model, however, no user-driven adaptation of the data model is allowed. This restriction applies to both structural entities and relation between them. BONAPART's model is only a light framework, which can be adapted to the needs of an organization. Likewise, Nautilus allows for adaptation with the extra advantage that intuitive symbolism can be used.

Regular checking on which human and material resources are currently available and who or what can be used in future processes is best supported by Nautilus. Its ability to create reports on the current deployment of employees, organizational units and material resources was impressive. Graphs of these reports and relations can show which employee belongs to which organizational unit and in which processes the employee takes part. Moreover, the tool reveals which material resources are available to which employee for completion of a process or task. Both ARIS-Toolset and BONAPART can create similar information, though with fewer details and a more complicated process. VISIO does not provide availability information for organizational resources at all.

An implementation which would allow for distributed organization design through all members in an organization was not found in any of the tools. ARIS-Toolset has the most complex access control structure, however, it does not have a distributed design. The security levels "Read", "Write", and "Change Access Control" provide for a weak gradation in terms of organizational modeling. BONAPART and Nautilus have only a single security level, and VISIO has no access control at all.

In the field of organizational analysis, organizational structures and processes can be examined in all tools except VISIO. Results can be documented in report or tabular form. ARIS-Toolset and Nautilus have the largest reporting functionality. Nautilus has the better evaluation mechanisms and graphical charting. All reports can derive recommendations for process oriented organization design.

All in all, the structural organization category is dominated by Nautilus, which fulfilled 62% of the required organization design necessities. ARIS-Toolset with 55%, BONAPART with 49%, and VISIO with 22% followed accordingly (see Figure 3-9).

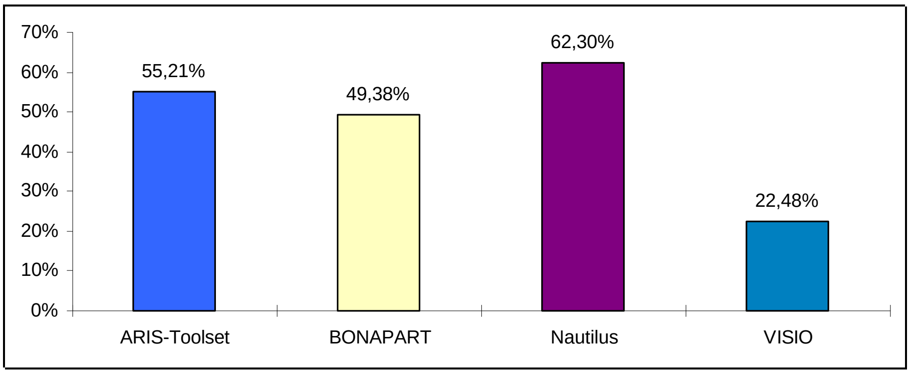


Figure 3-9: Result in category 'Structural Organization'

The delineated result of this category is of high importance to the investigation, and it supports some, if not all, of the above statements on structural organization design in WfM and BPR. While the most polished and *oldest* tools on the market, ARIS-Toolset and BONAPART, earned the best overall marks, they were outdone by the newcomers Nautilus and VISIO in terms of innovation. Structural organization design subsystems as part of WfMS will soon be identified as a critical requirement. Hence, the product developers at integra ISA (Nautilus) are already developing these subsystems. The analysis shows that in Nautilus some of the new concepts have already been implemented. However, in defense of all products, these tools must first fulfill the current market requirements. Hence, all tools, including Nautilus, still concentrate on process analysis, design, and support—an area that is still the most important in industry.

Two more comments are necessary. First, the terminology in the underlying organizational models varies enormously. This is confusing to the user. Table B-2 in chapter B in the additional technical documentation summarizes the organizational entities of the four tools and compares the respective terms. Nautilus' reference database "CW-Kompass" collects uniform terminology for the process aspects of business. Similar uniform terminology for structural elements is an inalienable requirement which can be solved with the enterprise model and organization database that is introduced in chapter 4. The need and acceptance for homogeneous terminology is underscored by ARIS-Toolset 3.2 and BONAPART 2.1, already having implemented an interface to CW-Kompass. Secondly, open, distributed modeling environments and strongly developed skill management characteristics are necessary in future BPR-tools. Currently, the four tools examined here only meet part of these requirements.

3.6 Summary

Several problematic details and contradictory developments in the research fields of WfM and organization theory were discussed in the preceding sections. Based on these findings, it is clear that the currently available organization design methodologies, be it technical or theoretical, do not meet all of the requirements for the operative WfM and office systems discussed in sections 3.2 to 3.4.

The organization development approach, for example, places a high value on influencing the practice of organizational change. Here we find many normative, *best practice* frameworks. However, the main problem concerns the relationship between theory and practice: It is of little good to do research on normative frameworks, since in these situations, complex practical changes are necessary. Organization theory explains specific dimensions of organizational structure, process or strategy. It may help to describe and classify an organization's environment; however, organization theory has little to say about *how to effect* change. This methodology ignores the individual in the organization and proposes managerial changes, carried out by a few people in high-level hierarchies in organizations. Similarly, the paradigm of new decentralization often has no theoretical foundation and is based on an inadequate and idealistic picture of employees. Moreover, it does not give suggestions for concrete support. Autopoietic approaches propose self-organizing systems. Still, in their current form, they appear to be much too theoretical to be implemented in IT.

Therefore, there is a need to investigate these approaches in depth, develop a methodology and software tools, and validate them through case studies in the real world. The realization of some theoretical ideas to organization design in and supported by IT is the focus here. Chapter 4 presents a comprehensive, technology-based enterprise model, explains it in detail, and addresses aspects of its implementation. Chapter 4 is the fundamental technical basis for the software tools, prototypes and products developed in this project, which are presented in chapter 5.

Chapter 4
The GEIMM as a Basis for Office and Workflow Management Systems

In this chapter the *GroupOrga Enterprise Information Management Model* (GEIMM) is presented. For the executing of WfM and office systems, this entity model represents numerous entities of an organization, such as persons (or actors), organizational structures and respective linkages, through roles played. Moreover, it includes office facilities and information (e.g. documents, folders). The connections between the various entities are expressed by specified relationships.

Before explaining the details of the GEIMM, section 4.1 outlines the requirements of enterprise models for executing WfM and office systems in addition to clarifying terms used in the context of enterprise modeling. Section 4.2 examines the different classes of enterprise models and conveys the idea of a multi-perspective enterprise model. Findings from this section are used in section 4.3, which introduces the GEIMM. Here, three different, yet related organizational perspectives of partial enterprise models, processes, infrastructure and information are explored.

In conclusion, section 4.4 describes the use and general structure of an organization database as a computer-based form of storage for the GEIMM.

4.1 Nature and Purpose of Enterprise Models

Models are limited abstractions of reality and effective models provide a means for understanding complex systems. Enterprise models are created, among other things, to help system designers analyze, design, evaluate and implement office or workflow management systems of any kind. Unfortunately, organizations are not easy to model; they are inherently open systems due to communication requirements between operational divisions and the external world, which is essential for the task of coordinating the organization's work. Problems occur when attempting to capture the essence of complexity while generating descriptions (i.e. models) that are readily comprehensible.

One of the difficulties involved in modeling something as complex as an organization is covering all the elements essential to understand an organization, while not overwhelming the model so that it can be understood easily. One way to handle this problem is to develop a unifying framework for organizations, so that real organizations can be depicted in a consistent, coherent way, and at the same time leave room for individual refinement and configuration. The GEIMM intends to offer such a framework without over-automating the problem. The GroupOrga enterprise model is not a theory of office behavior, nor is it based on one; rather it encapsulates the objective requirements and entities of office work, especially those that are important for computer-based support of office work and workflows.

4.1.1 Requirements for Enterprise Models

The first step in successfully designing and developing an enterprise model is selecting premises for the model. Before giving a definition of *model* and *modeling* as used in the GroupOrga project, the following requirements are proposed:

- **Nonconflicting terminology.** An enterprise model should allow the designer to express the various operations of an office in a natural and straightforward way, i.e. in terminology suitable to the field. For example the terms *unit* and *position* have specific meanings in organizational structure, and these meanings should be used in the enterprise model.

- **Simplicity vs. expressiveness.** The model should have the smallest possible number of structure types, composition rules, and attributes and still be able to express all relevant situations. It should be as simple as possible for a given modeling situation.

- **Decompositionability.** Individuals at different levels of an organization are interested in varying degrees of detail. Consequently, the model must allow for network and hierarchical decomposition, i.e. gradual exposition of detail of the system being modeled. For example, the unit manager needs to know all the details of the specific unit, while the plant manager usually does not have to be concerned with such details. Should the plant manager choose to access the details, however, the model must permit an easy and consistent way to do so.

- **Picturability.** One means of enhancing communication is the use of graphical aids, so the chosen representation must facilitate graphical representation. A screen-oriented interface displaying the model will promote human-computer interaction (see section 5.5) and lets the model to be understood and used by a wide range of non-technical, non-professional users.

- **Partitionability.** Communication between individuals and groups in different locations and departments of an organization is essential for advancing the always up-to-date model of an organization. Thus the model should have structures which facilitate the administrative partitioning of data and thus it must provide means for maintaining consistency across the models if there are multiple models or model segments.
- **Flexibility vs. rigorousness.** Flexibility is a major concern since an organization is complex and all the problems faced in it are ill-structured. Consequently, the evolution of the model cannot always be an orderly process and the model should therefore provide sufficient freedom to the modeler in developing the model. However, such flexibility should not result in an incorrect model of the organization: Flexibility and ease of use are essential for widespread acceptance, but these attributes should not be achieved at the expense of rigor. A rigorous model with well-defined and unambiguous syntax is a prerequisite for ensuring correctness and precision in modeling and workflow implementation.
- **Executability (for workflows).** Another important requirement is that the model should be executable, i.e. it should generate representations of entities which can actively support computer systems for control and simulation of office management procedures. Owing to its importance in the GroupOrga project, this particular aspect is outlined at greater length in section 4.1.2.

Picot and Maier [1993] give reasons why enterprise models are necessary in IT from a *transaction cost* point of view. A basic assumption is that the greater the uncertainty in a workflow, the greater the amount of information that must be processed among decision makers during its execution. The basic effect of uncertainty is to limit the ability of the organization to pre-plan workflows or to make decisions about activities in advance of their execution. According to Picot and Maier's argument, enterprise models provide structure and serve as a base for the effective coordination of business tasks, e.g. within WfMS. The use of such precisely defined enterprise models aims at the reduction of coordination and communications costs for the actors involved in the business process. Hence, an additional goal of such modeling efforts is to reach comparatively low transaction costs or respectively to keep exceptional information needs as low as possible (p. 11).

4.1.2 Enterprise Models for Workflow Execution

As the last section suggested, an enterprise model is not intended for documentation and information purposes only, moreover it should play an integral part in the overall architecture of a workflow management and office system.

Since the GroupOrga project is embedded in a comprehensive workflow and office research project, an enterprise model that is applicable to workflow design and processing has to provide interfaces for workflow applications to access organization-related information. By accessing these interfaces, all workflow applications within the groupware environment can make consistent use of the organization design elements. There are three reasons why such a model is necessary for corporate workflow modeling:

- **Cooperative workflows** are processes involving a group of persons in which each person is responsible for one or more actions in the procedure. The responsibility of an actor is inferred from the role this actor is assigned within the organization or from the workgroup or unit this actor belongs to. Assistance of ongoing office procedures requires information on the organizational structure and the roles of actors or their belonging to groups or units, as well as information on processing steps to perform. The enterprise model is intended to provide the organization related information.
- **Addressing** is a critical issue for workflow systems. The system needs to address users on different machines and in different locations for performing different tasks. But e-mail and network addresses are mostly designed to meet technical needs and capabilities and not organizational requirements. Therefore an enterprise model uses organizational elements such as roles or workgroups for addressing users.
- **Access Control** is an important feature of systems for cooperative processing. The cooperation of multiple workflow participants results in parallel and independent modification of process information. Access to information objects should only be given when a user is addressed to perform tasks in order to avoid conflict and corruption of data. Access rights depend on the assignment of editing rights on information objects to entities such as actors, roles or organizational units at the right time.

Thus, the term *execution* (or *enactment*) refers, for example, to a system which carries out an organization's processes. Having taken the effort and the expense to create the enterprise model for insight or communication, an organization does not want the added expense of creating separate models to help in building its process enactment systems: it makes sense to (re-)use the same model. Reuse also helps to ensure consistence between the model of what is meant to happen and what really does happen in the workflow and office system. If this consistency can be achieved, than change can be initiated by changing the enterprise model.

4.1.3 Definitions: Model, Generic Model, and Modeling

Given the difficulty in defining *organization* (see section 2.1.6), the notion of model has no universally agreed meaning. In contrast to the preceding sections, where requirements for enterprise models in general (and in combination with WfMS in particular) have been named, this section gives a clear definition of what a *model* in the GroupOrga context is, why a *generic model* is considered necessary and how *modeling* takes place.

"A *model* is a system, which purposefully depicts another system" ([Sinz 1996], p. 125). This informal definition alone, already introduces the three components which belong to the term *model* (see Figure 4-1). In [Ferstl/Sinz 1990], formally a model consists of:

- An object system S_O (original system)
- A model system S_M (depicted system)
- A modeling function f: $K_O \rightarrow K_M$, which transfers the set of elements K_O from the original system S_O into the set of elements K_M, making up the depicted system S_M

In the context of this project, the *object system* as shown in Figure 4-1 is an organization in general (its processes, structures and information objects), i.e. a detail of a real system. The corresponding *model system* is a formal system, i.e. the GEIMM implementation to be introduced in this chapter. Although, these formal functions are implemented in the applications of the GroupOrga project as well, such formalism in the modeling process is intended to be hidden from the end user.

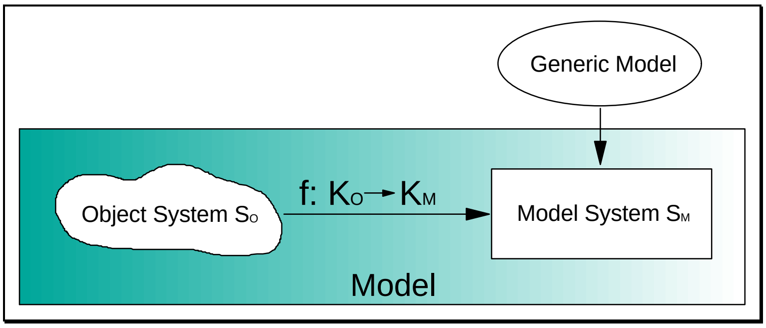


Figure 4-1: Model and generic model (meta-model)

It can generally be said that most information systems incorporate a model of some aspect of its structure, processes and information objects. The problem is that the legacy systems that support enterprise functions were created independently and, consequently, do not share the same models. Fox [1993] calls this the *correspondence problem*. Though each enterprise model may represent the same concept, for example *task*, they will have a different name, for instance *operation* vs. *activity*. Consequently, communication among systems based on different models is not possible without translation. No matter how rational the idea of renaming the concept is, organizational barriers often impede it. Secondly, these representations lack an adequate specification of what the objects (terminology) mean (*semantics*). This leads to inconsistent interpretations and uses of the model's inherent knowledge. Lastly, the cost of designing, building and maintaining a model of the enterprise is large. A solution to this problem leads to the term of *generic models*.

Generic models are models which are not built for a specific purpose: the implication is that they can be used for different purposes and with different tools. Often the term *meta-model* is used instead, also noting that such a meta-model represents the general structure (i.e. relations) of a given number of general entities which are somewhat in connection to each other. In this project both terms connote the same, yet *generic* will be used almost throughout.

Since the object system S_O is a real system, due to changing environments its set of elements cannot be specified formally. However, for the model system S_M this is practicable. Its *possible* set of elements K_M is specified in form of a generic model (see Figure 4-1). Hence, a generic model represents a type definition for a class of specific models. For example, the specific enterprise model of the firm *Miller Inc.* is an instance of the generic GEIMM.

Generic models are somewhat reusable, constituting a common view or perspective of a particular subject or domain, in this case the organization. The generic GEIMM proposed in the context of GroupOrga helps to provide a means for integrating what is currently a disparate set of modeling techniques and tools. It is both semantic and executable, i.e. it helps clarify the meaning of the terms used and supports task execution (see section 4.1.2).

Modeling, by extension, is the process of constructing a model, i.e. encompassing the activities of capture and description of all relevant aspects of the model. Since the two terms *model* and *modeling* have the same root, this term is used more frequently in connection with enterprise models. In the GroupOrga context, the terms *design* and *designing* will be used just as often (see 2.1.6), since in this context there is no difference between the two expressions and hence they can be used as alternatives.

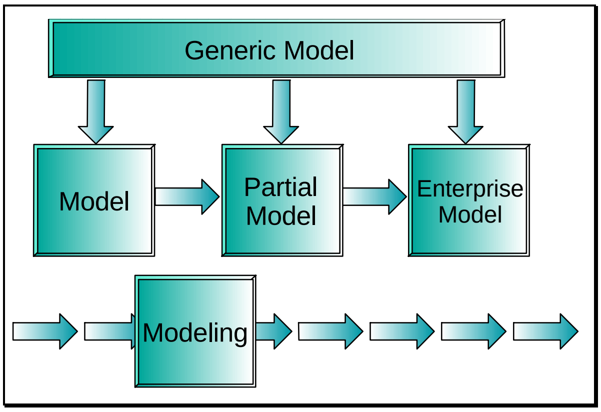


Figure 4-2: Context of enterprise model connotation and terminology

Enterprise models may consist of several *partial models* which relate and which make up the complete model. Information models, for example, represent a connection between the overall enterprise model and the organization's informational infrastructure and communication technology. So to speak, the information model mediates between organization and IT. It consists of descriptions of all entities concerned with the storage, processing, creation or dissemination of information in the organization. A characteristic of information models is the abstraction of physical and software components, such as computer systems, programming languages, databases, networks, and so on. Hence, information models do not depict physical manifestation, but the essence or the logical aspect of information processing. In section 4.2 three partial models, including an information model, which may make up an enterprise model will be introduced in further detail.

After these preliminary definitions and explanations as depicted in Figure 4-2, the following definition by Fox and Grüninger has been chosen from the many existing, to lay the groundwork for the GEIMM:

> An Enterprise Model is a computational representation of the structure, activities, processes, information, resources, people, behaviour, goals and constraints of a business, government, or other enterprise. It can be both descriptive and definitional—spanning what is and what should be. The role of an enterprise model is to achieve model-driven enterprise design, analysis and operation. ([Fox/Grüninger 1997])

Concluding, an enterprise model is always a simplified representation of reality that can never capture the full complexity of it. However, it can be useful in describing one or more dimensions of reality. Slovin and Di Nunno ([1994], p. 47) ironically state "All models are wrong, but some models are useful". They implicitly capture the purpose of enterprise models: to obtain a comprehensible insight into the structure and function of the system *organization*, which is to be made available for information system design.

To the best of our knowledge, little has been done so far to use the above constructs to model and develop a comprehensive generic enterprise model—especially, for the purposes of workflow enactment and active integration with WfM and office systems. This is what has been done with GEIMM. Before it will be presented, the next section examines classes of enterprise and office models to underpin the above statements.

4.2 Results from Investigating Classes of Organization Models

Section 2.1.4 has focused on different office perspectives that exist and has explored the notion of *office*. While different theoretical perspectives have been examined there, this section considers how these perspectives may reflect in forms of organization models.

4.2.1 Types of Current Organization Models

In the beginning the development phases for offices systems where similar to those found in conventional systems. Over a decade ago, Bracchi and Pernici [1984] proposed a procedure shown in Figure 4-3. In the requirements analysis phase, the organization's reality is studied. These requirements are then formally specified using a conceptual model of the problem field. After the requirements specification phase, an office system design is generated which meets the identified requirements. Next, the system is put together and implemented. An ongoing evaluation process is used to monitor the development, generating modifications when necessary.

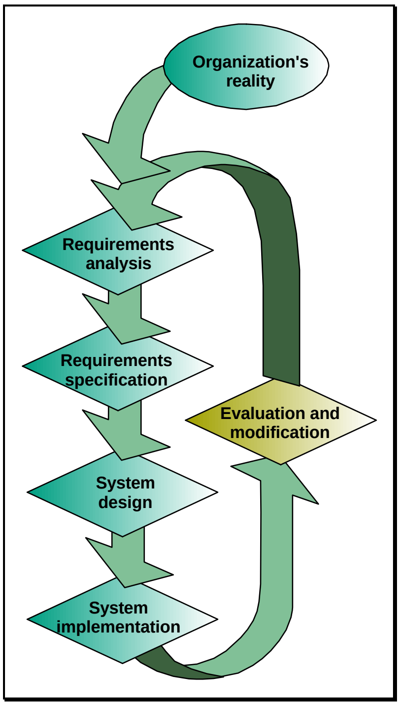


Figure 4-3: Phases in office systems development

Due to this process, much effort to date has been focused on the problem of formally specifying single organization or office entities and aspects in a model, rather than the development of a complete generic model (see for example

[Tueni/Jianzhong/Fares 1988] and [Lochovsky et al. 1988] on tasks or [Kozar/Zigurs 1992] and [Esswein 1993] on roles). Because information systems were developed as isolated islands from the larger organization, many existing enterprise models are really only partial models, emphasizing only a subset of the whole.

There are few existing models that offer complete frameworks, instead most emphasize the subsets of an enterprise model. In the following, several design methodologies are briefly illustrated and some of the most relevant conceptual models are outlined. This is not meant to be a survey, or even an exhaustive summary of these models but an attempt to identify the embedded concepts in them. Brief descriptions are presented while details of the models may be found in the references cited. The choice of model types was motivated by the intention of considering a wide representative range of past work in this area:

- **Information-action models** focus on the information used in office work and the actions based on the information. They try to depict office work in terms of parts of information (like files, memos, forms and so on) that flow between offices in an organization. These models are concerned with issues such as what information the action needs as input and what information the action produces as output. Models like these are useful in defining the types of organizational units engaged in the work and the operations that each unit has to complete regarding a specific information object. In contrast to the following approach where the focus is on processes, these models emphasize the information object (e.g. the data file, document, memo, note, folder) and determine a workflow on a step-by-step basis, when the information is handled. Examples: [Kreifelts/Woetzel 1987], [Jablonski 1992]

- **Procedural/process models** attempt to represent office work in terms of *procedures* or *processes*, i.e. as a number of predefined sequences of steps, that are executed by office workers. They involve operations (process steps) and operands (units of information). Procedural models stress the tasks-orientation in the sense that each procedure is designed to perform a certain complex task. They identify the roles played by the agents within the procedures. These models analyze and describe office work by looking at different activities performed by the users and the system. Examples: [Medina-Mora et al. 1992], [Nastansky/Hilpert 1994]

- **Decision-making models** relate to the activities of mangers and other office personnel. These models examine the enterprise from the viewpoint of the active elements of the organization: the agents. They associate a set of functions to the agents, such as the roles they play, the area they act within and the relationships that exist with other agents. Decision making models that focus on a particular agent, or on a role that an agents plays, model a dependency of office data and activities on the basic element in the system, the set of office workers and their organizational structure. Actions that are performed automatically by the system are treated as those performed by particular agents. Example: [Karbe/Ramsperger 1991]

- **Database models** refer to office work as being modeled in terms of databases in which information records are created and manipulated. Database models group data into forms, which can be compared to paper forms in the traditional office, and reports are generated containing the contents of the databases. Different types of data and the operations on data objects are the basic elements of this type of model. Business processes are understood as a series of operations on data. Data structures used in existing models are two-dimensional objects, e.g. forms, reports, hierarchical structures and documents. The main purpose of these models is to oversee objects manipulated by office workers in a way similar to traditional offices. They support the work of a single user at a time, connecting the users through a communication network. This reduces the workflow capability because the flow is not under system control. Examples: [Gray/Reuter 1991], [Malone/Fry/Lai 1992]

In an assessment, this summary shows that *information-action models* are declarative and define rich structuring primitives for abstracting a variety of office information. This model type is unique in the kinds of office information it addresses: ranging from text and forms to images and voice. Being a data model, no mechanisms or techniques are provided to describe office work or office agents. The model gives a static description and dynamics is scarcely captured. Data types with documents are usually well defined in *database models*. Unstructured data types, concerned for instance with time aspects, are presented in the most recent database models, but are currently only dealt with in process models rather than in today's database models. Different types of tasks are barely handled in database models, while process models allow an excellent description. In *procedural models* control data (e.g. time) is necessary for the definition of control flow aspects while database models do not include this aspect. In most of the process or decision making models it is possible to describe complex and unstructured tasks. Communication is characteristic of database models (via information sharing or message enabling) and exception handling is more characteristic of process models. Office agents and their hierarchies are hardly considered in procedural models. Neither the varieties of office information nor their structuring are dealt with in detail. In *decision making models* no primitive is provided for the management of office information alone. The association of agents to their functionality and decision making based on information is strong. Since an agent is the main basic entity, it appears that information or activities can hardly be specified in the absence of an agent; thus, the model does not represent office information explicitly.

Reflecting on the distinction between objectivism and subjectivism in office perspectives presented in section 2.1.4.1 and discussed in section 3.1, all of the above model types tend to share the objective or analytical perspective. In other words, a representation that seeks to analyze office operations by breaking them down in their tangible, constituent parts is chosen, rather than one that focuses on the understanding of the social actions and meanings of the participating actors in a social setting. It will become clear in this chapter, that the GEIMM also opts for the analytical dimension, however, some of its entities, such as *role* or *group* allow to represent a social, more qualitative office perspective.

The above models of cooperation embody a range of assumptions in regard to why people work together in an organization and often characterize how they should work. The commitment involved in these assumptions is often problematic when these models are practically used in work settings. One response to the experience made with the practical implementation of the model may be a focus on the development of more flexible models of organization.

4.2.2 Approaching a Multi-perspective Enterprise Model

Desai [1991] distinguishes between three types of models, which are somewhat similar to, yet distinct from, the classification made above. He proposes a coarser, however correct systemization according to the *basis of the concept* on which a model is centered (p. 43):

- Object-based (organizational information)
- Activity-based (organizational work)
- Role-based (organizational actors)
- Mixed models

For Desai, models based on a single aspect of organizations are often found to have deficiencies in capturing the other aspects. The first three types each emphasize such single aspects of an organization. Mixed models view organizations from a more general perspective and include more than one characteristic, independent of each other.

More often, in recent modeling approaches the three concepts organizational *information*, *work*, and *actors* have been identified as essential aspects of an organization. However, most of the existing models either barely consider this distinction at all or, if they belong to the group of newer modeling approaches, are based on one or two of the essential aspects and therefore do not capture all the nuances of an organization. Although in [Li/Lochovsky 1996] different terminology is used, their fourfold modeling perspective also indicates the new direction for organization models (p. 193):

- Data/knowledge modeling (data and knowledge created and used)
- Activity modeling (dynamic aspects of organization)

- Organization modeling (structure and actors)
- Resource modeling (working materials)

Such *multi-perspective approaches* observe and depict the organization from different viewpoints, rather than from one single angle. The GEIMM will be built to allow for a coupling of different perspectives of organizational circumstances; the traditional modeling restriction to either activity-related, information-related or actor-related aspects will be overcome by modeling in terms of a *process* model, an *infrastructure* model and an *information* model simultaneously:

Process modeling perspective. A process model is used to define the activities and tasks to be executed in a business process. Moreover, process models define the (chronological) order in which tasks have to be carried out. Process models are described by means of directed-graph structures, which allows the user to define which tasks have to be carried out sequentially, concurrently or alternatively. More details about process models are revealed in section 4.3.1.

Infrastructure modeling perspective. An infrastructure model is used to define which organizational entities are involved in a business process. For example, roles can be attached to organizational tasks. Infrastructure models in GEIMM are chiefly described by means of organizational diagrams. The relationship between organizational entities, such as roles, persons, units, etc. can be defined. The top level part of Figure 4-4 sketches an infrastructure model. Further details are given in section 4.3.2.

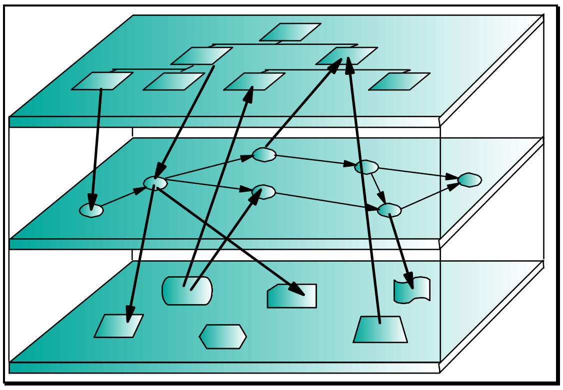


Figure 4-4: Multi-perspective enterprise model

Information modeling perspective. In GEIMM the information model is used to describe the structure of information objects (and their relationships), which may be manipulated within a business process. Entities like form, document, or folder are described. Such an information model serves as the basis for generating an information system on top of which the business processes to be modeled are to run. A schema of an information model is shown in the bottom level part of Figure 4-4 and its internal structure is discussed in section 4.3.3.

Because of this section's intention to give an introduction to the multi-perspective GroupOrga enterprise model, no further details on the three partial models need to be presented here. The succeeding sections concentrate on the partial models as briefly specified.

In order to integrate the three perspectives, all kinds of information from the three models will be interpreted simultaneously at run-time in the WfM or office system. In other words, all modeled entities are separately defined beforehand and stored into a database. After interpreting a process model, the necessary information from the other partial models will be retrieved and used for each concrete task. For example, for the execution of a task, the assigned role will be retrieved from the enterprise repository. Afterwards, a list of all agents performing this role will be generated and the task will be assigned to employee's desktops.

An investigation of recent approaches to enterprise modeling in the field of WfM and office systems (cp. chapter A in the additional documentation) has revealed that the above division into three partial models has already been initially discussed, however, these meta-models are strongly abstracted and scarcely implemented in full shaping. Admittedly, enterprise models in today's literature adjust to each other and Desai [1991] sees a "unification of underlying concepts in different office models" already. Nevertheless, he also states "that no office model [of those examined] embeds all the concepts listed in this section", so that he conceptually requests a model based on all essential aspects of organizations—as, his is still a tentative model.

Hence, by defining a comprehensive enterprise model we can capitalize on the latest research on integration of process, information and the infrastructure in one model, as well as learn from the limitations of traditional approaches to enterprise modeling. The following section presents such an enterprise model: the GEIMM.

4.3 The GroupOrga Enterprise Information Management Model

This section presents the generic *GroupOrga Enterprise Information Management Model* for WfM and office systems. According to the preceding section, this presentation follows the conceptual division of the generic model into three separated perspectives: process, infrastructure, and information. The documentation will be as comprehensive as possible and in adjustment with the few existing approaches.

Guidelines of Modeling applied in GEIMM

Becker and his team have developed a set of principles on how to orderly design generic enterprise models or portions of such models ([Becker/Rosemann/Schütte 1995] and [Rosemann 1996]). Their Guidelines of Modeling (GoM) (*Grundsätze ordnungsmäßer Modellierung [GoM]*) relate to Guidelines of Bookkeeping (GoB), yet only in structure and format, not in content. In their opinion, recent approaches to enterprise modeling lack in usefulness and applicability due to the fact that they are not concise enough. Discrepancies in terminology or inaccuracies in relations between entities have caused low quality enterprise models which are thus of little help. Hence, the GoM aim at increasing the quality of generic enterprise models by proposing design recommendations. The GoM's six recommendations are outlined below. Further reference is given in [Rosemann 1996] (pp. 85ff.).

- **Guideline of semantical and syntactical correctness.** The model has to correctly represent the reality; i.e. the correctness in content (the semantics). A model is syntactically correct when its entities and relations do not contradit each other, or disregard the object system S_O.

- **Guideline of relevance.** The model's entities and relations have to be *relevant* to the goal of the modeling process. The relevance of entities or relations in the generic model is difficult to evaluate objectively. An entity or relation is relevant in the context of WfM and office systems if the generic model cannot work without it.
- **Guideline of systematical structure.** When developing partial models of an enterprise model (as intended in the GEIMM), it is obligatory to provide for integration of the sub-models to later become one. In other words, each such decomposition following different perspectives must afterwards allow for a composition. Entities that may be used in several disaggregated models have to be used consistently.
- **Guideline of comparability.** A comparison of models becomes necessary in various situations, such as when the actual and the target structure of an organization are compared, or when two actual structures, such a subsidiary and parent company, are compared.
- **Guideline of clarity.** While formal correctness is a main motivation for the technical user, a clear and unambiguous model is most important for the end user in the departments. Thus, comprehensible structure, clearness and legibility are subsumed here.
- **Guideline of profitability.** The principles above have focused on technical aspects. Here, profitability means that the modeling process itself should remain beneficial. This principle restricts the modeling intensity, for example, by defining only *tasks* in the process model, and no further details, such as single *actions*.

The GEIMM has been developed in congruence with the GoM. For example, no equipollent terminology will be found and potential synonyms or homonyms have been eliminated or clearly distinguished. In terms of relevance, the objective has been to support WfM and office systems, which for example explains why no psychological peculiarities of an organization have been modeled. Very much attention has been paid to systematical structure and clarity. While the former principle is realized in form of technical aspects inherent to the model, the latter has found realization in the model's presentation in end user tools which are presented in chapter 5.

Scope of GEIMM

Considerable time and effort has been devoted to deciding on the scope and boundaries for the GEIMM. An unsorted list of words and terms for organizational entities corresponding to a variety of concepts relating to enterprise has been set up over the project's lifetime. These were then grouped into three distinct work areas, such that there was more context in meaning and a need to refer to terms in one area, than between different areas (e.g. process, infrastructure, organization). Within each work area, the terms were assigned priorities indicating the importance of including them in the generic model. At this point many terms were clarified or discarded and duplicates were removed.

The work areas were dealt with simultaneously, since despite their distinction there are connections between the respective entities. For each concept, a term was chosen and the definition in the model given. While the final three perspectives arose quite early (see [Ott 1995]), they evolved as new entities were added and others removed or moved to other areas. These major structuring elements of the GEIMM are reflected in the following three sections.

Many factors influenced the choice of entities in the GEIMM. The ultimate criterion was the judgment of what concepts are likely to be important to WfM and office management (guideline of relevance).

Choosing terms

In favor of end user orientation, semantical correctness was a great concern. The terms for organizational entities in the GEIMM have been chosen as far as possible to match the natural use of English by people in organizations. This is difficult, since a term used in a generic model should ideally have one precise meaning. But words are used flexibly (i.e. with varying meanings) and on occasions misunderstanding may occur. Thus, some terms may not be the natural choice for a particular reader. Sometimes, important aspects from the field of WfM are identified for which there is no obvious name (yet); in such cases unusual terms may be introduced. Ultimately there are no absolutely correct choices, but the main criteria were to conform to common usage and to avoid ambiguity.

Figure 4-5 illustrates the fundamental structure of the GroupOrga enterprise model GEIMM. It consists of three different but cooperating partial or sub-models: the *process* model (bottom left), the *infrastructure* model (top), and the *information* model (bottom right). These three sub-models are interdependent and have to be defined jointly in order to allow for a comprehensive organizational description. For the purpose of an overview, Figure 4-5 intentionally shows the GEIMM with few details only, however, other figures in the following sections will reveal more details.

The GEIMM addresses all workflow system relevant issues mentioned in section 4.1: cooperative office processes (workflows) are provided with information on organizational structures and responsibilities. Actor descriptions can be filled by names or by pursuing organizational relations (like "ManagerOf") retrieving the address from the infrastructure description. The enterprise model also comprises an access control scheme based on organizational structures and the entities like persons, roles or organizational units. The combination of these features in the enterprise model allows to build integrated workflow application environments that are adaptable to the needs of changing organizations. The connection between the three perspectives is gained over the task entity which plays a major role in the GEIMM.

As Figure 4-5 shows, the GEIMM has been defined using the extended entity-relationship (EER) modeling technique, which was first published in its basic version by Chen [1976] and then extended. It has served as the basis for many other models. Chapter D in the additional technical documentation reviews it in detail.

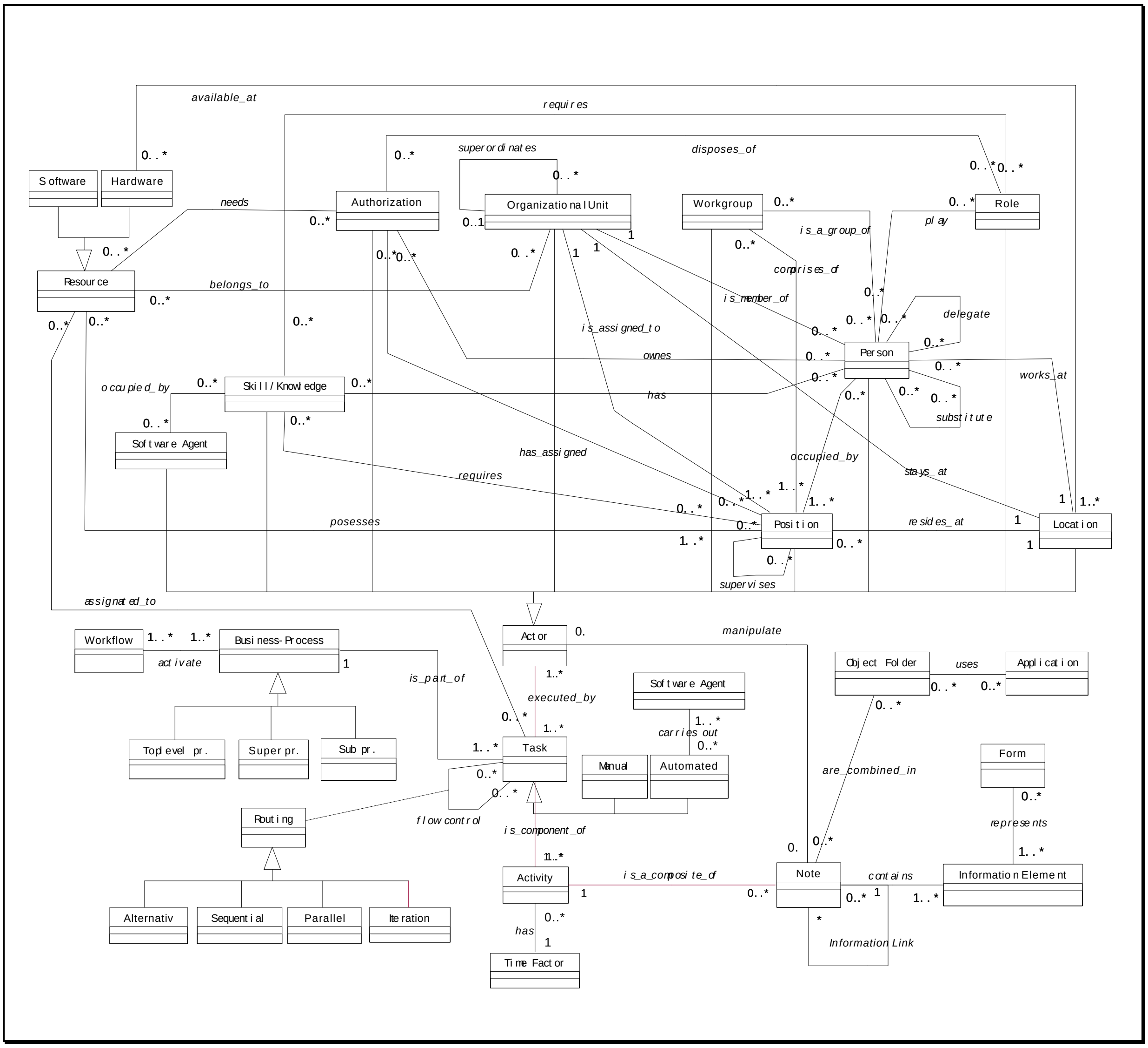


Figure 4-5: The GroupOrga Enterprise Information Management Model

In brief, the EER modeling technique proposes a fundamental abstraction mechanism which divides the description of object systems into various entities (hence the frequent use of the term *entity*) and the relationships (associations) between them. Entities may have attributes describing characteristics of the entity which are shown in ovals. The model system is restricted to those entities and relationships that information is wanted to be kept about. In the

EER modeling technique entities are shown in rectangles and relationships in diamonds, with lines connecting entities to relationships and vice versa. An additional feature provided in the EER modeling technique is cardinality constraints, showing limitations to the extent to which an entity may or must be associated with entities at the other end of a relationship. Figure 4-6 depicts the fundamental elements of the EER notation. A model can be a verbal description, a schematic picture, a physical or mathematical representation. With the EER modeling technique, this research uses a combination of verbal, schematic, and mathematical representations to develop its generic enterprise model.

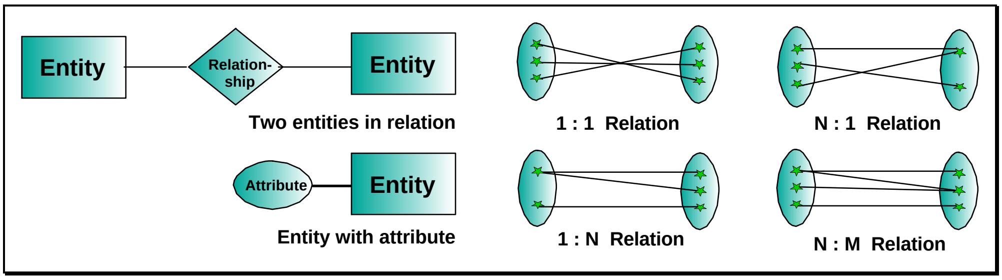


Figure 4-6: Notation of the EER modeling-technique used for the GEIMM

Documentation structure

The central content of the following sections is the definition of entities forming the GEIMM. As noted above, the structure corresponds directly to the organizational perspectives chosen. Within each section, entities have been grouped so that terms closely related appear close together. However, there is no perfect way to organize the entities to avoid references between the three sub-models—also, a suppression of these existing references is not wanted.

Each entity is introduced with a definition. Within each section those entities are presented first, which are considered to be the most basic, using these to later define other entities. Hence, the entities introduced first will be defined and described rather extensively, while explanations of other entities which are introduced later can fall back on these preliminary statements and are thus shorter.

Defined terms of the GEIMM are written italicized in the following sections, however for convenience of exposition, grammatical variations are also italicized as if they were defined themselves. In general, any defined entity will be—and has already been—presented italicized throughout this chapter. Occasionally, defined terms for entities will also be used informally. The general rule is that such terms that appear in normal type should be interpreted in their daily sense and in the light of their context.

4.3.1 Entities of the Process Model

The GEIMM process model captures the dynamic aspects of organizations namely that tasks usually effect changes on the other entities, such as *units*, *forms*, or *documents*. In the process model many *actors* are involved in the *tasks* of the organization as a whole, usually in a coordinated and/or collaborative manner, in accomplishing given activities. So, the process model shows the rules of *how* the organizational *tasks* in an enterprise are performed cooperatively. One way to model such a cooperative *task* is to decompose it in terms of the different *actors* involved in this *task* and the work that they contribute to it.

Complex organizational *processes* encompass many single process-steps or *tasks*. The process model therefore includes *tasks* and *activities*. Each *task* consists of several *activities* involving modifications on information objects, which can be found in the information model. For example, an ordering *task* would involve the actual end user who needs and hence orders an item. This *person* instantiates a *workflow* from a *process* definition. Afterwards, a clerk who prepares the order, the initiator's manager who may authorize the order, a clerk in the finance department who checks availability of funds, and a purchasing agent who carries out the order in coordination with an outside supplier, may be involved in the overall purchasing *process*. Each of these agents plays a certain *role*, has assigned *authorizations* and responsibilities, and performs well-defined *activities*.

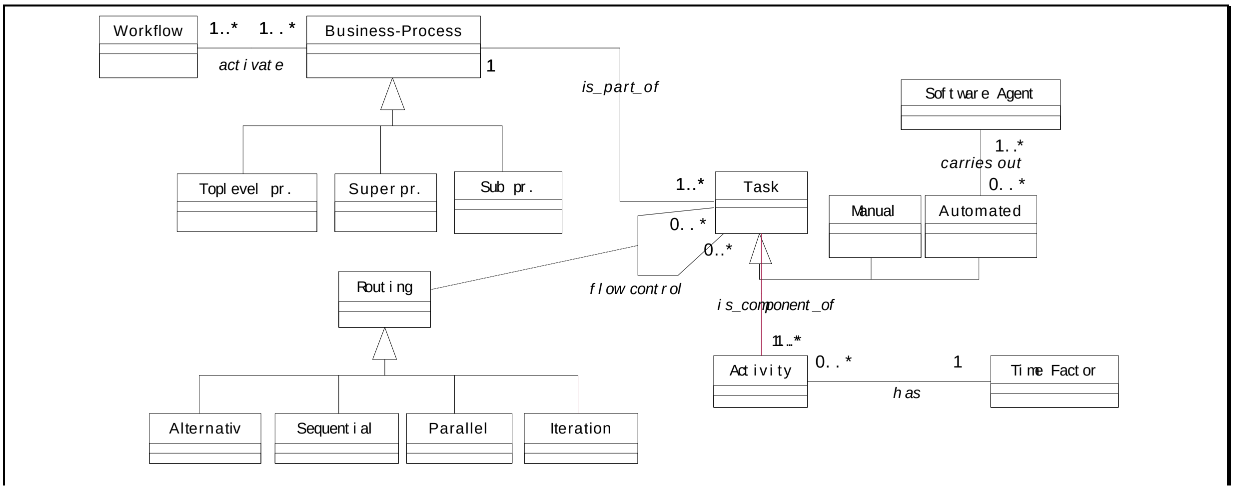


Figure 4-7: Process model in GEIMM

The perspective of organizational *processes* is one of a distributed environment through which messages flow and of actors by which information is processed. Thus, in this perspective, *task* and *workflow* are popular examples of this partial model. In order to represent the dynamic progress of *tasks*, it is necessary to introduce a class and instance concept for representing a general *process* template (such as a purchasing process) and an actual project in progress (such as the ordering *workflow* from the example above). Figure 4-7 shows the GEIMM process model.

Process Model - GEIMM Definition
A *process model* is the part of a generic enterprise model which captures the dynamic aspects of organizations. The *process model* shows the rules of *how* the organizational *tasks* in an enterprise are performed cooperatively.

4.3.1.1 Manual and Automated Task and Activity

A *task* is what *actors* actually do, it is a goal-oriented job to be performed by one or more *actors* and hence takes the position of the central and most important entity in the process part of the GroupOrga enterprise model. Each *task* is a component of a complex *business process* and consists of at least one (or more) *activities*. It uses, modifies or produces one or more *information objects*. *Activities* are the smallest unit of work in the GEIMM, which are scheduled by a WfM or office system. A list of *activities* can be considered a to-do-list for a *task* until completion. Figure 4-7 shows what relations exist for a *task* within the GEIMM to other entities.

An individual may wait, collect, check, organize, monitor, interact, distribute, identify, re-organize, report, plan, create routines, and so on: a number of minor *activities* add to a *task* and advance the *business process*. Next to these *manual tasks* that are performed by a person, a business process could also contain *automated tasks* (see Figure 4-7). In this case the *activities* are not performed by a human being, but are processed internally in the workflow system (see section 4.3.2.1.8), such as computing an invoice or copying data. In other words, these invoked applications are activated directly by the WfMS with no human agent being involved. The results of *automated tasks* present the input to further *tasks*, may they be manual or automated. *Automated tasks* do not have to be designed in detail, since their performance is given through the *application* which carries out the *task*.

The mapping of *tasks* to *actors* is done via infrastructure model entities explained in section 4.3.2. Carrying out a *task* usually requires one (or more) *resources*. Each *task* is an element of a *process* and may handle *information objects* (as input or pre-conditions) or modify and produce *information objects* (as output or effect(s)). *Tasks* are components, i.e. various *business processes* for different purposes may use the same *tasks* in different order.

Mostly, the *tasks* performed on data are non-structured. The same *task* can be performed in several ways, as long as the result is the same. Flexibility in performing *tasks* is essential for achieving office goals due to the large number of exceptions that can occur. A large number of exceptions that can arise in the office work should be accounted for at least to a minimum extent. It is necessary to determine how to proceed if a person needed for a special *task* is momentary absent/not available.

Tasks are assumed to be classes classified in the sense of having instances. An instance of a *task* class is created when the ongoing *process* is in a particular state that activates or triggers the specific *task*. Such a condition starts a *task* instance. For instance, in an organization that has a special policy for the payment of bills, the prerequisite of the *tasks* "pay invoiced amount" may be the fact that the bill is two months old and that the payment has been authorized. The required conditions and their valid combination are determined by business rules.

Task - GEIMM Definition
A *task* is a goal-oriented job to be performed. Each *task* is one logical component of a *business process*.

Manual Task - GEIMM Definition
A *manual task* is a task where the *activities* are performed by a human being.

Automated Task - GEIMM Definition
An *automated task* is a task where the *activities* are carried out automatically by computers, i.e. processed internally in the WfMS.

Activity - GEIMM Definition
An activity is a building block of a task. A task may have one or more activies. *Activities* are the smallest unit of work in the GEIMM, which is scheduled by a WfM or office system.

4.3.1.2 Routing Primitives

Process models are defined in form of directed graphs. In such graphs the succession of *tasks* is depicted in form of ordered connections, which may describe information flow in terms of *parallel routing*, *sequential routing* or *alternative routing*. In case of a division of the information flow, several *tasks* may succeed one *task*. In a *parallel routing* all succeeding *tasks* will be performed, while for the *alternative routing* it may be one or more than one succeeding *task* only, depending on which *routing control* condition proves to be true (see section 4.3.1.3). In both cases one or more threads of flow control will exist.

If more than one *task* precedes another *task* , the joining of the information flow is described. Similar to the division of the information flow, *alternative joins* and *parallel joins* are known. In case of the former, the following *task* can be executed if one of the preceding *tasks* has been completed, while in case of the latter all preceding *tasks* have to be finished before the following *task* is allowed to be started. In the GEIMM the join situation is implicitly defined through the choice of the related division of the information flow. No extra entities will be defined to denote a division or a join. In case more than one edge leaves a *task*, the *conditions* applied to the edge define whether a *parallel routing* or *alternative routing* is desired. Equally, all edges arriving at a *task* are synchronized there and the *condition* whether or not to continue with the *workflow* is deducted from the type of related division node.

Because of the modifications performed, an *information object* will be routed along different paths in the *process* which are defined by *routing control*. The decision which routing path has to be chosen due to the modifications is usually decided interactively by an agent. In case a division has been modeled, the route to follow can sometimes be determined by the workflow system through an examination of the data stored in the object. However, in both cases a *routing control* definition has to exist which will be evaluated.

Sequential Routing - GEIMM Definition
A division of the information flow where exactly one *task* succeeds another *task*. After sequential routing, only one thread of flow control exists.

Parallel Routing - GEIMM Definition
A division of the information flow where several *tasks* succeed one *task* in parallel. All succeeding *tasks* will be performed. After parallel routing, the number of threads of flow control, equals the number of succeeding *tasks*.

Alternative Routing - GEIMM Definition
A division of the information flow where one or several *task(s)* may succeed one *task* in parallel. One, more than one, or all succeeding *tasks* will be performed, depending on which *routing control* condition proves to be true. After alternative routing, one or more threads of flow control exist.

Iterations are yet another routing primitive that can be designed in GEIMM. In this case, a particular *task* is repetitively executed until a specific routing control *condition* becomes true which allows to chose an alternative edge. However, this routing primitive can be designed with the concepts already introduced and is thus no additional routing primitive, as such. An iteration may be modeled using an *alternative routing* primitive in combination with an additional *alternative* or *parallel routing* primitive.

Iteration - GEIMM Definition
An information flow where one or many particular *tasks* are repetitively executed until a specific *routing control condition* becomes true.

4.3.1.3 Routing Control Condition

A *business process* that is performed by *actors* follows rules and prescriptions of how the work has to be done. This routing control description expresses the logic of the *process* and how the *tasks* have to be performed. The rules have to be modeled with basic routing primitives to make the workflow environment route a *task* along the intended way. Each *task* can be connected to one or more other *task(s)* via a *condition*.

Conditions are applied to the edges which connect the *tasks* and these *conditions* are evaluated at run-time in the WfMS. Formally, *conditions* are logical expressions consisting of the three elements 'event', 'condition' and 'action' where the *condition* is evaluated using data and operations. Their structure is shown in Figure 4-8.

ON	event
IF	condition
THEN	action
ELSE	else-action

Figure 4-8: Formal structure of routing control *conditions*

In the *condition*, a distinction can be made between variables and constants. The value of a variable will be determined at run-time in the WfMS based on the process information. Afterwards, the result will be compared with the *condition* and the *information object* will be routed accordingly through the THEN or ELSE case routing path. In most end user oriented modeling environments, including GroupOrga and its GEIMM, the formal structure of *conditions* is hidden behind a simple graphical representation which operates according to the formula presented in Figure 4-8. At times, the event and *condition* components of such a formal structure are combined (see [Stonebraker 1992]).

However, these computable *conditions* are not considered sufficient to define workflow relevant information flow. Hence, *conditions* which encompass a more descriptive character are meant to control information flow in closer relation to the workflow's context. Such descriptive *conditions* are less formal than those described beforehand. WfMS should allow for such *conditions* which are evaluated by a human agent, which is why both concepts exist in the GEIMM. Depending on the organization and the process it might be sensible to allow the human *actors* to decide about the information flow or to only use automatic evaluation of *conditions* in other situations.

Some WfMS may define explicit pre- and post-conditions for tasks, especially those which rely heavily and state-transition diagrams. For simplicity reasons, *conditions* as defined here may be understood as post-conditions, representing the pre-condition of the following task at the same time.

Routing Control Condition - GEIMM Definition
Routing control conditions are logical expressions created from data and operations which are evaluated at run-time in the WfMS to decide which *tasks* within a *process* will be executed.

4.3.1.4 Business Process

A *business process* combines a number of *tasks* which have to be carried out according to *routing control conditions* along certain sequential and/or parallel *routing primitives* in order to achieve one of the organizational goals. *Process* specifications may decompose into other *process* specifications. Naturally, a *process* is found within a context of an organizational structure where it will be executed by one or many *actors*. It may be completely carried out within a single *organizational unit* (which due to its definition may be one complete organization, see section 4.3.2.1.5) or it may span several legally separated organizations. A *process* is triggered when defined circumstances necessitate it and each time new instances, i.e. *workflows* are created. A *process* may incorporate both, *manual tasks* and *automated tasks*. *Process* execution results in state transitions of one or several *information objects*, which in turn are invoked through the incorporated *tasks* and *activities*.

Generally, two basic types of *processes* may be distinguished according to the type of their primary *tasks*: In case of material *processes* the tasks are characterized by physical actions, while in case of informational *processes* the *tasks* are mainly of the intellectual type. Informational *processes* describe the handling of *information objects* and their exchange between participating *actors*. A *process* is chiefly addressed by the name or content of its most important or central *information object*. For example, a process dealing with the purchase of goods would most likely be called 'purchasing process'.

A *process'* central order of tasks, i.e. its main line, depicts the general structure and ordering of tasks in order to reach the intended goal. It abstracts from possible special cases, variations, and exceptions. Hence, it formulates the *ideal* succession of *tasks* and the main intended state transitions for the central *information object*. Any variants or exceptions will be related to this main line, i.e. they will be derived from the general structure when needed (e.g. at run-time).

Jablonski [1995b] separates toplevel-processes, sub-processes, and super-processes as shown in Figure 4-9.

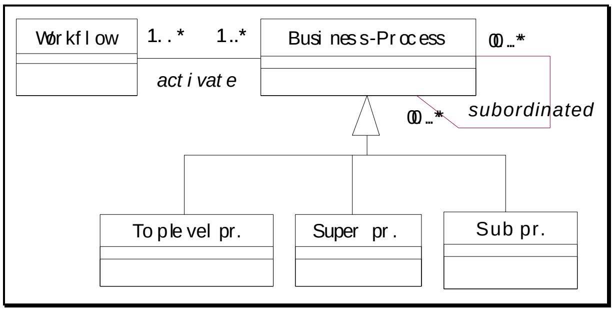


Figure 4-9: Distinction of processes

A toplevel-process has no higher *tasks* or *processes*, sub-processes are subordinated and super-processes are an overriding combination of *tasks*. This distinction is helpful for a better understanding and this aspect will be covered in GEIMM through the standard entity *process*. In other words, each *process* may be a toplevel-, sub- or super-process, depending on its position within an interlocking of *processes*. Specifications may also be accomplished through inheritance from higher level *processes*.

The number of hierarchies for *process* interlocking is not restricted in GEIMM. Such a hierarchical structure for *processes* is especially important when complex organizational procedures are to be modeled, in order to gain a comprehensible representation of the real situation. The definition of *processes* as elements within a process library allows for reuse or multiple use of the same *process* definition.

The decomposition of *processes* in sub-processes follows different criteria, such as service stage (i.e. a sub-process resembles the completion of a certain stage in the service), infrastructure characteristics (i.e. a sub-process is completed within a particular *unit*, *workgroup* or *location*) or according to IT specialties (i.e. a sub-process may involve one particular IT equipment for its full processing).

Business Process - GEIMM Definition
A *business process* is a set of partially ordered *tasks* manipulating one or more *information objects* to reach a goal. These *tasks* are carried out according to routing control *conditions* along certain *routing primitives*. When a *process* is triggered, new instances (*workflows*) are created. A *process* may incorporate both *manual* and *automated tasks*.

4.3.1.5 Activating Workflows

When a *process* describes exactly one succession of *tasks* to reach a given goal, a *workflow* is the actual automation of the *process*. Initially, i.e. after the so called *process* instantiation, a concrete *workflow* is activated, which may then be created, carried out and terminated by means of a WfMS. The *workflow* is the activation of a *process* and hence a logical copy of the objects contained in the *process model*, and it is treated as a new instance. To avoid conflicts between instances of the same *process*, a *workflow* is represented by a new configuration which is automatically derived directly from the initial configuration. It presents the corresponding *process* definition whenever a *process* is instantiated. This new configuration becomes an exclusive scope in which the new *workflow* is executed.

In other words, a *workflow* is the representation of a single enactment of a *process* including its associated data and concrete users, where data specifications and role names may have been defined but not bound in the *process* definition. Initially, a *workflow* comprises logical copies of all elements in the *process* definition, afterwards, it may evolve due to changes implied by *workflow* execution. Each such *workflow* represents its own thread of execution with its own state being controlled independently from other threads of this *process*.

After instantiation, due to the detailed *process* definition, a *workflow* may dynamically be extended by adding new *tasks*, reflecting the specifics of this particular instance of the *process*. Exceptions and modifications in the *workflow* are somewhat local, which means they do not necessarily reflect to the root *process* definition. All *workflows* are derived from the same root *process* definition by selecting subsets of its objects. However, newly created elements in the *process* definition will not be propagated to the *workflows* already initialized. Though, they may be used in subsequent instantiations.

Process instantiation is illustrated in Figure 4-10. Two *processes* P_1 and P_2 have been defined and *process* P_2 has three instances: w_{21}, w_{22}, and w_{23} which are executed separately. Workflow w_{21} is currently performing task t_1, workflow w_{22} is performing parallel routing, and w_{23} has already been finished. w_{23} has created three new or modified *information objects* (indicated by black dots), which are available only in its own configuration.

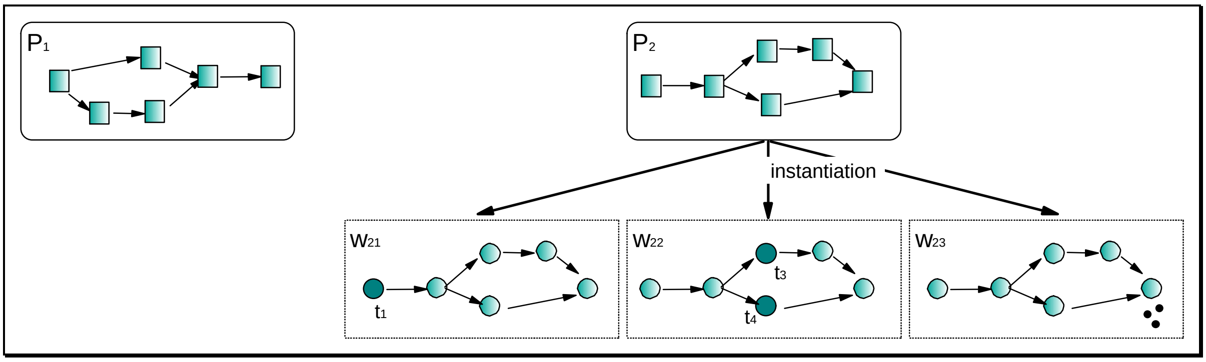


Figure 4-10: Activation of *workflows* through *process* instantiation

For example, in Figure 4-10 *process* P_2 may be a purchase definition and the three instances are concrete purchases, such as w_{21}: 'purchase hard disk', w_{22}: 'purchase software', and w_{23}: 'purchase keyboard' with the updated purchase list being one of the three modified *information objects*.

Workflow - GEIMM Definition
A *workflow* is a concrete activation of a *process* (or *process* instantiation). Each *workflow* represents its own thread of execution with its own *information objects*, and its state is controlled independently from other threads of this *process*.

4.3.1.6 Time Factor

An essential aspect for the specification of enterprise models is the *time factor*. It is used to either determine the life time of *information objects* or to specify the duration of *tasks* or timing constraints. The *time factor* may serve for cost accounting or control applications and is used for scheduling, calendar functionality, project management and performing control operations in the workflow environment. Time restrictions like planned or average duration of an *activity* form the basis for comparison between planned and actual task execution. The *time factor* must allow a certain amount of flexibility to define time constraints, which usually lack precision. A deadline requirement may be an important aspect of the *time factor* as for instance a business letter must be answered within a certain period of time. Precise observation of several possible *time factors* in *workflows* is a crucial necessity, since their violation presents a substantial cause for exception handling or ad-hoc modifications.

Time events may be absolute or relative. In case of absolute *time factors* reference is given to a fixed date or time in a calendar. Relative *time factors* are offsets from an origin, which start relative to a particular *activity*, *task*, or *workflow*. For instance, the product should have been delivered three weeks after a purchase workflow had been instantiated. Other examples for *time factors* include specification of retention periods for *information objects* and specification of periodic tasks (the salary processing is to be performed at the beginning of every month).

Several attributes which may be defined for a *task* involve the *time factor* for specification, such as:

- **Deadline.** This is a planned constraint which is compared to the actual processing and completion time. It requires a certain operation to be completed by a specific absolute or relative point in time. A deadline may be used to prioritize one workflow against another in a WfMS.
- **Start time/date.** The earliest starting point for the task or workflow in terms of periodic and non-recurring executions.
- **Duration.** The length of time that the *task* of *workflow* performance is expected to take. Adding the *task* duration on the longest path results in the maximum *workflow* duration.
- **Waiting period.** This is used to specify when an automatic reminder needs to be triggered if a *task* has not been worked on in a while. This enforces short return cycles.

Time factor - GEIMM Definition
A *time factor* is a time-based specification of *activities*, *tasks* or *workflows*. Time events may be absolute (calendar dates) or relative (relative time point).

4.3.2 Entities of the Infrastructure Model

A *workflow* will always be carried out within the framework of an organization's infrastructure and population. Such an infrastructure can be represented in terms of a set of entities and relationships, which may allow for the portrayal of any organization. Regarding the three organizational perspectives introduced in section 4.2.2, the infrastructure model to be introduced in this section is the main concern of the GroupOrga project.

Designing infrastructure is related to describing the structural aspects of an organization. It describes the different building blocks of an organization, their properties and how they are connected with each other. A *person*, *workgroup*, or *unit* represents some of the basic entities of each organization. They are correlated by means of relationships, as depicted in Figure 4-11, such as one person being the superior of another, one unit being a subunit of another, or a person belonging to a workgroup.

A number of explicit questions (i.e. directly posed by human beings) and system-intrinsic questions (i.e. occurring during execution of an office management system, for instance) can be answered through infrastructure modeling with GEIMM: 'How is the organization decomposed into units?', 'Who are the members of a particular unit of the organization?', 'What positions exist in the unit?', 'What position does a particular person occupy?', 'Who does the person report to?', etc.

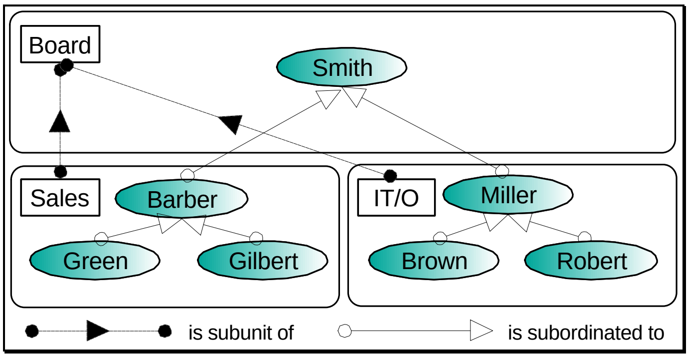


Figure 4-11: Example relationships and infrastructure entities

The *actor* is the central entity of the generic infrastructure model. It can be a *person*, a *role*, a *workgroup*, a *position*, an *organizational unit*, a *knowledge/skill* owner, an *authorization*, a *software agent*, or a *location* (in order of their appearance in succeeding sections). This enumeration shows that not only human performers may be represented in an infrastructure model of entities and relations, but also artificial ones, e.g. programs or machines. In respect to WfM, human and non-human performers make up an organization's population or its *actors*. For a *process* definition, each of these entities may be an *actor* (is_a) and will be used to design a reference from procedural entities (*tasks*) to structural entities. A *position*, for instance, is a description for an abstract performer within an organization, which comprises the performing of a number of similar and related *tasks*. Such an abstract performer is closer specified by its name, its rank within the organization's hierarchy (if applicable), its *authorizations* and requirements, etc. Each actor resides at a specific *location*. The *location* is not an attribute of an *actor* in GEIMM, but an entity which may be used for specification purposes in a *process* definition.

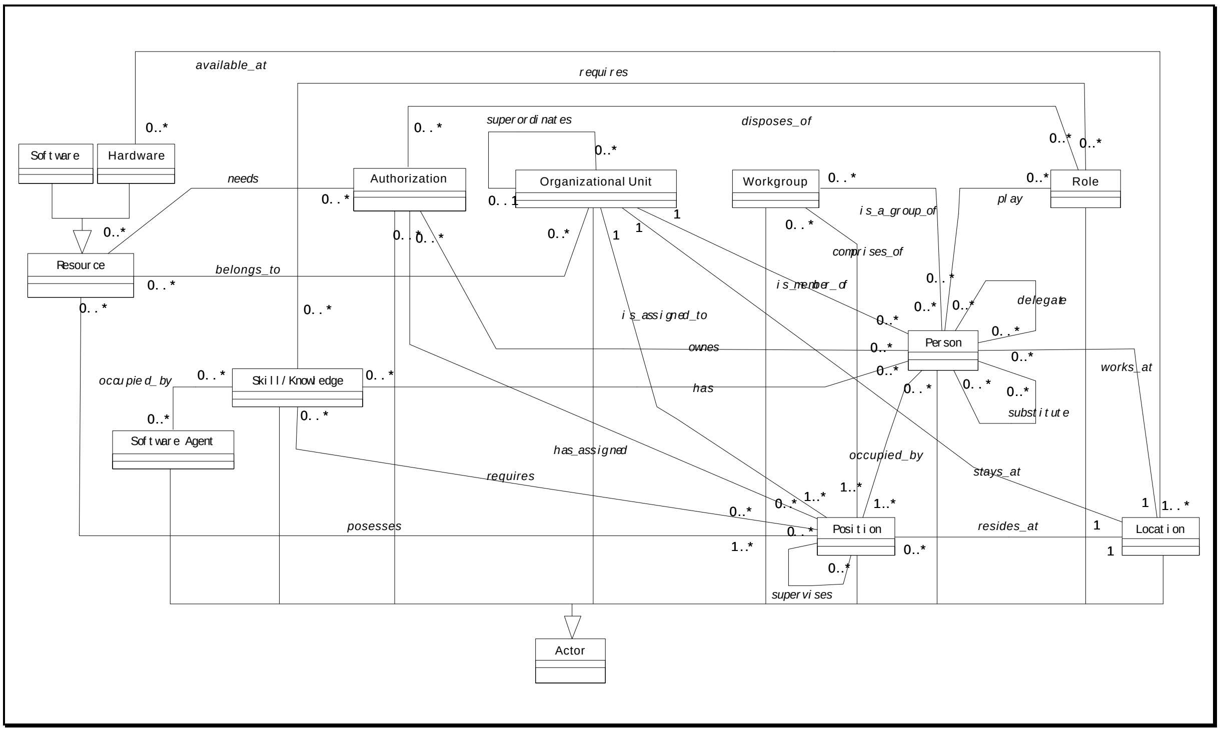


Figure 4-12: Infrastructure model in GEIMM

The representation of an organization's infrastructure in GEIMM, as depicted in Figure 4-12, allows for the creation of an organizational chart, which illustrates the structures of supervision and membership in *organizational units*. On the one hand, it possesses sufficient expressive power to design concrete organizational structures. On the other hand, it allows to depict less strict organizational principles, and model small workgroup organizations by means of *workgroup* membership or possession of *roles*. Amongst other authors, Gerstein and Shaw ([1994], p. 270) postulate that global change processes from hierarchical structures to such workgroup organizations only take place slowly. Therefore, the GroupOrga enterprise model allows for both approaches, for it being suitable to today's and tomorrow's organizations and still paying due attention to hierarchies. Although it is best suited for designing organizations with definite and distinct structures of moderate size (see section 3.1), it may also be applicable to large, hierarchically structured organizations.

A delineation as in Figure 4-12 also allows for the calculation of organizational metrics and hence for organizational analysis (see section 7.3.2).

Infrastructure Model - GEIMM Definition
An *infrastructure model* is the part of a generic enterprise model that represents the structural aspects of organizations. It describes the different building blocks of an organization, their properties, and how they are connected with each other. Human and artificial performers may be represented in an infrastructure model.

4.3.2.1 Actors Performing Tasks

An *actor* in GEIMM is an abstract entity which executes a *task* in a *process* by performing the specified *activities*. An *actor* my be a *person*, a *role*, a *workgroup*, a *position*, an *organizational unit*, an occupant of specific *knowledge/skills*, an occupant of specific *authorization*, a *software-agent*, or anybody at a specific *location*. In general, *actors* are objects which can respond to a requirement to execute a *task*. Thus, a ready to execute program code is also an *actor*. All of the aforementioned entities represent one or a number of human beings, except *software-agents*. A machine cannot take any responsibilities; it acts indirectly as defined by a *person*. The entities are defined subsequent to this introduction and may collectively be referred to as (potential) *actors*.

Actor - GEIMM Definition
An *actor* is an abstract entity who is responsible for the execution of a *task* in a *process* by performing the specified *activities*.

4.3.2.1.1 Person

Persons represent human office workers who are concerned with WfM and who utilize the WfM or office system. A *person* may be a member of the organization (i.e. internal) or the person may be an external partner represented by one particular human being. For example, in a purchasing process which spans several organizations, the supplier can also be specified as a *person* in GEIMM.

The entity *person* has intentionally not been defined as a specialization of an *organizational unit*, but as an entity parallel to it. In case of it being a subordinated entity of an *organizational unit*, a design for flexible workgroup organizations which exist without organizational structure but represent loose groupings of *persons* would be impossible to devise. Moreover, with such a structure of a *person* as an entity of its own, *positions* may be modeled, as well.

The *person* entity relates a specific *task* to one specific human being in an organization. Although this approach is straightforward, it is problematic insofar as it is inadequately adaptable to changes of the organizational structure. Whenever a job-shifting takes place each process definition has to be checked whether it is still valid. In other words, while *persons* may be identified directly within the *process* definition, they should rather be identified by reference. *Persons* play a number of different *roles*, belong to one or various *units* and to none or various *workgroups*, have particular *knowledge/skills* and *authorizations* and reside at one *location*. Any of these references to *persons* can then be filled and a *person* may change the *task's* status and other *task* attributes.

Person - GEIMM Definition
A *person* is a human office worker who is responsible for the execution of a *task* in a *process* by performing the specified *activities*. *Persons* are referenced in other entities, such as *role*, *unit*, *workgroup*, *knowledge/skill*, *position*, *authorization*, and *location*.

4.3.2.1.2 Role

Before giving an account of the details of GEIMM's *role*, a note on *roles* and *role* concepts in today's WfMS is considered necessary as an aside: Section 3.2.1 has mentioned that most WfMS and their associated enterprise models do not separate the aspect of process and infrastructure, and if they do so, they (only) rely on the concept of simple *roles*. One of the most recent investigations in WfMS by Becker, Vogler, and Österle [1998] reveals that the main meta-model entities of important standard WfMS only have the *role* entity as an infrastructure entity. This is considered insufficient even though this concept is used in many approaches, it is defined differently in each of them. Hence, the *role* concepts in theses various systems are not adapted to each other. Still, the implementation of *roles* is taken as a grant for the flexible design of organizational structure in WfMS. It has been outlined and thus will not be analyzed again that other entities (such as those presented in this chapter) are necessary, as well.

The GroupOrga project also represents the *role* as an entity in its enterprise model. Undoubtedly it may *again* differ slightly from *roles* in some existing WfM approaches, however its integration into a much larger enterprise model and hence its interaction with all other entities is considered a step forward. Within GEIMM the *role* concept has been adapted as close as possible to existing definitions and enlarged by allowing combinations with other entities. Concluding, in the GroupOrga project the *role* concept is not considered unnecessary and superfluous as such. On the contrary, it is a significant structural entity in enterprise models and will be weighted accordingly in GEIMM.

Two different views of a *role* are examined in the following: the traditional understanding of *role* as opposed to how it is used in today's WfM and office systems.

In their 1958 publication Gross, Mason and McEachern define: "*A role is a set of expectations*, or in terms of our definition of expectations, it is *a set of evaluative standards applied to an incumbent of a particular position*." (p. 60). Their definition depends on the definition of *position* which will be given in a succeeding section. Similarly, Slater [1965] defines "a role as a more or less coherent and unified system of items of interpersonal behavior." (p. 610). In [Roos/Starke 1981] one main connotation of *role* is that of status, reputation, standing, or prestige.

Such traditional definitions of *role* stem from large groups of role theorists and are still a valid and current topic in organizational research. This research stream's goal is to account for the variability of the behavior of role players of the same position. It focuses on social behavior and it understands a role as a person's pattern or type of social behavior which seems appropriate in terms of demands and expectations of a group of other people. Hence, a *role* has ingredients of cultural, of personal, and of situational determination.

This very brief insight into the role definition in social-science has been offered to differentiate it from *role* in the context of current WfM discussion. It will become clear that both associations with the term differ to a high degree, and might be a reason for the misunderstanding of *role* in WfMS. This is a fact which must be seen as yet another argument for not only relying on the *role* concept in WfMS.

Role is an organizational entity for representing certain organizational circumstances and for the implementation of organizational relations. In GEIMM the concept is applied as an abstract grouping of agents with similar competence and qualifications, i.e. in order to aggregate a number of functions which can be carried out by all persons playing this role. Thus, any of the members of such a grouping can perform a certain *task* requiring *knowledge/skill* or other attributes owned by all *role* players. At runtime, in a WfMS the current *role* definition is evaluated and all persons eligible to play the *role* are addressed. As long as there is one member who fits the *role* specification, it has not to be checked in case an agent is added or deleted within an organizational structure. As the assignment of agents to *roles* is independent of particular *tasks*, the lifetime of the *role* and its assignment depends on the organization in which it is defined. However, this assignment will not change very often.

In GEIMM a clear semantical distinction between *position* and *role* entities is made as follows. For *roles* the most important question is which function is carried out by a particular *person* within a *process*, whereas *positions* describe which (abstract) *persons* within the organizational structure have to fulfill specific functions. To clarify this, *roles* stress the procedural aspect of functions to be carried out by whom, while *positions* aim at the structural aspect. Due to this use of *role* in GEIMM it is possible to specify roles which are relative to a concrete *workflow*. Typical expressions for such *roles* are, for instance, "Vacation application underwriter" or "Vacation application clerk". If one would try to assign this expression to specific *positions* it would become obvious that *persons* who play this *role* may vary due to the initiator, i.e. the "Vacation applicant" of this workflow. Hence, *positions* and *roles* may be used interchangeably, but strictly depending on the point of view of *task* completion.

Strictly speaking, *roles* are kept distinct from agents for two reasons. First, the reconfiguration of an organization may involve changing *roles* and how they are grouped. *Persons* need not to be affected, though *role* requirements may change, and thus different *persons* may be needed. While this first argument holds true for other actor-type entities, as well, the second reason follows from the first and is mainly true for the *role* concept. *Persons* play *roles* in unique ways and their qualities may change over time. The assignment of a *person* to a *role* requires that the abilities of the former are appropriate for the requirements of the latter. If the two objects are not kept distinct, there is no basis for such a matching.

Generally a *role* is specified by its qualitative (*knowledge/skill*) and quantitative (capacity) requirements. Moreover, since a role will be referenced in various processes and sub-processes, *authorizations*, rights and duties are also connected with it. A concrete *person* can play none, one or more than one *role* in general, for example by being 'supervisor for purchasing processes' and 'insurance underwriter' at the same time and without conflict. A *person* then fulfills qualitative and quantitative requirements and takes on the associated *authorizations*, rights and duties.

A specific role, namely the 'process responsible' should be pointed out to. Each defined *process* should have a *person* responsible, who designs it, has the overall view, and controls its correct enactment. This *role* player is also responsible for any larger exceptions and modifications to the *process* definition which go beyond the current *actor's authorizations*.

Role - GEIMM Definition
A *role* is an abstract grouping of agents with similar competencies, attributes and qualifications (*knowledge/skill*) in order to aggregate a number of functions which can be carried out by all *persons* playing this role.

4.3.2.1.3 Workgroup

Currently personal *workgroups* are mentioned as the central entity of infrastructure reorganization in organizations, especially under the aspect of flexibility. In literature various terms and names may be found for this approach, namely *partly autonomous units*, *decentralized, product oriented units*, *long-term teams*, *task-oriented teams*, *flexible control groups*, or *clusters*. For some time their utilization was named in one breath with CA-technologies, such as Computer Aided Manufacturing (CAM). Task assignment to *workgroups* is attracting growing interest in the field of WfM and computer supported work. *Workgroups* as a powerful means for independent *task* assignments have been outlined in chapter 2. Comparably to the *role* discussion in social-science research, group aspects are also topic of much literature. Again, GEIMM will only address the technical aspects and leave social questions untouched. More importantly this section delineates the technical *workgroup* entity in GEIMM.

A *workgroup* comprises one or many *persons* who in turn play *roles*, process *tasks* etc. Members of *workgroups* can be characterized by specific forms of cooperation and by their mutual responsibilities. In contrast to *roles*, *workgroups* are real (tangible) groups. That is, they are social systems, complete with interdependence among members, and differentiated member *roles* in both contexts: the social and the system context. Moreover, members are dependent upon one another, and they play specialized *roles* within the *workgroup*. Accordingly, *role* comprises a number of alike *persons* (of course only concerning abilities and knowledge, not personality), whereas *workgroup* covers an intentionally different scope: the combination of a number of very different *persons* to reach a common goal. A *workgroup* is composed of members from varying organizational levels, different *organizational units*, and varying *locations*. Thus, the existing hierarchical structure will not be affected or even broken up. Members may also come from outside the legal organizational boundaries and only interact with the organization through their *workgroup* membership.

GEIMM allows for both, the definition of open and closed *workgroups*. Open *workgroups* are flexibly expandable and new members can be defined easily. Such *workgroup* structures are mainly used for *tasks* which are limited in time. Typical *workgroups* are event management groups or project groups. Closed *workgroups* in contrast consist of a given set of *persons* and are not intended to be changed at short notice. Such closed *workgroups* may perform *tasks* fully self-responsible and their members organize the division of work independently. In GroupFlow the according *tasks* are called team-tasks. Examples are bodies of experts or committees.

Workgroup - GEIMM Definition
A *workgroup* is a grouping of agents with different *authorizations*, attributes and qualifications (*knowledge/skill*) in order to aggregate these qualifications in a smaller organizational entity for reaching a specific goal.

4.3.2.1.4 Position

Positions are used to describe a not specified organizational member or not specified employees as positions are usually occupied by *persons*. *Positions* are considered another central element for designing organizational hierarchies. Abstracting from daily business, like employee fluctuation, and the development of human resources, the concept of *position* is still essential to plan, develop, and maintain organizational structures. *Positions* describe a set of functions which can be solved by a *person* and are thus held by one or many members of the organization. There may be times when a position is vacant, i.e. a *position* can exist independently of any *person* to occupy it. In case a new employee is trained on a particular job, two *persons* may theoretically hold the same *position* and *positions* may also be shared in GEIMM. Although many organizations cover similar *positions* (e.g. five sales representatives) each *position* is unique within the organization and represents one or many abstract organization members.

Positions stand in a hierarchical context to each other, i.e. one *position* may have supervisory status over other *positions*, which in turn may supervise other *positions*. This principle of subdividing is aggregated in the concept of *organizational units*, which is to be introduced in the next section. According to their depth in the organization's hierarchy, *positions* are grouped into main units, units, groups, and subgroups which are not to be confused with the *workgroup* entity which spans hierarchical structures. Sometimes *positions* are hard to be assigned to one or another *organizational unit* which is the reason for so called staff positions which in turn are aggregated to *staff units* (see section 4.3.2.1.5).

A *position* description comprises three different parts: Firstly, qualifications (*knowledge/skill*) are required by a member of the organization in order to be able to occupy a *position* in which certain responsibilities have to be taken. Secondly, *authorizations* may be assigned to a *position* such as to sign special contracts or to approve a travel allowance. Thirdly, the *persons* who occupy a *position* have to take part in operations of the organization, and perform *roles* and complete *tasks* for which *resources* may be assigned to the *position*.

Last it should be noted that the concept of *position* is inherent to the German organization theory and less well known in other societies' organizational structures. Due to GEIMM's openness to design alternatives and its ability to flexibly design various different kinds of infrastructure situations for WfM and office systems, its possession of *position* as an entity of its own is no design restriction. Infrastructure models can (but must not) be designed using the entity *position*. Workgroup organizations may as easily be designed as strict, hierarchical structures containing *positions*. In German practice *position* descriptions (Stellenbeschreibungen) are set up, which are written specifications of how *tasks*, responsibilities, *authorizations*, duties and rights interact in the *position*. Additionally, this description can take in information regarding resource requirements, such as room space, furniture etc. Besides using *position* as an active entity for workflow enactment, GEIMM and its practical implementation, the GroupOrga enterprise knowledge base, allow for such documentation purposes, as well.

Position - GEIMM Definition
A *position* describes an abstract organizational members or employee because positions are usually occupied by *persons*. A *position* describes a set of functions which can be solved by one *person*. Thus, a position is held by a member of the organization.

4.3.2.1.5 Organizational Unit and Organizational Staff Unit

Looking at today's organizations, the presence of hierarchy must be noted. There seems to be an universal function to such hierarchies, as they are efficient and robust against confusion or disorder. And although formal organizational charts are obviously hierarchical, it can be argued that informal organization would also be found to be hierarchically structured. In a chart of informal interaction, the clusters of interaction may identify a rather well-defined structure, as well. The theme that all organizations have an aspect of hierarchical structuring can be found with empirical support. This is the reason for GroupOrga, despite its innovative approach towards flexible workgroup concepts, to implement the hierarchical element of *position* and *unit* in its generic enterprise model GEIMM, as well. Smaller organizations may refrain from using these entities when designing their concrete organizational model, however the idea of hierarchical structures still pertains to most organizations and is thus necessary as an entity.

The first concept associated with the infrastructure model of an organization is that of *units* and organizational charts. As Mintzberg [1979] already points out, they describe an organization as a system of formal authority representing "... an accurate picture of the division of labor, showing at a glance (1) what positions exist in the organization, (2) how these are grouped into units, and (3) how formal authority flows among them ..." (p. 37). Even though organizational charts fail to reveal information about how business is really done in the organization and veil real power dependencies, nearly all organizations use charts at least for administrating their employees. Therefore, a comprehensive enterprise model such as GEIMM has to cover the underlying concept of *organizational units* and *positions*.

Organizational units are an aggregation of none, one or more *positions* to broader entities in the hierarchy, such as accounting, sales or marketing. This distinction leads to the common hierarchical structure of an enterprise as depicted in Figure 4-13. A grouping of *positions* into *units* can follow object specifications or functional aspects. While *resources* can be assigned to single *positions*, as elaborated in the proceeding section, they can also belong to *organizational units* and are thus available to every member of the respective *unit*. Each *unit's location* can for example help to determine the degree of distribution of a particular workflow application.

Except the highest level, each *organizational unit* is under the sub-ordination of a higher *unit* and has none, one or more subordinated *units* in turn.

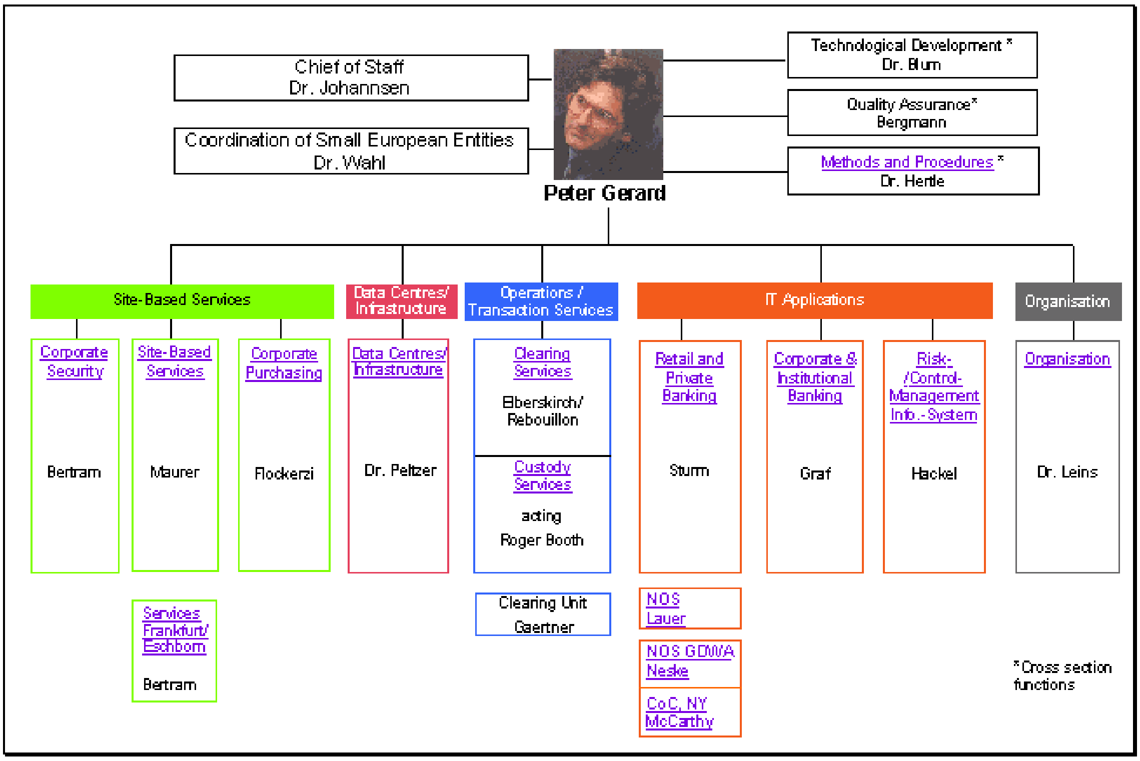


Figure 4-13: Organizational chart of the Deutsche Bank IT/O branch as of 1997

The term *organizational unit* is purposely defined with no constraints on its size or place within an organization. In addition, no discriminated terms are defined for *units* of different size, e.g. division vs. department. This is because no consistent use of these terms exist, as to when a *unit* is a department and when it is a division. An *organizational unit* can be a very small and simple grouping of *persons* or *positions*, as well as a large and complex structure. Following the same arguments no such entities as 'enterprise' or 'organization' are defined—they can be considered high-level *organizational units*, as well, perhaps corresponding with the highest *unit* in a specific instance of the enterprise model.

In addition to the above design of *positions* belonging to *units*, GEIMM also allows for a design without *positions*, i.e. for an assignment of *persons* directly to *units* without specifying *positions* as an intermediate stage. This openness in the generic model may be considered a weakness, since it might open up the way to inconsistent design. However, GEIMM is expected to be applicable to small, midsize and large organizations on various levels of infrastructures with a strong workgroup orientation. In case of innovative workgroup organizations which still live with some remnants of organizational hierarchy, the constraint to have to use *positions*, presents an unnecessary restriction.

If an *organizational unit* has only one *position* or *person* as a member, in GEIMM this would still be referred to as 'a *unit* with one member', as opposed to the *person* or the *position* being interpreted an *organizational unit* itself. In more formal words, *person* and *position* are not in a 'is_a' relation to *unit*.

Organizational staff *units* are defined as *units* which indirectly contribute to the organization's goal as a supporting element to regular *units*. In GEIMM a staff *unit* and a regular *unit* are distinguished by an attribute of *unit*. Figure 4-13 shows a graphical representation of staff *units* similar to that applied in GroupOrga tools.

Organizational unit - GEIMM Definition
An *organizational unit* is an aggregation of none, one or more *positions* or *persons* to broader entities in the organizational hierarchy. This distinction leads to the common hierarchical diagramming of an enterprise in organizational charts. A grouping of *persons* or *positions* into *units* can follow object specification or functional aspects.

4.3.2.1.6 Knowledge/Skill

A *person's* knowledge or skill has great importance in today's human resource systems and evaluation processes. This entity has not yet widely found its way into WfM and office management systems. *Knowledge/skill* is characterized in terms of expertise in fields, such as programming language, hardware technology, foreign languages, etc. For example, to start a *process* for the purchase of a personal computer, it should be known who has control and operative responsibilities. In the same case, to ask for advise on which personal computer to buy, it should be known which *person* has or which *persons* have the technical expertise on personal computers. *Knowledge/skill* is characterized in terms of the knowledge each one has acquired in the various fields. The occupant of *knowledge/skill* is the most suitable partner from whom to obtain information or support concerning the given field during a *business process*.

In today's information-based organization, the knowledge will be primarily at the bottom, i.e. in the minds of the specialists who do specific work and direct themselves (see [Drucker 1988]). Therefore, today it is difficult to find out who has which knowledge and who can be helpful in certain *business processes*. This process of determining process participants, of course, should most effectively happen disregarding the organizational hierarchies. Those *persons* are asked to support in processes who have a particular knowledge, regardless of their current *position* in the hierarchy.

There is no framework for knowledge in organizations. Knowledge creation is subjective—it comes or emerges from individuals who depend on intuition as much as information. New knowledge always begins with the individual. This is a reason for *knowledge/skill* being an entity in GEIMM which has a relation to *person*. Knowledge starts at the individual level, and is often not easily expressible or accessible in an organization. The knowledge may become formalized, i.e. explicitly specified, by defining it in an organizational handbook stating who has which knowledge and how it can be utilized. This conversion of informal to formal representation and specification involves the participation of the individuals. An enterprise model may thus serve as a formal collection of discrete pieces of information combining them to create larger pieces of information. Later, individuals may either internalize such knowledge or they may refer to information from the enterprise model to call other individuals with the appropriate *knowledge/skill* into action.

By means of an organizational database, an individual's knowledge is transformed into organizational knowledge valuable to the organization as a whole. The central aim of such entity *knowledge/skill* is to make information about personal knowledge available to other individuals managing *business processes* who in turn may involve these individuals in their processes, since they have a particular knowledge (cp. [Ackermann 1994]).

Although, so far the *knowledge/skill* entity has been defined only for *persons*, it may also be applied for *software agents*. In this case the term *knowledge/skill* admittedly applies more to human beings, however if it is interpreted in the context of 'capability', it also denotes capabilities and functionality of a *software agent*.

Knowledge/Skill - GEIMM Definition
A *knowledge/skill* is a capability or expertise that spans a defined knowledge field. It denotes a specific range of skill, knowledge, or ability and is closely associated with detailed knowledge in terms of technology and individuals. Having *knowledge/skill* indicates the ability to be the *actor* for a specific *task* in a *process*.

4.3.2.1.7 Authorization

An *authorization* defines what the occupant of an *authorization* is entitled to do within a *business process* or more generally within the organization. This concept of permission describes two things, the right to use and access certain resources for *task* completion and the right to perform certain actions within a *business process*, such as "is allowed to sign contracts of any kind". Hence in GEIMM it describes what *tasks* a *person* may execute without explicit permission.

Authorization is only considered to be valid within the WfM or office management system it has been designed for and the process enactment within this system. In other words, it does not necessarily cover authorization within (heterogeneous) company applications. In a combination of infrastructure and *process models*, access to *tasks* in WfMS can be given by means of *authorization*. By defining that a *person* has access to a *task* as being responsible or only as someone who has to be informed, it can be defined how access is given. For instance, while an office clerk may have full editing rights to all purchasing information while the corresponding process is still in its initiation phase, the same clerk may have only read access to it after the purchase has been approved, so that no changes can be made afterwards.

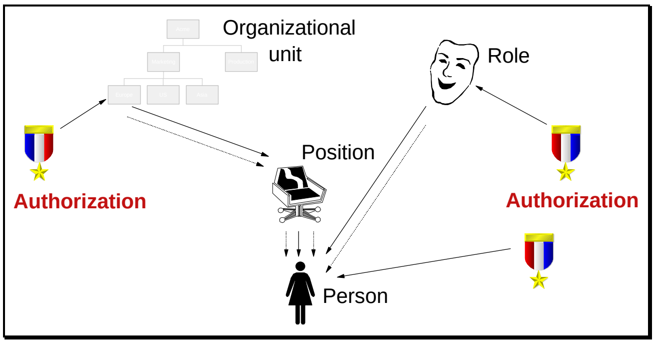


Figure 4-14: Inheritance of authorization

Other entities are in relation to *authorization*, such as *positions* or *roles* possessing such a warrant, so that all occupants of a particular *position* or *role* can perform the same *tasks*. As depicted in Figure 4-14, a person's *authorization* depends on this person's placement within the organizational structure and not necessarily on the individual.

Moreover, for a *task* within a *process* definition it may be defined which *authorizations* must exist to be allowed to execute the task (as opposed to specifying particular *persons*, *roles* or *units* for *task* enactment). For example, a *process* definition could formulate "everyone can perform this task, who is allowed to sign contracts of low value". Similar to other actor specification, such as *position* and *role*, *authorizations* may be given to one or many *persons* in order to design exclusive or cooperative responsibilities for *processes*.

Authorization - GEIMM Definition
Authorization describes the competence, responsibility or power to perform *tasks* within a business *process*. It can be assigned to *positions*, *persons*, and *roles*.

4.3.2.1.8 Software Agent

IT has long been considered a mere *persons'* aid for performing organizational *processes*. The concrete enactment of *tasks* within organization is, according to traditional theory, a distinct area for human beings only. But *software agents* can perform certain *tasks* within organizational IT independently, which spans the boundaries of *task* assignment to single *persons*. An example is the checking of items in or out of stock, for which sale is permanently announced and electronically updated by several sales representatives. The machine may self-directedly initiate a purchase of goods when certain stock runs below predefined values. Each such *task* performance must be defined as a logical transaction, i.e. as a functionally capsulated *task* which can be started, performed and completed by a *software agent*. Its start is either initiated online through a *person* or via time or content conditions evaluated within the WfMS and started as batch technology.

This application program may be interacting with, and on behalf of a *person*, but GEIMM designs and several software platforms (such as Lotus Notes groupware) allow selected *tasks* to be performed by *software agents* without human intervention. One must keep in mind, that only a *person* can assume responsibility for a *task*; hence, a *software agent* on a machine acts only indirectly in the name of a *person*. Reusable *software agents* are invoked by workflow applications and are thereby included in *process* enactment. The *software agent* and the workflow application exchange input and output data during these *automated tasks* (see section 4.3.1.1).

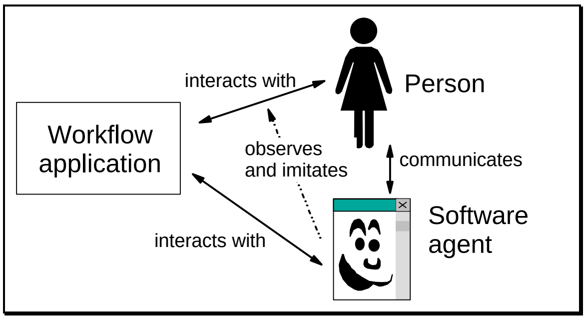


Figure 4-15: Software agents interact with human user

Hence, it is meaningful to view IT and *persons* equally as possible performers of *tasks* in business *processes*. Theoretically both may occupy *positions* in organizations, however this has not been reflected in GEIMM. Varying capabilities of different IT components (software or machines) may equally be taken into consideration as different skills. Due to the parallelism of *person* and *software agent* as equal actors in GEIMM, *software agents* are not designed as an interface or a layer between the user and the workflow application. They are considered a personal assistant that cooperates with the user on the *task* and fully takes on certain *activities*. The user is always able to bypass the *software agent*, as Figure 4-15 depicts.

The GEIMM approach for designing automated *task* completion in enterprise models is to define them as *actors*. For each *actor* of the type *software agent* a program must be available that can execute that *task*. Thus every *software agent*, like any other *actor*, has a queue of *tasks* that it has to execute.

Software agent - GEIMM Definition
A *software agent* is an application program that executes *automated tasks* in WfM and office systems. Each *task* performance must be defined as a logical transaction, i.e. as a functionally capsulated *task* which can be started, performed and fully completed by a *software agent*. It is either initiated online through a *person* or via time or content conditions evaluated within the WfMS and started as batch technology.

4.3.2.1.9 Location

As the last entity specifying a general *actor*, the notion of *location* has been introduced in GEIMM. It describes the locality of potential *task* performers, i.e. *persons*. A *location* is a physical, geographic point. Various *actors* are based at a specific *location*. *Locations* may be either a country, region, city, or floor whereas a plant or department are subsumed as an *organizational unit* since a plant or department may theoretically be dispersed over various *locations*.

The locality of participating *actors* can serve as an information about how far the *process* is distributed. Additionally, this information may be used to narrow down the number of potential actors for a *task* according to their physical presence. For example, *any actor* residing at a specific location might take on the *task* of taking the stock, due to the fact that an organization's storehouse resides at the same locality and since no special *knowledge/skill* is required for this simple *task*. So, in GEIMM yet another aggregation of actors takes place due to their locality in an entity of its own and not as an attribute of a *person*, *unit* and so on.

Location - GEIMM Definition
A *location* is a physical, geographic point (a country, region, city, or floor) where various *actors* (*person*, *position*, and *unit*) are based.

4.3.2.2 Delegate and Substitute

In an organization *persons* may be absent due to various reasons, such as illness, vacation, or projects at external locations. These *persons* occupy *positions*, play *roles*, belong to *groups* etc., i.e. they are involved in *business processes* over various mechanisms. In case any of them is absent due to the above reasons, *processes* may be interrupted, slowed down or come to a halt, because a specific *person* is absent. Especially in case of automated WfM and office systems this is even more true, since a growing computerized *task* queue is not as obvious as a paper-based one which piles up on an employee's desktop. Hence, specifically in such distributed, electronic workflow environments comprehensive substitution regulations are inevitable, specifying who has to perform another *actor's tasks* during this *person's* absence.

Different types of substitution are distinguished in organizational theory:

- **Placeholder.** A placeholder is a *person* who is not entitled to assume all the *tasks* of a particular *person*, but only a subset of the *tasks* and only under precisely defined situations. For example, a placeholder has to decide whether the *person* being substituted must be informed about important *tasks*, whether a third party must be informed, or whether the *task* completion can wait until return of the original actor.
- **Replacement.** A replacement is a *person* who temporarily replaces another *person* and acts on the behalf of that *person* and not in the name of that *person*. Hence, this *person* replaces the other *person*, and fully assumes this *person's position*, *roles*, *workgroup participation* and responsibilities.
- **Substitute.** A *substitute* is a *person* who performs all of the *tasks* of another *person* as that *person*, i.e. using the name of this person. The *substitute* should match the description of the *position*, or *role* of the substituted *person*. In other words, it cannot be expected that a *substitute* can fulfill the *tasks* as well as the original *actor*, since there is no equivalent experience available, but only to the degree specified in the *task* description. The *knowledge/skill* of the substitute is

hence defined through the *position* occupied or *role* played (i.e. ad rem) and not through the *person* substituted (i.e. not ad personam).

The abstract entity *substitute* in GEIMM is intended to represent the third of the above interpretations of substitution in organizations. However, the three cases are more of a social and legal matter, than one of technology support. Based on a GEIMM enterprise model, a WfMS will route a *task* to a particular *person* substituting another regardless of the implicit rights given to this person. Such technology support can ensure that appropriate personnel is notified in case of absence of other employees, but it cannot (and it is not intended to) watch over implicit organizational regulations.

In addition to substitution, delegation is another similar, yet distinct case in organizational theory. In GEIMM a *delegate* (a *person* normally of a lower hierarchical level) is defined in case the original actor does not want to act (due to several reasons), whereas in case of a *substitute* the original actor cannot act. In some literature the aspect of intentional substitution is stressed by using the term *emphasized* delegation (see [Theuvsen 1996], p. 60). While this differentiation appears to be subtle, it is important for generic enterprise models: A substitution rule may always be established, in order to come into action whenever a particular actor is unexpectedly not available, while a delegation is intentionally set up for specific purposes and time periods, although the person to be substituted is available. Moreover delegation fosters the idea of decentralization and spread in organizational structures.

It can be concluded that delegation is a strategic means to relieve high-level positions, to increase the *authorizations* of lower-level positions, and to reach higher flexibility in organizational decisions. In contrast substitution is a technical means to keep up smoothless workflow execution and *task* enactment in case of exceptions.

The substitution problem has been examined extensively in progress of the GroupOrga project. While this section only addresses the results in terms of its implementation in the GEIMM, [Jaschik/Lang 1997] considers particulars of the concept of substitution.

Substitute - GEIMM Definition
A *substitute* is a representative of another person who can exercise full authorization in the person's absence and has equal authorization in emergencies, i.e. during the person's (unexpected) absence from the office. The *substitute* must perform the *tasks* of the absent *person* and should match the description of the absent person's *position*, or *role* (i.e. substitution 'ad rem' and not 'ad personam').

Delegate - GEIMM Definition
A *delegate* is a (normally lower hierarchical level) *person* who is officially elected or appointed to represent another *person* when the other person does not want to perform certain tasks. Delegations are intentionally set up for specific purposes and time periods, even though the other *person* is available. A *delegate* may or may not be entitled to act in the name of the original actor.

4.3.2.3 Resource

In addition to specifying entities such as *actors* in organizational hierarchies and flexible entities, such as *workgroups* and *roles*, an enterprise model should also represent software and hardware *resources*. These *resources* can be assigned to *tasks* during *process* design in order to support actors in completing the *task*.

There are two types of *resources*:

- **Software.** A software resource can be an editor or a compiler, in other words, any type of software program. A software resource may need certain hardware resources to be executed.
- **Hardware.** A hardware resource is any type of hardware, in particular, the computers and peripherals that are located at a given *location*.

Software *resources*, such as application programs or databases, can be referenced in the enterprise model by specifying their functionality and methods for their invocation. Furthermore, links to particular pieces of information such as database records, documents or web pages can be managed in a concrete realization of an enterprise model in order to ease their reuse. This may help *process* designers choose the appropriate tool and help administrators get an overview of the inventory. WfM systems use this information to automatically launch an application or display a web page that is associated to a *task* in the *process* model. Another advantage of administering these software objects in the enterprise model as entities of their own is that the actual *location* on the enterprise network, the users' workstations or the Internet is independent of the *process* design. Objects can be moved and tools can be exchanged without having to modify the *process* definition to which they are assigned.

Information about non-physical *resources*, such as network connection time or application programs, can be stored in the organization's infrastructure model for using these *resources* according to their specification when required. This facilitates the maintenance of these objects because they only need to be updated in one place—the entry in the organization database—and the new information is automatically available to all employees.

Another aspect is that accessing *information objects* through the organization database allows to assign access rights to software *resources*. When an *actor* wants to access the *resource* via the directory, access can be granted, (the *software agent* is launched) or denied, depending on the *authorization* necessary.

Some of the concepts described above are valid for hardware *resources* as well. Managing devices, such as computers and presentation equipment, facilitates their use for *process* designers and gives administrators and users an overview of the equipment they have. In contrast to software (i.e. intangible *resources*), the control of the organization database over hardware *resources* is less restrictive. Launching a program or running an agent are actions that might be performed directly by directory services or application programs, whereas gaining access to a computer or a meeting room is harder to do. However, reservation schedules for these devices can be stored and managed in the model equally.

Resources are either assigned to a *position* or any higher *organizational unit*. Thus the *resource* is at the disposal of every member of that *organizational unit* or any occupant of a particular *position*.

Resource - GEIMM Definition
A *resource* is any work material that can be used by *actors* for help or to support the completion of *tasks*. It is available to *units* and *positions* and can be drawn on from members of these entities when needed. There are two types of *resources*: software tools and hardware or machines. Access restrictions to a *resource* may be specified by requiring *authorization* for its use.

4.3.3 Entities of the Information Model

"No organization of two or more people can function without information. Indeed, in any organization, the character of information flows is one of the most critical variables determining the speed and accuracy with which decisions get made—and thus the qualification of execution" ([Charan 1991], p. 112). The *information model* examines the *information objects* which are handled within an organization. It delineates the structure of containers for data and knowledge created by and processed within *business processes*. These objects are considered in terms of their general structure, rather than in terms of their concrete form. In other words, the schematic description of objects takes place through definition of general object types and their relations, in the same way it has occurred in the two preceding generic partial models. Both, objects, as well as relations may be further specified through attributes. From a workflow and office perspective, the data and knowledge perspective includes such entities as *forms*, *information objects* or *object folders*. With these entities, messages may be created which appear in organizations in various types. In their concrete forming, these messages can be characterized as being structured (e.g. records or forms) or unstructured (e.g. letters or reports). They may be created and/or used during completion of various *tasks* in different *applications* available to the *actors*, such as database systems, spreadsheets, or word processors.

It has been pointed out in chapter 2 that today's WfMS are aimed at flow control and management of *information objects*, but the actual data processing itself is often disregarded. Data manipulation within the *business processes* is expected to take place in external application programs and not within the WfM or office system and input as well as output data for those application programs are expected to be provided through the WfMS. However, experience shows that this integration appears to be difficult due to outdated software and inadequate interfaces. Hence, a logical requirement for WfMS is to integrate elementary data processing and manipulation functionality directly into the workflow or office system.

Due to the above reason representation of *information objects* in enterprise models for WfM and office systems is still under-represented. GEIMM tries to tackle this problem with the *information model*, which is to be delineated in this section. Due to the project's foundation on groupware and its concrete technical realization on top of the groupware platform Lotus Notes, representation of *information objects* will be closely related to the *information model* inherent to Notes.

However, this option was not chosen because of technical restrictions caused by the use of Lotus Notes, but for good reason: in the GroupOrga project not yet another *information model* has been developed. However, in literature cited, the concerns expressed about the IT architectures produced with traditional approaches describe what an *information model* should not be: e.g. complex, not understandable, specifications not subject to validation, and perhaps at the core of the issue not integrated with the rest of the infrastructure and process model. The real issue, then, is to construct a framework where organizational integration is the central paradigm, and not IT exclusively.

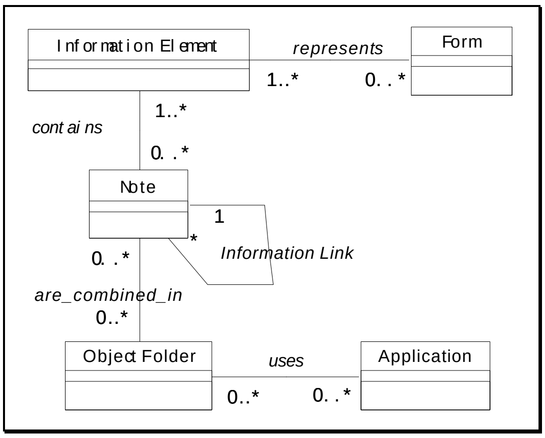


Figure 4-16: Information model

The Notes *information model* provides such a simple and understandable approach for data storage which can easily be integrated with numerous other WfM and office systems, if necessary. In other words, compliance with the Notes *information model* is not considered a restriction, on the contrary it is considered a sensible decision for an existing, powerful model of data storage and manipulation. But data is in many respects the least important dimension of information. It is of more importance to share information about experiences, critique, and opinions—soft information that cannot be captured in traditional databases and spreadsheets. Notes offers adequate means for this.

Secondly, the Lotus Notes groupware platform already provides means for both data processing *and* flow control, so that the above requirement of integrating both aspects in WfMS can be met without extra effort.

This section will present the entities and relations of the information model as illustrated in Figure 4-16. While specific terminology already exists for information management within Notes (e.g. note, field, form, view, section, etc.), generic terminology will be defined and used to express GEIMM's general practicability in the following specification.

Information model - GEIMM Definition
An *information model* describes the *information objects* which are necessary for an organization's *processes*. It presents containers for data and knowledge created by and processed within the *business processes*. From a workflow and office point of view, this data and knowledge perspective includes entities such as *forms*, *information objects*, *information links*, *information elements* and *object folders*.

4.3.3.1 Information Elements and Forms

For regular office work some kinds of *forms* are necessary to structure information storage. In the widest sense, such forms are records, receipts, card indices, blanks, lists, etc., i.e. mostly administrative forms. *Forms* are somewhat comparable to a technical drawing. Similar to how this drawing sketches the contour of a product, a *form* describes how *business process* information is captured and structured. The *form* is a basic entity of the information model in order to gain structure in information storage and to define how *information elements* are to be combined. Together with the following entities, it, for example, serves as an information container within the *processes* defined in the *process model*.

Forms are provided for communication between the *actor* and the WfM or office system. In other words, these applications use *forms* in order to realize an interactive step within the process model. They provide the adequate *form* to the *actor* at the correct time, so that the *actor* navigates through it in a predefined order and fulfills the required *task*.

Form - GEIMM Definition
A *form* combines several *information elements* and specifies their formal structure, and their relations. Components of a *form* may be other (subordinated) *forms* or *information elements*.

A *form* itself may consist of *forms* and of *information elements*. These *information elements*, in turn, are those entities which (in a given context) cannot meaningfully be divided into smaller elements. For example, a *task* may receive an input which consists of a *form* which is made up from two *forms* belonging together. These two subordinated *forms* may stem from two preceding *tasks* and have to be combined into one.

Information element - GEIMM Definition
An *information element* is an entity which (in a given context) cannot meaningfully be divided into smaller elements. *Information elements* carry single information values which are used in *processes* in combination with other *information elements*.

4.3.3.2 Information Object

The entity *information object* in GEIMM is the representation of work to be processed and hence denotes all office elements with their particular features. *Information objects* are data entities that can be distinctly identified and that may contain attributes and data fields. They are the important information carriers in organizations, since they are those entities which are 'visible' to *actors*. Relating to the Notes data structure (see section 2.2.1 and Figure 2-11), an *information object* consists of a *form* which is dynamically combined with relevant information content represented through the *form's information elements*.

For clarification purposes: The respective terminology in Lotus Notes is 'Form', 'Field', 'Note' and 'Document', however, for the purpose of generality this terminology has intentionally been avoided for GEIMM.

The *information object's* format is given through a *form* which is the structural base for each *information object*. Its particular format and exchange formats are not discussed here. Several approaches for the mapping of different kinds of standard formats onto *form* design exist (see e.g. [CCITT 1985], [ECMA 1985], [Horak/Hoffmann 1986], [Kronert 1988]). Although the *information objects* can contain different kinds of media and can be of different structures, in principle, two parts which are common to all *information objects* can be found: the index and the content itself. The index, as an analysis of various office application has shown, may consist of three parts: the administrative description, the referential description, and the content description. Table 4-1 shows the intention of these three elements of an index.

Administrative description	identification, author(s), creation date, last changing date, type of *information object*, access rights, etc.
Referential description	references to various other documents, e.g. a report is related to other reports from previous years
Content description	summary of the *information object's* content using controlled vocabulary, e.g. in form of keywords

Table 4-1: An *information object's* index

Each *information object* is used, modified or produced in the context of at least one *task* and can be used and modified in the context of another *task* within a *process*. *Information objects* serve as containers for data and have several attributes such as creation date, modification date, length or owner which are mainly used in their administrative description. Because an *information object* is based on a *form*, each has at least one *information element* and one attribute: an *information object* that does not contain any data or information is of no use and vice versa, and an *information object* that does contain data has at least a creation date.

An *information object* is in relation to none, one or several other *information objects*. During the performance of a *task*, actors will be involved through receiving and sending *information objects*. *Information objects* are to be presented to an actor as abstract objects that are meaningful in the context of the *task* to be performed, and the actor does not deal with low-level system details, such as file-names and directories. For each *information object* it may be specified which *actor* is entitled to open and/or modify its content (e.g. via a *role* list, a list of *person's* names etc.). Thus *information objects* are related to *actors* and *tasks* of the two other partial models.

Information objects store inactive information. They serve the same purposes as files in a traditional office. However, they are more 'intelligent' than files and file cabinets. There may be rules applied to *information objects* which are used to trigger events when certain conditions become true. The content of *information objects* may be categorized into pure informational data or workflow relevant data. Section 4.3.1.3 introduced *routing control conditions* which are based on workflow relevant data.

Such workflow relevant data, which are stored in *information elements* of an *information object* are used by a WfMS to determine how to forward an *information object* within a *process* due to previously defined *routing control conditions*. For example, for a decision within a credit card application, it may be crucial whether the equity capital is more or less than a certain amount. This decision can be made by the WfMS as a computerized activity, through evaluating *information elements* of *information objects*. This content of *information objects* has to be marked off from pure flow-control data which will have been specified at design time to define process flow. Such control data will not be manipulated during run-time.

For the content of an *information object*, the types of data used in conventional or classical database systems such as character, string or numeric data in a traditional field structure are necessary but no longer sufficient in the modern office environment. Other types of data like unstructured data contained in messages, letters, texts, annotations, graphics and oral communications are currently discussed. Also soft or natural information or multimedia has to be supported, such as image, video or speech objects.

Information object - GEIMM Definition
An *information object* is an abstract data object that is manipulated during *task* performance through *actors*. An *information object* may have *information elements* (structured through *forms*), attributes, and links to or relationships with other *information objects*.

4.3.3.3 Information Link

In general, *information objects* needed to perform office *tasks* are distributed among several office workers in the same or different *units* or *locations* of the organization, and can also be located externally to the office environment. An *information link* is an association or dependency between two such *information objects*. There are several data elements in each office which are related by several connections. With the linking of *information objects*, the combination of information from several objects is possible without actually copying. Thus, an integration of different information types such as textual, graphical or tabular information into one *information object* can be reached. It is possible to distribute information in an organization by linking the respective *information objects* via hyperlinks or hotlinks, without actually distributing every single piece of data to the user. Links have source *information objects* and destination *information objects*.

For cases where an *actor* needs information that has not already been associated with the *task*, *information objects* can be referred to by traversing *information links* to find additional relevant information. Although a process designer should have attached as much relevant information as can be anticipated with the *task*, so as to minimize the need for *actors* to search for additional information, the context of a *task* should be dynamically extended by adding references to other *information objects* by means of *information links* during run-time.

Information links may also be used to represent any kind of relationship that an *application* needs to track, e.g. for representing the structure of a complex information purpose, or for recording dependent *information objects* that are affected and may need revision when a given *information object* is updated.

In GEIMM, a link can be established from one *information object* to another *information object*. Several *information objects* may be linked with each other by establishing a number of such binary *information links*. It is important to mention that an *information link* connection is not only intended to work in local, but also in distributed environments.

Information link - GEIMM Definition
An *information link* is a binary relation between a source *information object* and a destination *information object*.

4.3.3.4 Object Folder

The aforementioned *information links* are available to describe relations between *information objects*. A set of such linked *information objects* may represent an entity by itself within an information model. In other words, relations between *information objects* in the context of WfM are important to indicate a closed unity of *information objects* in terms of *object folders*.

Object folders may describe a *process'* context in contrast to something which may be called a task's context and which should ideally by covered by one particular *information object* assigned to the task. The *process'* context represents the set of information relevant to the performance of the *process*. The information is managed by and available to those involved in performing the *process*. For instance, if an actor finds an *information object* to be useful to others involved in the *process*, the *information object* can be linked to the same context, i.e. into the same *object folder*. Besides defining interdependence via *information links* it can also be defined by the use of identical keywords or categories in *information objects*.

In analogy to conventional offices and their circulation folders, GEIMM represents all *process* relevant information in an electronic *object folder*. *Object folders* exist at run-time and contain data for a particular *workflow* and may be archived after it has been completed. On the contrary, a *process* definition as such (see section 4.3.1), specifies which *information objects* will generally be used during execution.

Object folder - GEIMM Definition
An *object folder* is a collection of all *information* objects relevant to a particular *workflow*. As a set of linked *information objects*, it represents an entity in GEIMM. It can be created in two ways: via *information links* or by using identical keywords or categories in *information objects*.

4.3.3.5 Application

When an *actor* 'reads' an *information object*, some sort of editor for that particular information container is to be brought up that allows the *actor* to view the content of the *information object* and possibly edit it. In the GEIMM information model, such an editor is called an *application* in order to divert from the notion that all *tasks* within *business processes* involve editing of text. In other words, information processing during *task* execution takes place in *applications*. Generally speaking, an *application* supports the coverage, processing, storage, and distribution of relevant information. It is characterized by its functionality and by its data used to operate meaningfully.

In concrete terms, an *application* might be a word processor, spreadsheet, drawing tool, legacy system (human resource or bookkeeping etc.), or image painting tool, depending on the content of the *information object*. *Actors* should not have to be concerned which *application* to invoke or 'load' an *information object* into; the WfM or office system should determine the *application*, and automatically invoke it according to the *information object's* content. An *information object's* attribute allows for a specification of the related *application*. That is, for each possible value of this attribute an *application* should be available which is capable of displaying the information in the appropriate format.

In case Lotus Notes serves as the workflow engine's base (section 4.3.3), many *tasks* which normally have to be executed outside the WfMS may be performed with Notes' on-board functionality. Some WfMS (including Notes) provide some sort of a programming language for direct implementation of *applications*. Equally, *applications* take the form of elementary program functions which use only a small functionality of a larger program.

However, in general an *application* is considered to be started and run outside the WfMS or to be loosely integrated into it. Therefore, it might take on and exchange relevant data with the WfM or office system.

As Figure 4-16 illustrates, a relation exists between an *application* and an *information object*. One job of a WfMS is to allocate *information objects* properly to *applications* according to the previously mentioned attributes. Hence, a WfMS based on GEIMM is not restricted to special data formats, since every type can be processes if the corresponding *application* has been integrated.

Application - GEIMM Definition
An *application* is a general software program, invoked by a WfMS, used to create, view, and possibly edit *information objects* during *task* execution in *workflows*. *Information objects* have a relation to the *applications* that edited them.

4.3.4 Elementary References and Sources for GEIMM

The GroupOrga Enterprise Information Management Model was invented at the Department of Business Computing and developed from scratch during the GroupOrga project's lifetime; however its development was inspired and influenced by shortcomings and drawbacks, as well as ideas and advantages of other preliminary projects and efforts, too numerous to mention. Generic enterprise models for WfM have only been under development for a short time now and some of these influencing projects are listed below, together with their main references. For a complete overview and for a collection of graphical representations of other models refer to chapter A in the additional documentation. The references listed are subdivided into the more recent approaches and—to start with—those which have served as the basis for the younger models listed later:

- The model of an organization described in [Ang/Conrath 1993] and [Ang 1996] is part of a more general office model covering dynamic aspects (office procedures). Ang distinguishes active and passive office objects. The paper describes a concept not an implementation.
- The *electronic organization manual* described in [Chrapary/Rosenow-Schreiner/ Waldhör 1991] is a knowledge base in structures, procedures, co-workers, products or services of an organization. The knowledge base is structured into four

layers: taxonomy, organization, tasks and procedures. It existed as an implementation created during the WISDOM project ([Lutze/Kohl 1991]). Its development has not been pursued.

- [Faidt et al. 1990] contains a description of an organization knowledge base modeling the specific organization of a research institute. The knowledge base contains representations of persons, areas, work fields, rooms, technical equipment and relations between these objects. Work areas correspond to organizational units.
- The Office Model One (OM-1) described in [Ishii/Kubota 1989] and [Ishii/Ohkubo 1991] is a knowledge base containing representations of office procedures, organization structure and resources (documents, files etc.). The model was implemented as a prototype and can be considered the archetype for many such approaches.
- Kreifelts and his team ([Kreifelts/Hinrichs/Woetzel 1993], [Hennessy/ Kreifelts/Ehrlich 1993]) are concerned with addressing in the office procedure system DOMINO. Their main issue is an organizational addressing scheme, which is necessary because their system relies on e-mail for coordination among office workers.
- Karbe's work ([Karbe/Ramsperger/Weiss 1990], [Karbe 1994]) describes the office procedure system ProMInanD which is based on *electronic circulation folders*. An electronic form of an organizational handbook is split into the organizational structure and the migration specification.

All of these (sub-)models are concerned with office procedures requiring information on organizational structures. The organizational structures modeled differ, but all models allow the user to represent hierarchical static organizations. As indicated in the summaries above, most models present only partial models and not comprehensive enterprise models. Recent approaches have tried to fill this gap:

- Galler [1995] presents elements of meta-models of workflow management. The paper positions such meta-models as a means for explaining how organizations operate and not as a technical specification. This is the reason for several entities being described but not implemented as a comprehensive framework of relations and cardinalities.
- A "Metamodel Workflow" is proposed in [Derungs/Vogler/Österle 1995] (see also [Österle 1993]). The title indicates the report's intention: explanation and specification of entities and relations in a process model. While the report mentions aspects of an information model, the infrastructure aspect is ignored.

- The "Enterprise Project" (see [Fraser 1994], [Stader 1996]) brought about an "Enterprise Ontology" which is explained in [Uschold et al. 1996]. It is a collection of terms and definitions relevant to business enterprises. It is not a data model.
- A reduction of maintenance effort for organizational handbooks is the main goal of the ODB/OIS (Organization Database/Organization Information System) project team ([Heilmann/Simon 1989]). This project has developed a data model. Newer concepts aim at the integration of structure and process ([Heilmann 1994]).
- Baligh tries to measure effectiveness and efficiency of various forms of organization structure with the "Organizational Consultant" ([Baligh/Burton/Obel 1990] and [Baligh/Burton/Obel 1994]). The "Organizational Consultant" does not specify the entities of an enterprise model.
- The "DESIGN 6" project was performed at the same time by the same team ([Baligh/Burton/Obel 1990]). In contrast to the Organizational Consultant, DESIGN 6 does not mainly support the description and analysis of existing organizational structures, but follows a *design-first* approach, meaning that a desired structure is proposed and then adapted to a real organization. Again, no specification of entities of an enterprise model is found.
- Jablonski and Bußler ([Bußler 1992], [Jablonski/Bußler 1996]) have made several contributions to the topic of enterprise and workflow modeling. Among other topics in this field, they have touched the aspect of integrating organization design in terms of structure and process in [Bußler/Jablonski 1994].
- WorkParty is the Siemens Nixdorf WfMS that was introduced in 1992. The comprehensive infrastructure model ORM (Organization Resource Management) developed by Rupietta and his team ([Rupietta 1990], [Rupietta 1992], [Rupietta 1994], [Rupietta 1997]), is the most developed and furthest implemented infrastructure model to date.
- The Workflow Management Coalition's "Terminology & Glossary" ([WfMC 1996a]) contains technical definitions for terms used in the WfMC specifications and discussions. The definitions help with the consistent use of workflow terminology; however, no data model is defined.
- The goal of the TOronto Virtual Enterprise project (TOVE) at the University of Toronto is to create a generic, reusable data model. Its ontology puts forth a number of conceptualizations for modeling organizations: agents, roles, positions, goals, communication, authority, and commitment ([Fox 1992], [Fox 1993], [Fox/Barbuceanu/Gruninger 1996], [Fox/Grüninger 1997]).

4.4 An Enterprise Knowledge Base as an Electronic Organization Handbook

Transforming enterprise knowledge such as that about organizational structures into a form that can be executed by IT is difficult, because knowledge in the world is expressed in a different form than that required by the machine. This *representational mismatch* has partly been overcome by abstracting characteristics of organizational reality into the generic enterprise model GEIMM. The aim of GEIMM is to adequately model the object system. Additionally, such an enterprise model should serve as a bridge to an implementation. While the primary concern of this chapter so far has been to describe the enterprise model which facilitates the conceptualization of the semantic content of enterprises, from now on implementation concerns will be considered.

Consequently, as a second step to overcome the representational mismatch, the knowledge about organizational structures needs to be implemented in any kind of computer-based form of storage. To conclude this chapter on GEIMM, the general concept of an *enterprise knowledge base* is presented.

4.4.1 Structure of an Enterprise Knowledge Base

Since the organizational knowledge needed in WfM and office systems has been identified as threefold (processes, organizational structure, information objects), it has been felt necessary to implement a hybrid database structure which includes a process repository, an organizational database and an application database containing information objects. This architecture is illustrated in Figure 4-17.

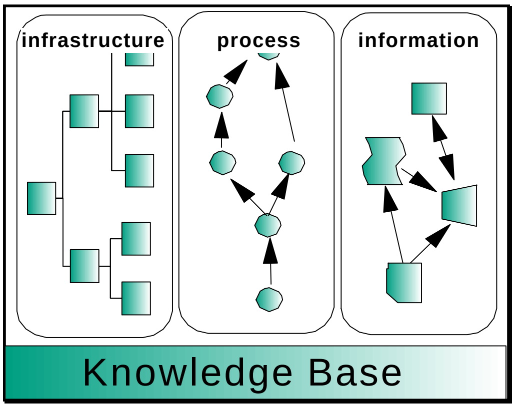


Figure 4-17: Hybrid enterprise knowledge base

The focus of GroupOrga is on organizational structure, so that the following will concentrate on an organization database. Here a brief and rather general concept of organizational databases is presented; the following chapter 5 will focus on the concrete implementation of an organization database in the GroupOrga project.

4.4.2 The Electronic Organization Handbook

In order to solve its purpose, the structuring of an enterprise has to be documented in written form. This function is met by organizational handbooks, which are in general a collection of all important organizational definitions. Besides an enterprise wide organization handbook, which is generally administered in the *organization department*, practically each unit and position needs an extract of it. Auditing and human resources departments may need larger extracts than other departments, however, every department should receive the documentation necessary for its work.

Organizational reality is in constant flux and changes in positions, group membership and role occupation are topics of daily work. This cannot and should not be prevented. Even full-time *organizers* will have difficulty to follow up on all the changes, get them in printed form and distribute the copies to the units and positions in question timely and precisely. Hence, traditional organizational handbooks are hopelessly outdated when the administration is done manually. However, it is not only the aspect of manual updating, but also a question of collecting all relevant information from distributed members at a central point.

In contrast to human organizers, IT recognizes every consequence of change, no matter how small the correction and how large the information base may be. Thus, it appears sensible to administer an organization handbook in electronic form.

With an electronic organization handbook, updates in time can be supported. Moreover, with a distributed architecture for the electronic handbook continuous contribution of changes through everyone in the organization is possible.

The workload for these changes is reduced with an electronic organization handbook compared to traditional handbooks, since adaptations and following consistency checks are performed automatically. For example, structural changes automatically reflect into the process enactment and vice versa. Changes in resource allocation or modifications in information objects or forms have direct effect to processes related with these entities. Logical inconsistencies, which may for example turn up since tasks are assigned to positions which are not entitled to carry out these tasks are detected by means of the organization database.

An organization database contains all relevant data concerning the organizational structure which was usually stored in traditional handbooks. The organization database is based on a entity-relationship model as outlined in section 4.3.2 and allows any kind of request on the infrastructure model, as well as its evaluation. These requests can be both manual and automatic through any kind of organization information, WfM or office system.

Such an information system based on an organization database primarily supports the maintenance of the organization database and allows for simple and complex queries concerning the entities and their relations. In addition, analysis of the data in an organization database assists the identification of organizational inadequacies and provides for periodical comparisons of organizational structuring. Based on an organization database, structural design planning in terms of simulation and 'What-If' analysis may be conducted.

The organization database meets the common requirements mentioned above and integrates a generic infrastructure model. In particular in GroupOrga this enterprise model is part of the GEIMM.

4.5 Summary of the proposed Model of Enterprise Design

Traditional approaches to enterprise modeling have established the importance of a 'model of business' through the creation of enterprise models only covering some enterprise aspects. Problems with the existing models stem from their limited constructs (e.g. only activity, unit and person) and scope (e.g. hierarchical forms).

The emergence of new organizational arrangements necessitates role and group level specification and broader linkage between enterprise entities. In response, in GEIMM both, purpose and schema associated with enterprise models have been reconfigured. The goal of an enterprise model is to interconnect IT with the reality of organization. The broadening of the enterprise model from one that merely specifies information systems, to one that depicts organizational functions in every respect constitutes a fundamental step. This includes operationalizing the process, the infrastructure *and* the information model, i.e. the generic enterprise model has now three dimensions, instead of only one.

Chapter 5
Concepts and Architecture for the Modeling of Infrastructure Information

The preceding chapters have outlined an intensified need for new concepts and IT frameworks for organization design. This results in novel business requirements in organizational and technological support. Furthermore, chapters 1 to 4 revealed that basic procedures and concepts of organization design have been under examination for decades. But the degree of satisfaction with the existing approaches varies immensely and is judged quite differently by management.

Literature on organization design shows that the use of IT for organization design is necessary and important. By contrast, the vast spectrum of technologies and methods is often perceived as unclear and complex. In relation to the high costs of these new technologies, the applications are considered unsatisfactory. At present, the existing technologies have functional characteristics that give isolated help in certain areas of organization design. In addition, innovative, networked systems can fundamentally rearrange the procedures for organization design and fulfill the requirements. Information and communication systems for data storage, distribution and management play an important role. In order to meet the requirements, integrated solutions based on these technologies are vital. The support of organization design *as a whole* is the aim of these integrated systems. This notion was followed during the development of the GroupOrga system.

GroupOrga is a conceptual framework and synergetic combination of prototype applications for distributed organization design. Both have the aim of supporting the heterogeneous and wide-ranging necessities of method-based design of organizational structures in their great variety. Furthermore, it provides a technological base for process-oriented organization design. It is built to cover and solve the spectrum of problems in the focal field. These include the issues and topics of chapter 3, such as integration of workflow IT and organization design, flexibility concerns, distribution aspects, tool support in the design, and process orientation.

Sections 5.1 through 5.4 explain and discuss GroupOrga's central concepts and methods as a general solution for the topics tackled in the problem definition in chapter 3. Section 5.1 introduces the basic concepts of GroupOrga, providing a basis for the subsequent sections. Section 5.2 consolidates the distribution aspect in the framework, and section 5.3 looks at the core technology of GroupOrga: the Enterprise Knowledge Base (EKB) for WfMS. In section 5.4, the overall layered architecture of the GroupOrga prototype system is explained. Section 5.5 sketches the individual components of the prototype system and explains its selected functionality. Section 5.5 gives no explicit description of the components. Refer to the tool's manuals and technical documentation.

Some minor procedures and concepts of GroupOrga result partly from the description of the information model in the preceding chapter and from the presentation of the system architecture. Some of these aspects are summarized, while other aspects are discussed in depth.

5.1 Basic Concepts of the GroupOrga Framework

To begin with, the basic GroupOrga concepts are presented against the background of findings from section 3.2. They are displayed as an overall thought for modern information management of organization design. Thus, this first part outlines the *vision* of GroupOrga. It names the most important characteristics and it explains their possible advantages. Elements of this vision are an integration of workflow IT and organization design, the idea of evolutionary organizational subsystems, and the use of distributed technology. The implementation of computer-based tool support for the realization of a participative, learning organization is an additional point.

Furthermore, the GroupOrga tools for organization design provide a graphical modeling language which supports the modeling of a concrete organization. Such an organization model is based on the comprehensive data model GEIMM. GEIMM was identified as a requirement in section 3.2, and was introduced in chapter 4. Therefore, the data model is only summarized in the following section.

Figure 5-1 illustrates how the separate ideas of GroupOrga are integrated by using groupware technology. It shows how they are implemented in a computer-based modeling environment to support simultaneous, participative modeling. This procedure is realized in a continuous, consistent and distributed, cooperative modeling of structural reality.

The modeling of information related to structures is carried out with existing and newly implemented tools. The benefit caused by integration of specifically developed tools is that designers of the various areas can work with tools that are familiar to them and which were developed for their individual tasks. Hence, the participants of the planning procedure (bottom-level employees are usually included) are not overburdened by the mega-functionality of a comprehensive expert tool.

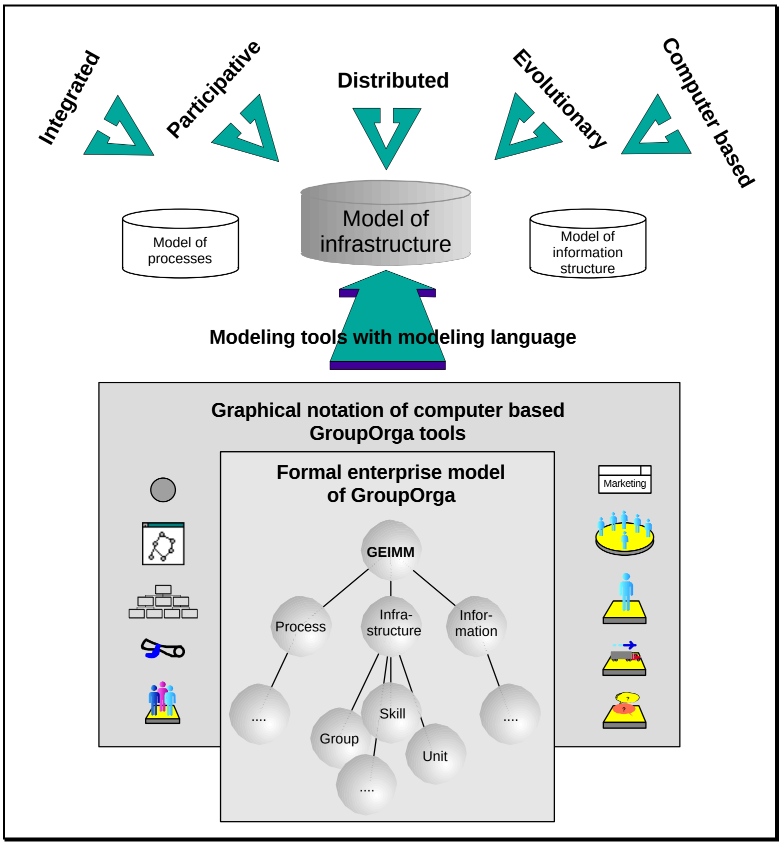


Figure 5-1: Elements of the GroupOrga vision

The realization of the structural design tasks is initiated constantly by forming (and disbanding) workgroups consisting of employees as well as specialists for organization design. Different planning stages may actively be taken over by the different members of the group, and the subtasks are worked on continuously and parallel to each other. In practice, several group members handle the complex, diverse design tasks in addition to their daily activities. They can work simultaneously and can also be physically distributed. Thus, the problem solving process of this distributed, participative design can be characterized as a group process using groupware concepts. Groupware methods aim at providing common access to resources.

Organization Design Continuum

Throughout this study, different *forms* of organizational structure are mentioned, such as traditionally hierarchical or network structures. It is also emphasized that a transition from traditional forms to innovative structures must be seen as an evolutionary process, and not taking place at once. Figure 5-2 illustrates the potential of the GroupOrga framework to accompany this development.

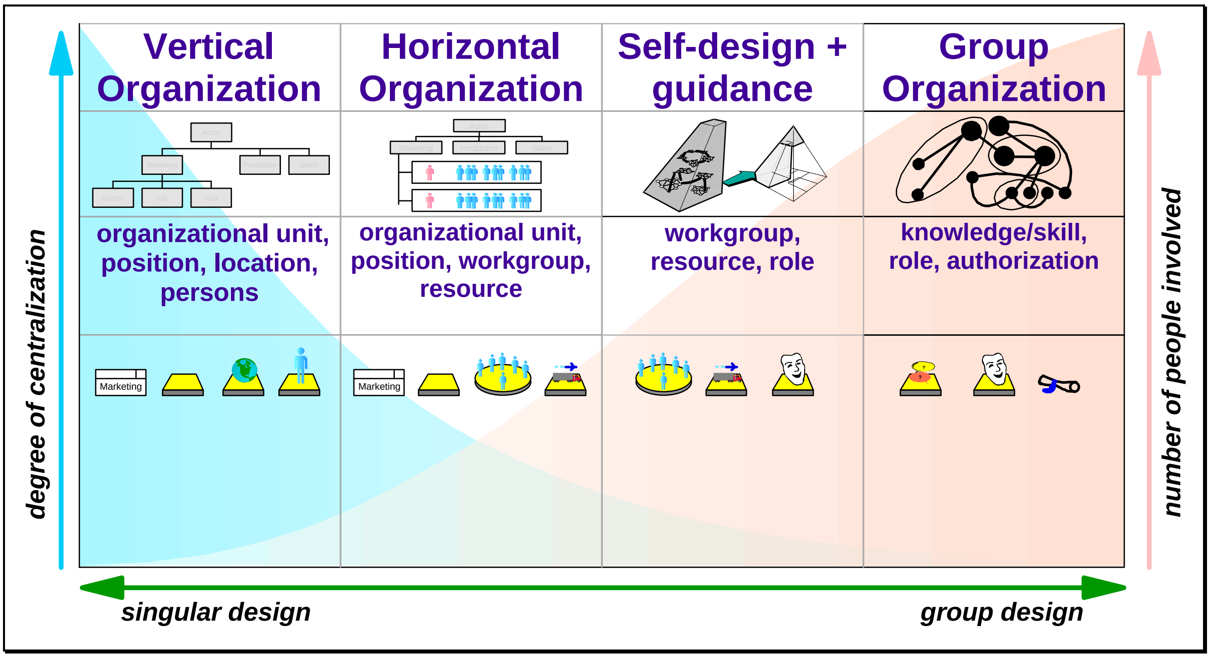


Figure 5-2: The GroupOrga continuum for support of future organizational forms

Vertical organization and *group organization* describe the two extremes on a continuum of possible organizational forms.

The left side of the scale is a bureaucratic, tayloristic organization with a linear dependence between the operational entities. Such vertical organizational forms can be designed and illustrated with the entities that resemble these forms, such as *organizational units*, *position*, and general hierarchical subordination. The tool support of GroupOrga can also be adapted to this form of organization design. A central enterprise repository may be set up and all design decisions can be made from a single organizational expert. This singular design approach *can* be implemented with GroupOrga, however it is not the marked goal.

From here a horizontal organizational form (common in many organizations) is the next step on the continuum. This configuration has already often been reached by firms. It is an early form of project organization within a hierarchy. Further to the right, a combination of top-level hierarchy and bottom-level self-organization can be found. With GroupOrga, this is obtained through its possibility to assign administration responsibilities to different people. While top-level structuring can be carried out by team leaders, the refinement is undertaken by group members. The essential GEIMM entities are still *organizational unit* and position, but the flexible entities, such as *workgroup*, *role* and *resource* gain significant importance in this form.

As the final consequence, Figure 5-2 positions a network structure to the very right of the continuum. In this case, the team's collaboration within GroupOrga excludes all forms of hierarchy. The entities *organizational unit*, *position*, and *workgroup* are superfluous and groupings of employees are defined by their *knowledge/skill*, *authorization* and *role*.

Although Figure 5-2 gives the impression that the forms illustrated are the only possible shapes, the movement on the GroupOrga continuum has to be understood as a smooth transition from one to the other. Indeed, there are many substages between the sketched configurations. The two extreme poles on the continuum cannot be advised as practicable structures. Whatever degree of centralization is desired or how many organizational members are to be involved in the design process is almost infinitely variable within GroupOrga. Section 5.2 gives a pragmatic approach to overcome this contradiction and introduces a parallel structure which combines hierarchical coordination with self-organized entities such as teams and skill-groups. The enlisted elements of the GroupOrga vision are introduced in the following sections.

5.1.1 An Enterprise-Wide Data Model

GroupOrga's specification of a general form of representation for organizations (chapter 4) is flexible enough to be applied to various kinds of organizations. Because it is impossible to develop a single, fixed enterprise model that fits the large range of organizational constructs and needs, the GEIMM provides a means for the representation of basic organizational entities. Based on that, the enterprise model supplies building blocks for the representation of basic organizational entities. It has a 'toolkit' character. This means that the entities can be chosen and then applied for the modeling of a specific organization. This allows a compatible modeling of different organizations.

The GEIMM is a library of entities that defines the objects of an organization that are generic across any enterprise, and it can be employed in defining a specific enterprise. With it, a shareable representation of knowledge—one that minimizes ambiguity and maximizes understanding and precision in communication—is available. In addition, this formal representation eliminates much of the programming required to answer *simple* questions about the enterprise.

In contrast to starting from scratch when implementing WfM in an organization, with this generic model the system designers are provided with a set of organizational entities, allowing them to quickly move on to the realization of an organizational structure model. Moreover, the workflow designer benefits from the fact that a complete organization model already exists—there is only a small chance that entities may be 'forgotten' when setting up the organizational subsystem for a WfMS. Thirdly, by using the GroupOrga enterprise model, all other parts and members of the organization have the chance to understand what is represented and participate in the organization design process. Section 5.1.5 focuses on this aspect of participative design.

Hence, this generic, reusable data model:

- Provides a common lexicon that each agent can understand and use
- Defines of each term in the lexicon in a precise and unambiguous manner
- Implements the definitions in a way that enables WfMS to automatically allocate workflow tasks to organizational entities
- Allows for implementation of a query language that enables users (human or computer-based) to answer *common sense* questions about the enterprise
- Defines a set of symbols for depicting a term or the concept constructed in a graphical context

The dynamic GroupOrga enterprise model provides process, infrastructure, and information views of an organization in a repository. These views are cross-referenced to provide an integrated picture of the enterprise. Although all three views are discussed in chapter 4, the main interest is in infrastructure models that are "a special aspect of *enterprise modeling* which is an attempt to cover all aspects of an enterprise ..." ([Rupietta 1994], p. 115).

5.1.2 Integration with Workflow and Office IT

Since the GroupOrga system represents structural information, its framework allows direct access from the appropriate workflow and office systems and services whenever needed. The user is not forced to switch application platforms when structural information is retrieved. Hence, GroupOrga provides means for the inclusion in external process management services. The GroupOrga enterprise model combines general structural information with the knowledge about the communication addresses. GroupOrga is also set up as a distinct, independent framework. It is not a feature or component of process-software and office-software, but a combination of its own applications.

The starting point of the approach chosen here is the information system. In this framework, the organization (or in a narrower focus, the office) is understood as a system that has to process information. The division of the enterprise into organizational units, subunits, workgroups, and so on, can be understood as its information system. The consequence is to describe all organizational entities by means of expressions from the information system. This approach is currently the best one to bring the fastest results. It describes an organizational system as if it were part of an information system.

The main advantage of this approach is that the specification of interfaces between the two systems meet the high formal demands of computer-based tools such as WfMS. In GroupOrga, the process organization and the structural organization (section 2.1.6.1) can be designed and visualized by means of modeling tools—the business processes are designed in process models and the organizational structures are *simultaneously* laid down in organization models ([Ott/Nastansky 1997d], pp. 94f.). In the following, the models can be simulated, analyzed and optimized. Hence, an integration not only of the two disciplines but also of the necessary modeling was reached.

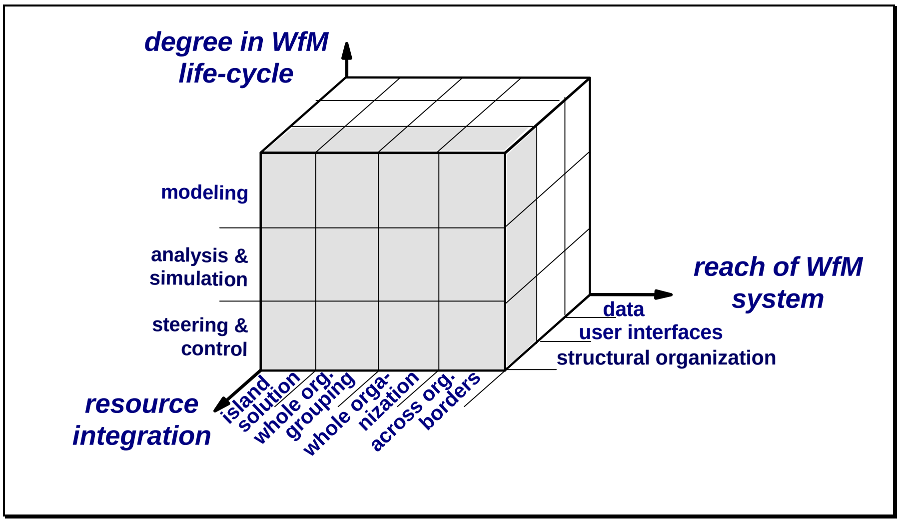


Figure 5-3: Dimensions of workflow integration

Heilmann's dimension of integration between workflow management and resources is an abstract form of GroupOrga's integration of CSCW systems and organization design ([Heilmann 1996], p. 151). The degree of integration with the structural organization defines how much structural elements can be modeled seamlessly and used in WfMS (see the shaded part in Figure 5-3). This includes organizational units, organizational charts, posts, employees, roles, locations, and authorizations. A *complete* integration requires these structural elements are modeled before or during the design phase. Moreover, they must be assignable to process steps either at design time or during run-time of a process. GroupOrga meets these requirements.

5.1.3 Flexible and Evolutionary Organization Design

The capability required in section 3.2 under the header of "Flexibility in Organizational Subsystems" can be provided by a dynamic organization component for WfMS that enables the workflow participant or the organization designer to consider various environmental scenarios and their impacts on the organization and the workflow. GroupOrga is such a component that functions as a support tool designed to facilitate the adaptive change process within the organization.

In addition to the fact that the underlying enterprise model of GroupOrga provides numerous means to be applied flexibly to various kinds of organizations (section 5.1.1), the framework itself and its tools allow dynamism and flexibility in the organization design process. For the adaptation into the personal working environment, it is useful for groups or even individuals to refine or extend the current enterprise model for their needs. For example, it supports the design of teams that are composed of members from various backgrounds. These teams are often restructured and temporary.

Everyday organizational rearrangements require model modifications. This makes it essential that the organization model can be adapted to changes that happen after the development of the initial model. In GroupOrga, this initial model is documented and managed in a distributed database environment ([Ott/Nastansky 1998b]) that gives everyone quick and easy access to the data. Due to this advantage, changes can be carried out with little bureaucratic overhead.

Most importantly, with easy-to-use graphical tools, this procedure can take place daily and be executed by everyone in the organization. Thus a *continuous* and *evolutionary* design process is guaranteed. For example, it is supported by the provision of the organization object modeler introduced in section 5.5.

With this capability, GroupOrga meets one of the seven *dynamic modeling requirements* from Li and Lochovsky ([Li/Lochovsky 1996], p. 195) in which they cover the same need under the term of *flexible and dynamic composition*. In their example, units may be composed in various ways, and may be created and destroyed at any time.

Likewise, with GroupOrga it is possible to specify detailed organizational rules, depending on the degree of predetermination of a process. This includes, for instance, which role is covered by which position within an organization during a certain period of time. In Li and Lochovsky's collection, this requirement is listed as *flexible and dynamic binding,* which allows organizational entities to be bound to other entities in flexible and/or dynamic ways. In other words, for workflows to be easily adapted to changing environmental circumstances, the GroupOrga framework does not try to cement organizational structures in static role models (see [Scheer/Nüttgens/Zimmermann 1995], p. 433). This is true, since no fixed and universally valid structures are stipulated.

5.1.4 Computer-Based Organization Design

This section addresses two of GroupOrga's aspects concerning the requirement that an innovative form of computer-based support in organization design procedures is necessary. First, the end user tools (tools which are available at every organization member's desktop to assist in the design and administration procedures) are presented. Second, the technology that supports GroupOrga's overall architecture necessitates a look at computer-based organization design from a completely different angle. For this perspective, a broader view onto computer assisted design of organizational structures is taken, and basic architectural consequences are addressed.

In contrast to Grudin [1990], who discussed computer interfaces of stage 4 or 5 in fields other than organization design, the GroupOrga concept makes the first suggestions for tool support in stage 3 and stage 4 with the tools implemented (see section 3.2.5, Figure 3-1, p. 60). The first step is being made with the use of color, bit-mapped graphics, windows, and other capabilities becoming more widespread for purposes of organization design in process management systems.

For information and browsing purposes about organizational information, the graphical GroupOrga interfaces provide the means for the visualization of overviews (of workgroup networks, organization charts, role accumulations, and so on) and rapid browsing through structures. Although the modeling tools in the GroupOrga framework also use the well-known and widely-accepted organization chart and the flow chart as descriptive elements, other ways of depicting organizational circumstances have also been explored. Detailed information about particular organizational entities can also be presented. With the tools in the framework, this information can be presented by using different media types. For instance, the enterprise repository is capable of managing and representing different media types that exceed textual description, such as photos and graphics.

Besides these desktop computer-based applications for organization design, which are used by everyone taking part in the design process, yet another computer-based application platform area has great influence in the GroupOrga framework: groupware.

The support of the system with a fundamentally open and distributed groupware platform has various, essential advantages. These advantages result mainly from the characteristics of groupware presented in section 2.1.2. In the following, some of GroupOrga's advantages (in terms of how they have become possible because of groupware technology) are discussed.

Scalability

In general, GroupOrga is designed as a variable approach for a growing orientation of organizations towards WfM solutions. This means that the framework supports large workflow and office management solutions, as well as mid-size to small projects. With this scalability, it is ensured that a sufficient collection of organizational data from the base of an organization can be collected. In particular, GroupOrga supports the design and planning of mid-size design projects, since the underlying groupware platform can effectively be installed for a smaller numbers of clients, as well. For GroupOrga, there is no need to have large scale computer systems or huge numbers of client computers and users for it to be correctly sized. With this characteristic, the proposed framework addresses the specification given in section 3.1, where some 60 or more clients have been identified as a reasonable size. Another aspect, which is somewhat connected to scalability measures, is that of distribution. Distribution is elaborated in section 5.1.6. The groupware communication platform makes distributed design and planning with GroupOrga possible.

Integration into operative system environments

Almost every existing organization has some sort of historically grounded operative system for electronic documentation, user access directories, and management of data. It was stressed earlier that the ability to integrate an organization design system is necessary in principle. This results in the need for organizational data to be accessible from other systems (section 5.1.2). The chosen groupware platform possesses a number of interfaces that allow for this integration into heterogeneous IT landscapes. Moreover, the platform provides for a complete integration into the electronic infrastructures that are currently expanding enormously.

5.1.5 Participative Organization Design by Everyone

Participation and distribution are two different, yet interconnected aspects of the GroupOrga approach. While the latter refers rather to the technical ability of the GroupOrga framework to support people working at separate locations, the former denotes the wish to integrate not only a few experts at different locations, but all organization members. In other words, *participation* entails a non-technical quality of such a group process.

Section 2.1.3 and parts of chapter 3 have thrown a light on groupware in learning organizations and the overall learning aspect in organization design. Section 5.1.5 demonstrates how a participative design process, as guided with the GroupOrga framework, can result in a learning organization structure. It is an *operational view* of organizational learning and how it can be supported with the framework presented in this research.

The underlying concept of organizational learning is the notion of continuous learning. Through continuous learning, an organization is able to respond more effectively to change. Hence, the learning organization is one which is continually improving its organizational structures and processes in response to environmental changes. However, learning organizations are only possible because individuals learn. The requisite environment for continuous learning is an organizational structure technology that empowers the individual to document real-world related circumstances and development decisions.

Therefore, there is a need of an organizational structure that supports individual exchange of new ideas, problem solving and innovation. Moreover, a technology that facilitates immediate and open communication across the organization as well as with external entities, such as suppliers and customers, is a basic building block.

With the GroupOrga framework, the culture of a learning organization about organizational structures is based on the interaction of the individual, team, and organization level. For example, individual skill acquisition is the foundation for team learning. Only when it is made public which skill the individual has recently acquired, and which skills or knowledge is already available in the team, can the other team members make use of it. In contrast to traditional development efforts, the learning organization is characterized by the individuals self-assessing, self-directing, and self-documenting their training and development successes. The driver is *knowledge*. The individual determines what type and what level of knowledge should be documented and thus be made available to the organization. In GroupOrga, the method of delivery may be the groupware enterprise directory and the use of GroupOrga tools.

In practice, the knowledge documented (in business-wide databases for example) and thus offered by the individual is focused on this employee's positioning in the organization. Though the knowledge is task oriented, it is usually generic and transferable to other tasks.

Ideally, it makes the individual more marketable. Examples of generic skill or knowledge include: C-programming skills, expertise in graphical user interface (GUI) techniques, and outstanding communication skills.

Team learning is an extension of individual learning. Whereas individual learning is characterized by knowledge acquisition, team learning is characterized by knowledge transfer, information sharing and problem solving. A functional skill management environment helps in this kind of information distribution. The GroupOrga enterprise directory serves as such a skill management environment. It offers the ability to document one's skills and knowledge, and to retrieve this information by other team members for use in the organizational processes.

One topic of organizational learning is that of organizational infrastructures. Individual members of an organization find out which organizational form is or was the best to solve a given problem, and they *learn* how to (re-)arrange themselves in times of change. The employees can compare various structures and forms within their own department and workgroup over time, and they can judge which appears to be the best in a given situation. If this knowledge about structures is conserved and constantly updated, team learning, and subsequently organizational learning, can be facilitated. Also, the organization must provide the technological mechanisms to support this specific form of individual learning.

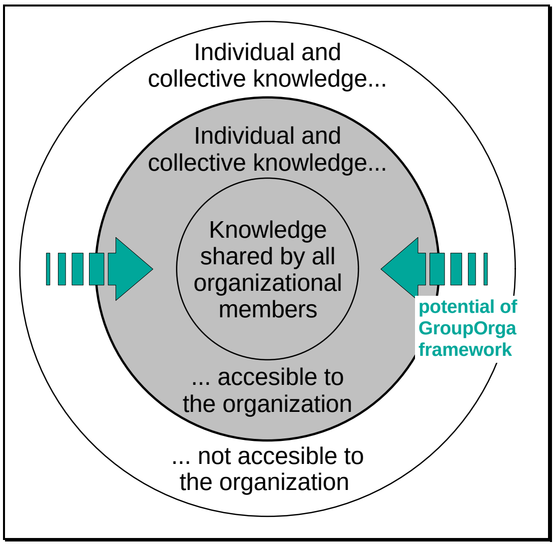


Figure 5-4: Layers of organizational knowledge

Bauer [1998] argues that a framework such as GroupOrga can facilitate the central activity of the learning organization. The purpose of which is to transfer personal knowledge to other individuals as well as to the entire organization (see Figure 5-4). Hence, the enterprise directory should supply visibility of the modeling concepts to everyone. That is, the organizational model entities should be visible to and changeable for the users. The GroupOrga framework provides tools for this. Graphical viewing applications which illustrate the current organizational structure and can search for particular skills are available. In addition, with the appropriate access rights, these graphical tools assist in additions, changes and modifications. These tools allow users to recognize structural correlation of the current organization model; thus, the user can see how the organization is set up and appropriate changes can be made when necessary.

5.1.6 Distributed Organization Design

GroupOrga meets the requirement that organizational data should not be administered by a single organization expert. Different people can be responsible for the management of different types of information in the enterprise model. With GroupOrga, this may vary from technical administrators to the human resources department, as well as to individual employees. The framework provides the means for modeling in a distributed manner. In addition, it supports coordinated data management.

An example may illustrate how the distributed management with GroupOrga takes place. The human resources department creates a new entry in the existing enterprise repository with the basic data of the new employee. Then, the technical administrators are automatically informed about the new person in the organization. They install the appropriate services for the user and enter the relevant technical information into the repository. Simultaneously, the new employee's supervisor keys in information about this person's relation in the relevant department, such as responsibilities, workgroup membership, and roles. The new employee may now include additional information that he wants to make public. A similar procedure may take place when information in the enterprise repository is modified, which in turn requires changes by the technical administrators or the employee.

Distribution in GroupOrga involves spatial distribution, as well as administrative distribution. In other words, not only is the repository structure itself spatially distributed, but also the people responsible for administration reside at different locations.

This support for the distributed management of the organizational information is provided by the administration responsibility elements of the enterprise model's entity classes. In other words, the GEIMM supports the assignment of administration responsibilities and rights to different people in the distributed environment. This is achieved by the consideration of schema administration rights and administration responsibilities at the meta-object level, which separates the organizational distributed administration from the technical distribution (see section 5.2.4).

Furthermore, this distributed administration is assisted by the hierarchical naming schema which is implemented in the GEIMM through the hierarchical organizational unit entities.

Distribution, as the main characteristic of GroupOrga, is addressed intensely in section 5.2.

5.2 Insights into Distributed Organization Design

Distribution of organization design among the people responsible for it is a natural advantage. Requirements for it and the advantages of distribution in this scenario were discussed in chapters 2 and 3. This section consolidates the development within the GroupOrga approach, which deals with the techniques for distribution of organization databases across networks. It distinguishes distributed databases vs. application-specific distribution concepts, such as distribution according to organization structure.

A data architecture defines the arrangement of databases within an organization. Although every organization that maintains data has a data architecture, in most organizations this architecture is more the result of the evolution of application databases in various departments than of a well-planned data management strategy. With organizational databases this must be different. Because these enterprise directories exist only in very low numbers, with most of them in an application specific, a distributed data architecture design can still be created from scratch. This section also examines different forms of cooperation in organization design and uses these findings for setting up the GroupOrga data architecture.

A lack of an information management strategy often results in distinct directories, having multiple attributes, coding schemes, and values across directories. The data management strategy proposed here does not, however, imply that all these distributed databases should be replaced by new repositories. In contrast, in GroupOrga they remain, but there should be a disciplined structuring of the repositories among corporate and functional application repositories in the distributed scenario. Replication is the technology used in the framework to support this goal.

5.2.1 The Concept of the Distributed Organization Repository System

The goal of a distributed system is to coordinate collaborative work with distributed computer applications. In order to allow for such a coordinated collaboration, a suitable technical infrastructure, which supports the distribution of applications and data is necessary. In GroupOrga this infrastructure is a client-server relation, which is the most common model for cooperation between computer-based applications in distributed systems. In this case, the server offers the services, such as providing information from organizational repositories, replicating data between such repositories or offering whole application programs (such as the graphical modeling applications). Each activity is started when an individual worker requests a service, such as asking for information about an organization's structure, changing the information, and entering additional information.

As in each client-server model, the roles *client* and *server* are only temporary in GroupOrga. In other words, a server in the distributed GroupOrga environment can become a client itself, when requesting a service from another server in the architecture, since the initial request could not be answered sufficiently.

Closely connected with the server is the server-interface. In GroupOrga, this interface allows the user to retrieve data from an organizational repository or to add data to it from various applications. The interface translates the request for the organizational repository and again retranslates the answer. The GroupOrga server component offers different interfaces for various applications available on the client (graphical modeling tools, analysis tools, modeling languages, and so on). In order to prepare a request, the client software allows the user at the local desktop to define it (for example in a graphical notation), then the application translates it into the syntax and semantics of the server interface and later receives the result.

This client-server model used for GroupOrga is the analogy to real world organization design processes: Employees request changes to certain models which are then distributed into the overall system.

The knowledge required to achieve a complete and updated organizational model is broken down so that people in the organization have to work together to create this model. This division of knowledge makes different individuals responsible for different modeling tasks. In the GroupOrga framework, every subfunction in an organization can have its own client computer and a separate repository that contains only the information valuable to this subfunction.

To the extent that the structural information required by a subfunction in the organization is unique, this information is made available only to that subfunction on their own client computer. Some structural information is needed by more than one individual and more than one subfunction and is thus distributed over the client-server distribution channels.

Finding the best configuration for a given distribution of structural information is a complex problem. At how many sites should organizational systems be located? What hardware, software and staff are justified at each site? What communication links should be established? An alternative strategy for finding a reasonably cost-effective solution to the GroupOrga distribution problem is to use a heuristic solution technique. In considering the possible alternative locational configurations, there are some common sense rules that can serve to eliminate many distribution alternatives from consideration (see [Gessford 1991]). Using these rules simplifies the distribution problem:

- Specific structural data should be at the same location as the people who use and edit it.
- Each fragment of information should be placed in the repository at the location where it is most frequently accessed. This fragment should be replicated at other locations if this reduces communication costs.

These heuristics draw on the fact that employees do need to know the peculiarities of the surrounding structure, the workgroups they belong to and the roles they play or the skills they have. Moreover, the heuristics indicate that communication costs are yet another aspect. The last heuristic has the defect of ignoring the fixed site costs, that is the costs of establishing a computing capability are ignored.

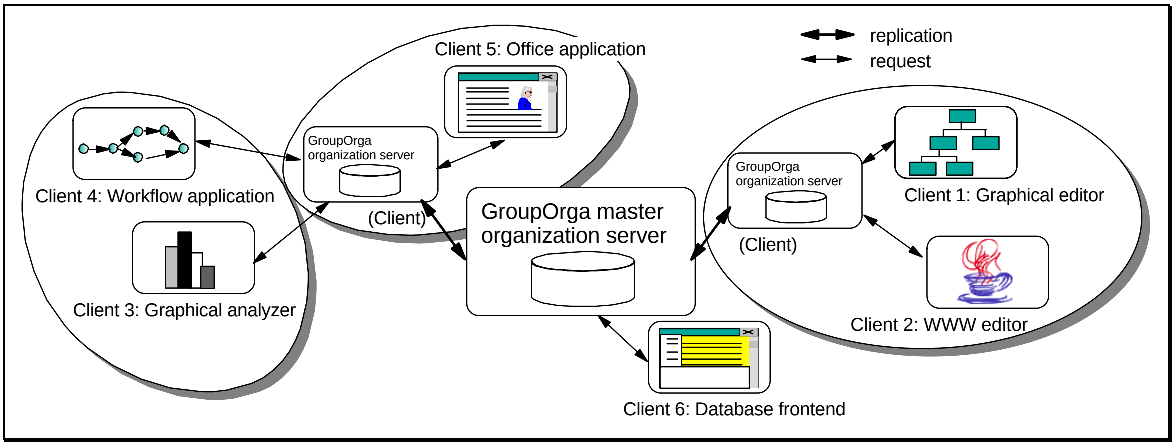


Figure 5-5: GroupOrga client-server architecture

However, in GroupOrga, distribution is defined through a groupware-based client-server model that has several distributed servers available to many clients, as illustrated in Figure 5-5. In this architecture, the costs for the client are almost insignificant and servers are provided within the organization computing infrastructure for other computing purposes (like workflow). Distribution in GroupOrga—and thus the inclusion of every organizational member—is not defined by large scale computing sites. Distribution is more a question of providing simple, small scale tools to the end users to let them participate in the process and use the existing replication technology for background distribution purposes.

5.2.2 The Top-Down Modeling Approach in GroupOrga

GroupOrga proposes a distributed modeling strategy that is based on a top-down procedure. For organizations with more than one site, the best solution most likely is to be found somewhere between the extremes of total centralization and total decentralization of the design process.

As was outlined before, the distribution of directory information can be allowed onto a variable number of information systems (clients and servers) within the network. This guarantees an enormous scalability of the data model. Each node in this network can store an optional portion of the complete data set.

Through a specification of which partial information is stored on which node in the network, a distributed design and administration of the complete data set becomes possible. In this environment, a central authority might be responsible for some coordination of the structural information. In addition, the decentralized end users in the organization with their respective GroupOrga information technology provide the detailed organizational structure information about their particular role.

The central point of this approach is that continuously designing the structure of a networked company requires at least two levels of organization design. The first level is a top-level (or superordinated) design, which is the responsibility of the coordinators in an organization. (In a *traditional* organization, this group is called *senior management*.) This group is concerned with framing and constituting the infrastructure of assets, organizational units, workgroups, resources, hierarchies, and management practices. These predefined structural elements are used by the individual designers throughout the organization to perform the second level of design. The second level is a termed bottom-level design, which is a self-design process. Of course, this self-design must not be restricted to only one level below the structuring component. On multiple levels it might involve the individuals using the proposed infrastructure to shape their own working environments and organizational sub-structures. For simplicity reasons, the following discussion focuses on two levels only.

Top-level and bottom-level design cooperate

All networked (bottom-level) end users in GroupOrga model and disclose their own structures, competencies, roles and workgroups (as far as they are considered important for the whole organization but not confidential) to help build the unique structural appearance towards the environment. In this context, these two levels of design are described as a "network floating on top of a hierarchy" (Figure 5-6).

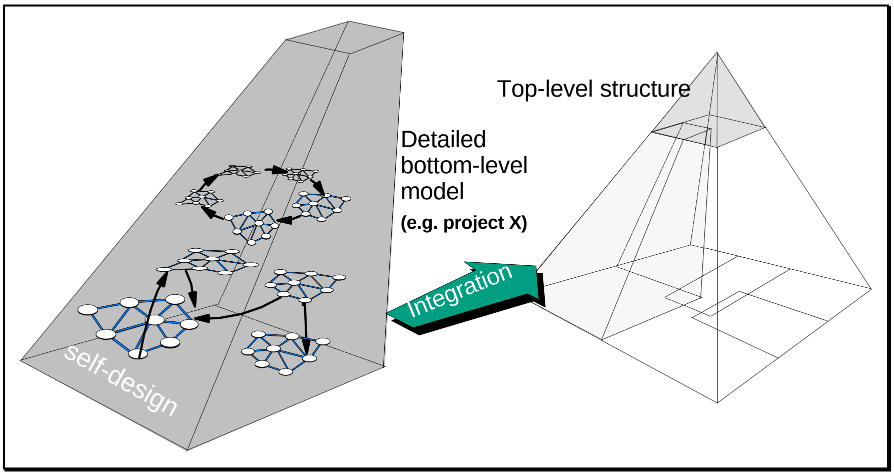


Figure 5-6: The top-level design and bottom-level self-design

Top-level organization design

The *top-level organization design* consists of establishing major shared infrastructures and hierarchical elements in which the organization will operate long-term. Although they must be flexible, these infrastructures are the points of stability in which the organization conducts work and by which the outcome is effected. Using the term *hierarchy* in connection with *flexible design* for this top-level management seems to be a contradiction. However, what is designed here are the rough bounds of the flexible infrastructures, while the detailed planning of the organization's structure itself is done by the knowledge workers, that is the multiple participants in the design process (see section 5.1.5). To operate in an environment of high uncertainty, the organization must rely the participating organizational members to be innovative and continuously learning.

The broad structural outline of an organization exists primarily for organizing its human assets and may have very little to do with how work actually gets done in the network. This structural profile is the functional hierarchy on top of the self-designed networks of relationships between the partners in different workgroups, groups or projects. Although senior management is no longer explicitly responsible for creating this structural basis, they are responsible for providing tools (like the GroupOrga tools described in this framework) for doing so.

Bottom-level structuring processes

The multiple *bottom-level design* processes which actually define how the integral parts of the organization are structured are performed by the workers at the distributed sites. The distinction between top-level and bottom-level design made here should not be mistaken for the opposition between centralized and decentralized performance of tasks. Top-level structuring is a superordinate task, but its main purpose is to provide a framework for the necessary outcome to be achieved. Nor is bottom-level design *only* decentralized decision making. The purpose of self-design in GroupOrga is to allow the knowledge workers to shape the surrounding environment in whatever form they find feasible for carrying out the tasks in the most beneficial manner.

Organizational structure is the most apparent, most discussed variable in this study. The *network* does not totally replace the hierarchy, but operates with it. The network structure is designed by anyone who needs to get something done, whatever the designer's level in the functional hierarchy is. Because the overall network structure of the organization is the result of a collection of many distributed workgroups and people, it may be extremely complex and shifting constantly. No one person, at any level in the organization, has a total picture of what the structure looks like, but nevertheless it is optimized due to the optimization of the sub-structures by the responsible knowledge workers.

In the first stage of GroupOrga implementation, the decentralization possibilities can be described in general terms by considering the kinds of locations and functions organizations typically have and the data that may be economical to place there. Most organizations have a headquarters, for example, where people who do the top-level structuring, among other management, administrative and planning functions, are found. If this location is centrally organized, it may well be the place for a master enterprise repository. Then there are other kinds of locations, such as branch offices, laboratories, and warehouses. Each of these locations needs different information about organizational structures for their business processes and can contribute different entities to the organization's repository. However, each of them can give a detailed bottom-level picture of how work is organized, and how it can later be integrated into the overall organizational model.

When locating this stage on the GroupOrga continuum in Figure 5-2, it is found somewhere on the border between a horizontal and a fully self-designed organization, still in need of top-level coordination authorities. From here, distribution no longer sticks to geographical distribution in the design process. The GroupOrga framework enables participants to design the details of the structure they are affected by, not only those which are incidentally located at the same branch.

Technology support of concurrent bottom-level design

The GroupOrga structural design process involves more than one person at the bottom-level. It has to reflect this by providing support for the cooperation of designers. First of all, designers access and modify the GroupOrga repositories concurrently. Hence, the framework allows a maximum of concurrent activities by the participants of the design whenever they work on different *clusters* of the repository. In this sense, a cluster may be a set of connected entities in a repository: all project groups in a specific country, organizational units dealing with a particular product, and roles for a specific purpose. Designers working on the same cluster of the repository should be prevented from accidentally destroying each other's work. This design occurs in different forms, which, in the framework, are labeled *individual*, *loosely-coupled*, and *tightly-coupled* work. These modes differ in the level of *awareness* each designer has of the activities of the co-designers.

In *individual* work, a single participant may manipulate an entry in the repository. Through annotations in the repository, the person can inform other users about changes.

In *loosely-coupled* work, several users are working in the same cluster in the repository. During this mode, they need to be aware of each others' activities. The *top-level* design authority could, in this case, preside and watch the design activities. Technically, the respective GroupOrga tools are all in *loosely-coupled* mode. Designers may be made aware of each other via a list of all concurrent users by highlighting the entities locked by other users. Currently, GroupOrga supports no locking at the repository level, but uses the underlying groupware platform's technique of conflict detecting.

In *tightly-coupled* work, designers must cooperate and coordinate their work in synchronous, conference-like meetings. In this mode, the users should be provided with functionality requested in concurrent-authoring scenarios. The GroupOrga tools are not designed for this kind of work. In the framework, it is assumed that the people who edit organizational entities are the ones who have the rights and the knowledge to do so. Hence, the occurrence of conflicts should already be prevented on the organizational level, and not on the tool-level. In the event of conflicts, GroupOrga makes use of the underlying groupware platform's technique of conflict detecting.

5.2.3 Employing the Replication Concept in GroupOrga

In formulating plausible distributions of parts of an organizational model to sites other than a central site, two types of distributions are possible. One is to keep different occurrences of a certain type of organizational information at different locations. This is called data partitioning. In this case, the system knows which occurrences are at which sites and any requests are referred to the appropriate site. The alternative to partitioning is to maintain logical copies of the data at various locations. This is called data replication, and was described in section 2.2.1. It requires that a modification (or an addition or deletion) of organizational entities in the repository at one location must be made at other locations, as well. Thus a distinction must be made in this case between merely retrieving a copy of organizational information and changing it. Only for the latter type of access it is necessary to revise the comparable data at the other location.

For GroupOrga, the possibility of partial replication was chosen. It is not necessary for all organizational entities in one organizational repository to be replicated to all locations and vice versa. For example, as an alternative it might be considered to replicate only the structural information necessary for a specific factory location. The infrastructure information about this factory would be maintained at (minimally) two locations, while the data of all other sites would only exist at the central location and where the data is relevant.

Changes in replicated data in repositories made at one location must also be made at the other locations where the information is replicated. Of course, if the structural information is not replicated, no updating at other locations is required. So, distributing replicated data between GroupOrga organization structure repositories can result in an increase in communication costs if data values are changed frequently. Regarding the overall philosophy of GroupOrga, this may happen quite often, since it involves an evolutionary and continuous design procedure. The more volatile the replicated data, the more costly it is to maintain the replicas of it. Hence, in order to reduce communication expenses, the strategy in the GroupOrga framework is to keep as much structural information in the local repository only as possible. However, with the top-down modeling strategy explained in section 5.2.2, this is not too difficult. The idea is to keep the long-lasting overall structures untouched as a guideline for the organization's daily work. These top-level structures of organizational units or workgroups may thus be replicated to some or all sites, whereas the people who work at these locations undergo frequent (continuous) restructuring processes of their local working environments. In automated workflow or office environments, work is assigned to the coarser entities in the organization's structure, which is then broken down according to the detailed specifications at that site.

For the overall data security in such a distributed repository architecture, there are two choices:

- One (or more) of these repositories contain(s) all the structural information available in the organization
- Each distributed repository holds only some information while other information resides at other locations

Which choice is better cannot be determined for all possible cases. Most likely, however, the GroupOrga framework suggests that there will be a central repository (at the headquarters) which holds all structural information. One good reason for setting up the enterprise directory like this is that of security. If one distributed repository crashes, all information can be replicated back to it from the central enterprise repository once it is running again.

Whatever choice is taken to replicate the distributed GroupOrga directories with each other is again up to the organization. In the following, two technological options are further outlined.

In case of the *single-master replication* (gray arrows in Figure 5-7), the distributed repositories do not communicate with each other but only indirectly via a central repository. Each requested change (A, B and C) is forwarded from the local directory to the single master. Once all changes have been documented in the single master (or after a certain period of time), the changes are propagated back to all local directories.

The *multimaster replication* allows changes (X, Y and Z), immediately when each of the three clients requests such a change (blank arrows). Changes are then forwarded to multiple directories (and not only to the single master), depending on configuration. These changes continue to propagate until all local directories have copied all modifications.

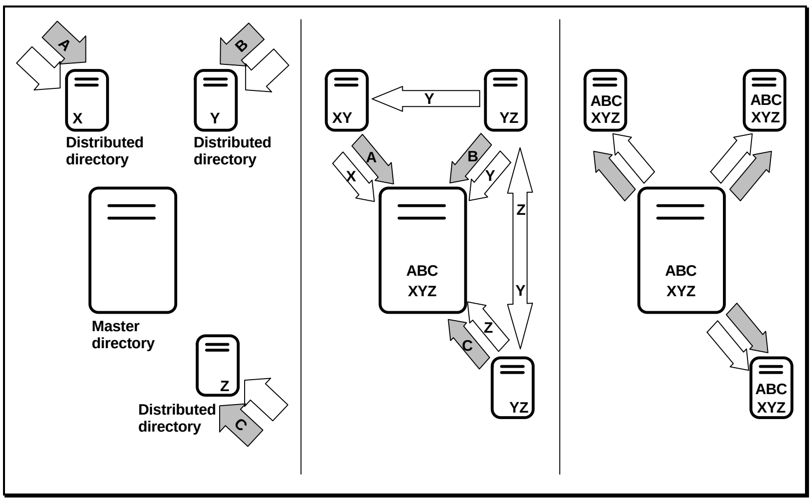


Figure 5-7: Single-master vs. multimaster replication

Although both configurations have one (single) central repository, their respective designation is still correct. The term *multimaster replication* refers to the fact that all distributed repositories have the right to realize the changes as if they where master repositories themselves. It does not relate to the position (the center) or its importance (being located at headquarters).

Because the single-master configuration is less scaleable and multimaster's quicker change propagation, multimaster replication is the architecture which is recommended for GroupOrga setups.

5.2.4 Installing Distributed Directories for Participative Design

The overview of distribution in section 5.1.6 mentioned responsibility elements of the enterprise model's entity classes which assist in setting up a distributed directory. This section examines this point in more detail.

A special feature of the GroupOrga concept is that it provides for the specification of administration responsibilities of the data model's entities in distributed environments. For allowing this, a layered administration model was defined in GroupOrga. Different administration rights on entities can be assigned to different administrators and users in the distributed environment. Therefore, some users may be allowed to examine or modify certain parts of the infrastructure, while for other parts they are refused.

To simplify the specification of administration rights at the entity level, each entity within the infrastructure model may contain default rights. These rights apply to all instances of such an entity, unless they are overwritten by specific administration rights for the entity.

For the specification of these administration rights, two major methods have been distinguished in the project ([Nolte 1997], p. 9):

- Access control lists and replication technology on the database level
- Specification of supplementary access rights on the entity level

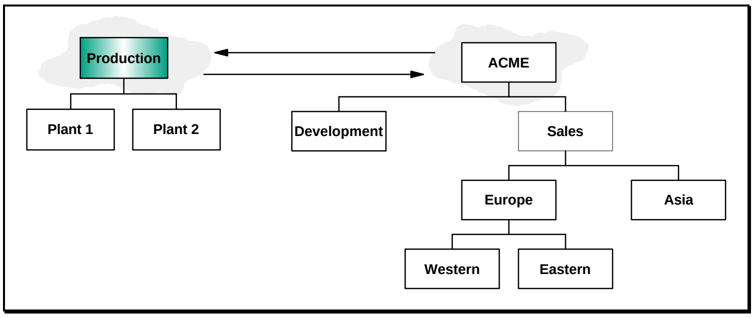


Figure 5-8: ACL specification and replication in distributed directories

Access control lists (ACLs) in combination with replication technology are the technologies offered by the underlying groupware platform (see section 2.2.1). An ACL may contain an entry for each user or group of users of an organization database. The ACL specifies all operations which the person or group is allowed to perform to all entities (documents) in the organization database. Thus, ACLs are associated with organizational databases. For the purpose indicated above, however, its low granularity is a disadvantage. It is not able to restrict users to only parts of the organizational model, and hence to only parts of the organizational database. However, the ACL concept is used in GroupOrga to define the distribution model between larger organizational entities, such as servers at different locations depicted in Figure 5-5. ACLs can be used in a scenario where a location (for example, the production department) designs its own infrastructure in its replica of the organization's enterprise repository. Afterwards, this repository is replicated with the central repository shown in Figure 5-8. According to the ACL settings, only changes which affect *production* are exchanged.

Below this level, a more finely tunable concept was applied: the specification of supplementary administration rights on the entity level.

Three types of administration rights on the entity level are to be distinguished:

- Read access
- Insertion access
- Design access (of a new entity)

When a user has *read access,* this means that this user has permission to read the information contained in this organizational entity. The default read access is not specified, which generally gives everybody the right to read.

With the *insertion access*, admission in GroupOrga is granted for assigning organizational members to infrastructure entities or for revoking their assignment. That is, with insertion access, the user can be specify who belongs to which organizational unit or workgroup, who has what role and who has what position. Owners of insertion access can also allocate read access and insertion access. With insertion access, read access is automatically granted, however, no new entities can be created.

The *design access* is usually assigned to administrators of the different higher level entities in the organization. It includes all rights of read and insertion access. Moreover, with design access, new organizational entities can be created or dissolved and their attributes can be modified.

As chapter 4 revealed, in GroupOrga there are different types of infrastructure entities, some of which are purely non-hierarchical (roles, knowledge/skill), while others are arranged in a hierarchical manner (organizational units and sometimes workgroups). For the latter type, an algorithm to determine the default setting of the administration rights has been conceptualized in GroupOrga. This is explained in the following.

From the GroupOrga organizational perspective, there is no need for members of an organizational entity (unit, workgroup, role grouping) to get detailed modeling rights in the current sub-structures of a neighboring organizational entity. Rather, it is important to know who to contact within this entity, what services are offered by it, and what the overall structure looks like. In case of a cooperation between two organizational units or workgroups, only read access rights are needed in order to ask questions and establish contacts. It can also be argued whether it is necessary for members of an organizational entity to actually read the *sub*-structures of neighboring entities or if this read access should also be restricted to reduce complexity.

Based on this analysis, a rule for the technical realization of distributed modeling in GroupOrga was defined: *Default read access in the overall organizational model for the members of a hierarchical organizational entity is granted to all subordinated entities, to all higher entities and to peer entities, but not to their subordinates.* Figure 5-9 illustrates this rule with an example. All members of the unit *Sales* have default read access to the indicated adjoining units.

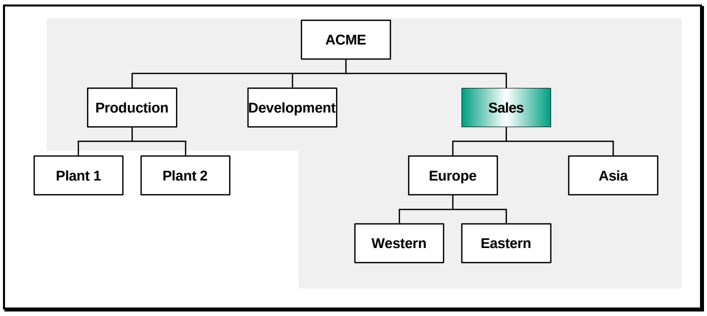


Figure 5-9: Default read access for members of *Sales*

According to the top-down modeling approach of section 5.2.2, this administration right scheme supports a modeling process where top-level management can set up lower level organizational entities, and from there even more lower level entities can be defined in self-organization. The following example illustrates this:

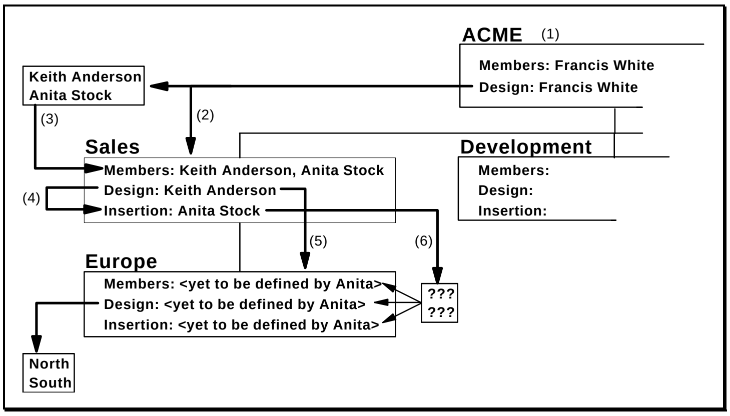


Figure 5-10: Creation of sub-units and assignment of access rights

For the imaginary organization shown in Figure 5-10**,** Francis White has initially been defined as a member of the top-level organizational unit *ACME* (1). She is also equipped with design access to this entity, which also includes insertion and read access. Since she is ruling on her own, she has not explicitly assigned somebody to have insertion access. Francis then designs the new sub-unit *Sales* (2) and assigns Keith Anderson and Anita Stock as members (3). Keith is given design access, who in turn nominates Anita to select and insert employees (4). Keith then implements yet another sub-unit *Europe* (5) while Anita is allowed to assign employees to positions or roles within her own unit and within the sub-unit (6). In step (6) Anita could also assign or reassign employees within her own organizational unit. For simplicity reasons, the default read access as depicted in Figure 5-9 has remained untouched in this example. The designer of *Europe* can then define two new sub-units *Western* and *Eastern* and again administration rights can be granted to selected members (7).

Using these restricted design and insertion access rights, an uncontrolled change of an organization's structure can be prevented, while at the same time extensive rights for self-organization can be given to the employees. Persons without such access rights have to address their colleagues who are responsible for the design process and persons who have been given these rights can contribute immensely to the design. The more people in an organization are given access to the design, the more participative the process can become and the more adaptive the organization can be.

Again, the specified administration rights are available for every entity type with the GEIMM; however, for hierarchically structured entities, a default process was defined to automatically assign default read access (Figure 5-9). A default description can be overwritten on the particular entity definition level.

After a specification of these administration rights, every intended action of an employee in the organization's model is checked against the current access rights and is denied if the access rights are not sufficient. For example, before opening an entity's description in the enterprise's repository, the read access needs to be checked. In the same way insertion access for job-shifting or designer access for intended structural modifications needs to be scrutinized. Since GroupOrga provides various end user front-ends to the organizational model (such as the database front-end, the graphical modeling tools or the Java-based modeling tool for Web access), each of these tools is based on the administration rights. Hence, none of these tools allows access to data which is not intended for a particular person's access. In the graphical GroupOrga modeling tool, for example, the entities with restricted access rights (read access only) appear in a gray box with gray text, while other, accessible entities are in color with fully readable text. The entities which are of no importance to the user, and to which the user has no access, are not shown in the modeling tool.

5.2.5 A Variety of Supportive Tools for Platform Independent Modeling

Although an organization design process is successful only, if all affected parties can take part in the design, the following list shows that this is not always possible due to technical impediments:

- High royalties for some organization design software prevents companies from buying several licenses of these tools and thus hinders enterprise-wide availability.
- Access to the enterprise models and their modification often necessitates highly specialized software, and thus accordingly qualified and entitled employees. High costs for training prohibit a sufficient number of employees from qualifying.
- An enterprise-wide provision with the organizational models often fails because of improper data storage. Often these specifications cannot be exchanged due to a lack of standardized interfaces. Instead, the models are hidden in complex database management systems.
- Largely distributed organizations (have to) support a large number of unlike operating systems, desktop software, and end user applications. Moreover,

organizations have a large variety of hardware platforms collected over years, and these still have to be maintained.

- Very large organizations or trusts use various organization design systems. These may have disintegrated organizational databases or incompatible graphical design tools. This situation is defeating any efforts of setting up a distributed design process across the partner's borders.

In order to reap the advantages of participative and distributed organization design, these hardware- and software-born restrictions must be eliminated. Platform-independent modeling is the long-term aim of GroupOrga. Sections 5.2.5.2 and 5.2.5.3 explain in greater detail how these technical restrictions were solved in GroupOrga. They introduce a combination of the underlying groupware technology and WWW technologies.

Before going into architectural details, section 5.2.5.1 throws a light onto the topic of platform independence from another angle: Different users' needs have to be fulfilled with various types of applications on different platforms. Chapter E in the additional technical documentation deepens the considerations on various user classes made here.

5.2.5.1 User Classes in an Organization

Because collaborative organization design includes different user types, participative organization design is carried out through different user classes in the organization. Their varying requirements result in a scale of possible user classes which are illustrated in Table 5-1. The first version of this GroupOrga scale of possible user classes was presented in [Ott/Nastansky 1998a] (p. 568). It shows that the target group of this organization design process are all members of the organization. In other words, it ranges from employees who only want to retrieve information to those who actively and regularly participate in the design.

"Push-button" information needs	**Occasional changes or adaptations**	**Regular departmental design and planning**	**Regular planning, analysis, reporting**
❑ Information "at the touch of a button" ❑ Administration of one's own organizational data	Sporadic adaptations and modifications to structural organization data	Regular adaptations and modifications to structural organization data across organizational segmentation	❑ Regular design, analysis, reporting ❑ Design from scratch
end user ←			→ administrator

Table 5-1: GroupOrga scale of varying requirements by different user type classes

The leftmost position on this GroupOrga scale is occupied by the end user, who needs information about the current organizational structure *at the touch of a button*. End users do not need to be equipped with highly specialized software, nor are they well trained in the use of organization design software. Rather, their task is to simply maintain their own organizational information (knowledge/skill, group membership). Moreover, often users in this class are engaged with tasks that do not require well-equipped hardware at the person's desktop. The discussion on the Network Computer (NC) would find a good example in this kind of user. Hence, the technological prerequisites are rather low-level, and network driven software needs to be considered here.

The second user type on the scale comprises the organizational members who sporadically perform adaptations in their immediate organizational surroundings. Examples are employees in the operational bottom level management, workgroup leaders, project group managers, and heads of smaller organizational units: For their everyday work, users in this class have mid-sized computer technology (high-end NCs or standard PCs), to support their level of organization design.

The third class of users regularly performs modifications to the organizational structure. Members include middle management and the human resources department. Due to these employees' integration in complex business processes which may require heavy computation and heavy use of computer-based desktop applications, these users require high-end computer technology. This technology is capable of servicing dedicated organization design applications and specialized software.

The last category of user types describes the employees who are engaged with strategic management and carry out large-scale design from scratch or redesign. This class also includes employees who hold "organizer" positions. Besides them, employees of the human resources department fall into this class, as well. These users require high-end technology at the desktop, and they are equipped with even more specialized software for structural analysis, simulation and other purposes.

Each of these user types and their task descriptions imply different requirements regarding the necessary toolset and the functionality needed to perform the respective tasks in the organization design process. With the various GroupOrga tools, adequate functionality can be offered respectively. The next two sections examine these tools and their purposes more closely.

5.2.5.2 Platform Requirements and Implementations

Important arguments for implementing the GroupOrga project on top of the chosen groupware platform have been its necessity of distribution technology (e.g. replication), high security standards, distributed database architectures, and the fact that many WfMS are also groupware applications. While these reasons remain valid, the following arguments present an extension—not a replacement—to it. The extension is a list of concepts and implementations in the project to ensure platform independence, and thus, a solution to the identified technological problems (Figure 5-11). In addition to highly specialized applications such as the graphical organization modeler to be introduced later, several GroupOrga

- *Organization design by all organizational members*
- *Short preparation-time*
- *Low servicing expenditures*
- *Independence of a particular workplace*

Figure 5-11: Distribution requirements in GroupOrga affecting platform- centered topics application areas have also been covered with WWW and Java technology which brings about applications that run on virtually every hardware and operating system. Thus, the extremely important platform independence is reached.

Organization design by all organizational members

The participation in the design process of every organizational member implies that this task can be fulfilled from every computerized workplace. However, most existing organizations have heterogeneous hardware technology and operating systems, as the introduction to section 5.2.5 suggested. This aspect becomes even more valid in scenarios of virtual organizations with cooperating partners. While some partners may have internal standards for their organization design technology, this cannot be assumed for all parts of such a trust.

Java technology is a cut above other technologies of this kind because of its ability to produce programs for every platform (*write once, run anywhere*). With this technology as a source for GroupOrga applications:

- It is not necessary to recompile the software developed for all supported or possible system platforms. Hence, no knowledge about the respective hardware platforms is required to translate the source code.
- The software no longer needs to be distributed in dedicated versions of the respective platforms. In addition, it is no longer required to explicitly establish distribution channels for the software.

For platform-independent GroupOrga applications, the source code is translated into byte code, which is executable on every platform that supports the Java Virtual Machine (JVM). The existence of JVM together with the GroupOrga applications reverses the present situation: The system environment is adapted to a uniform software, rather than adapting a software package to every platform.

When a new platform is invented, a new version of an operating system is released, or in case a new partner with its own hardware joins the organization design environment, the GroupOrga tools can run on it immediately. Thus, the tools are future-oriented.

For users who fall into the user classes of *"Push-button" information needs* and *Occasional changes or adaptations* (see Table 5-1), yet another advantage can be gained with this form of software implementation. The GroupOrga applications can be run as Java-applets in Web browsers. In the field of end user applications, the Internet, in combination with Web browsers as front-end applications, serves as the largest conforming basis for software applications in order to reach a large and multi-layered group of users.

Short preparation-time

A substantial goal for today's organizations is to be able to react quickly to changing environmental circumstances. An application environment for the design and modification of organizational infrastructures should also be ready to use after a short preparation time. From an IT point of view, this implies that the software is easy to install and the tools are simple enough to render training unnecessary.

Because some of the GroupOrga tools are based on Java technology they can be used in a Web browser. Hence, when installed on a WWW server, the effort for the end user to deploy the organization design tools is comparably low. The only task left to do is to invoke the modeling tool via its address on the server, which may be accessible via the organization's homepage. The modeling tool can be downloaded and used with easy.

Low servicing expenditures

Users of the first two classes do not use their modeling software very often. Servicing, in terms of software packages, generally means to periodically install new versions and distribute bug fixes and patches. Installing new software on many unlike computers in an organization is time-consuming and expensive. Thus, for these classes, there are few benefits from installing new software.

With GroupOrga's WWW and Java-based tools for organization design, the cost of their installation is drastically reduced. Their architecture allows developers to immediately distribute any changes undertaken to the end user, no matter how often this user actually employs the application, how many users are affected and what the cost/benefit situation would be in the different cases. This is a breakthrough in bottom-level design participation, since every user on every platform can be equipped with modeling tools.

Independence from a particular workplace

Another characteristic of dynamic organizations is that design teams are formed for a short period of time. Employees in project-oriented organizational forms do not have a stable working environment over a long time. Instead, their workplace may change within the borders of their location, and a project-driven change to a completely different location may be unavoidable. In connection with further principles, such as teleworking and mobile environments, it becomes important to allow user-specific organization design from every workplace. The platform-independent GroupOrga tools offer this independence from any workplace, since GroupOrga does not require any specific computer configuration.

5.2.5.3 Varying Types of Organization Design Applications

Depending on the background of an organization, each organization may have some sort of organization design environment, organizational database or enterprise repository, or graphical design tools. They are often centralized systems, administered by a few organizational experts, and they very likely correspond to the other characteristics that were identified in section 3.4.

However, often these legacy systems contain large amounts of infrastructure or personnel data that may be useful for the design process as envisioned in this project. Often, organizations cannot easily get rid of these old systems to implement an application framework for participative, distributed and computer-based organization design. Because of this, the GroupOrga framework provides the ability to be integrated with existing organization design environments, with simple organization databases or with large enterprise wide repositories.

In contrast to the Java solution presented in the preceding section, this form of integration aims at the power end users and administrators who occupy the right side of the scale shown in Table 5-1. While previously the potential end user was sufficiently equipped with standard browsing and retrieval functionality, in this case there is a justified need for highly specialized and proprietary tools. Due to this fact, GroupOrga recognizes that it is essential to retain the existing toolsets and to integrate them with GroupOrga. Moreover, operational workflow systems or office management environments may be based on the existing database structures, making them essential for the future.

This aspect brings about yet another quality of platform independence for the GroupOrga framework: interoperability and interchangeability with other systems. In GroupOrga, this means importing the data structures of other applications, modifying, and exporting them. At this point, the importance of a standardized but yet flexible enterprise model such as the GEIMM becomes plainly noticeable. Without a comprehensive model it would not be possible to take over an organization's model into GroupOrga structures. However, with the GEIMM, it is possible to convert many other forms, such as role-based models, competence models, purely hierarchical models, personal models, and so on.

For GroupOrga, a simple but highly efficient solution is proposed. As shown in Figure 5-12, the interface software GroupOrga Connector synchronizes structural data between different kinds of infrastructure models which reside in different applications for organization design.

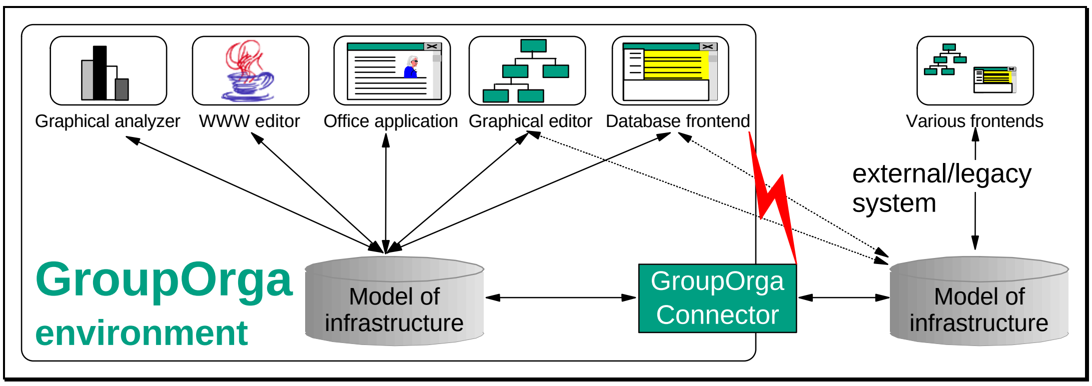


Figure 5-12: Positioning of the interface software "GroupOrga Connector"

According to the goal that interface software should have a generic basis which allows for an integration of various applications, an extension to the architecture, which provides a configurable import and export functionality, was developed. The interface is thus flexibly adaptable to other organizational infrastructure repositories. The dotted double-headed arrows in Figure 5-12 indicate another alternative. Each administration tool in the GroupOrga environment were adjusted to yet another external system. Due to the unpredictable programming effort and to the inflexibility of such an attempt, it was not a possible solution. When using GroupOrga Connector, the GroupOrga tools understand a given and unchanging set of import and export commands. Hence, adaptations for newly integrated external systems can be made outside the complex tools in GroupOrga Connector.

In the course of this project, two concrete organization design systems were integrated with GroupOrga Connector. This integration concerns OIS (from Siemens Nixdorf Informationssysteme AG) and BONAPART (from UBIS GmbH). Both forms of integration are tackled from a technical standpoint later in this chapter.

5.2.6 Skepticism Arising with Participation in Distributed Environments

It was discussed that the building of an enterprise knowledge base, such as the GroupOrga organization repository, requires a team of potential *knowledge engineers*, that is, basically everybody in the organization who has some expertise in the field of organization design. Such a team can accelerate the structural design process. This is important, since otherwise there is a danger that the resulting organizational repository will be obsolete even before it is completed.

However, skeptics cite several problems that arise from having a team of *knowledge engineers* in a distributed environment:

Danger of multiple definitions of organizational entities

Sometimes the same entity is defined differently by several members of the organization, and sometimes different names may be used for the same entity. For example, the organizational members may all want to design the entity *person*, but use different names (employee, colleague, member, or secretary).

To cope with this problem, it is necessary to have a well-organized information exchange between the members of the design team about the meaning of the GEIMM and its entities. The GroupOrga organization repository provides a template structure for the definition of an organization. With this predefined structure and an organizational structure library, it is clear which entities exist and how they are in relation with other entities. Likewise, the graphical GroupOrga modeling tools guide the inexperienced user to a well-defined and understandable organizational model, which is free of overlapping and misunderstanding. GroupOrga keeps a list of synonyms of terms which appear during the modeling process and a list of other terms. This list is somewhat similar to the content of chapter 4.

In addition, the distributed design approach is a cure to this problem in itself. Since every member of the organization is entitled to see and search the whole organizational repository, every potential designer can check which entities have already been modeled, which entities do not yet exist, and how the own modeling fits into the existing structure.

Omission of entities in the overall domain

In contrast to the multiple definition of entities, it may also happen that some important entities of the organizational structure are forgotten.

With GroupOrga, there is a clear distribution of responsibilities among the team members. Different areas of modeling are automatically assigned to different employees in an organization by means of the administration rights allocated in the top-down modeling approach (see sections 5.2.2 and 5.2.4). Although is may sometimes difficult because there are no clear boundaries between the areas which are to be represented, the underlying groupware platform facilitates this coordination process between the affected parties without the need of a central coordinator.

On the other hand, the omission of entities appears to be more of a problem in centralized modeling settings than in distributed ones. In a centralized modeling setting, the single organizer does not know about the peculiarities of a particular workgroup or about holders of certain roles. In a distributed setting like GroupOrga, the users know these attributes because they *live and work* in the structure to be represented. Thus, these entities are more likely to be remembered.

Different arrangements of organizational structure

Different members of the design team may model their respective surroundings differently. For instance, some members describe the knowledge about a workgroup as one entity. Others spread their description of the same workgroup over several entities.

A method to achieve a common structuring is to agree on a skeleton of the structure (taxonomy) before a detailed modeling starts. The GEIMM predefines such a taxonomy clearly by not leaving many choices on the entity level. This is not to say that the GEIMM restricts the user in the modeling process. But when a choice is made to use a subset of GEIMM entities (*workgroup*, *role*, and *knowledge/skill*), the given entity structure of GEIMM does not leave too wide a margin for employing these entity types for a concrete model. However, what taxonomy and what degree of granularity should be chosen for the design process needs to be discussed by the design team beforehand and individually.

Diverging opinions about the same entity

Organizational members may have different opinions on the same entity. This may result in different descriptions of these entities. The following example shows how two organizational members see the entity *person*. Although the two team members refer to the same entity, the descriptions contain different attributes.

```
person:                          person:
     name                             first name
     office                           last name
     phone-number                     birthday
     age                              institute
     company                          education
                                      project
```

In GEIMM there is a regulation that eliminates the need to spend time on such discussion. Its standardized definition of all entity types in the organizational model prescribes entities and their attributes clearly and still leaves enough room to flexibly combine them for a specific organizational model.

Diverging knowledge representation formalisms

Concerning the use and preference of knowledge representation formalisms, great discrepancies may exist between organizational members. This topic of representational mismatch was covered in section 4.4, where it was solved by abstracting organizational reality into the generic enterprise model GEIMM. With it the members of the distributed design team no longer have the problem of determining what formalisms should be used to model a specific piece of the organization. For the participating end user, there is no longer the need to study and compare several representation formalisms according to their adequacy since the hybrid enterprise knowledge base uses a net-based approach for the processes, a hierarchical and networked approach for the infrastructure, and an object oriented approach for the information model. This differentiation does not refer to the form of data storage, but it rather describes the semantics to be used in the three partial models.

Varying depth of modeling

Between the organizational members, great variety may occur in the granularity of the structural knowledge description. While some prefer a scarce description for *their* infrastructure, others use many entities for the description. The latter may also include common sense information for the representation of entities in the model.

The representation of a middle-level organizational unit as a scarce description of organizational sub-structure is not sufficient. There needs to be further specification about which sub-units exist and what persons are engaged in this part of the organization. On the other hand, in a participative process, the degree of specification is up to the concerned organizational entity. If the users decide that a very scarce representation is adequate for their internal work, there is no need to interfere in their personal modeling approach. Which depth has to be chosen can only partly be defined by the requirements neighboring departments or workgroups have when interacting with the organizational entity. Anything else is up to the organizational entity in question.

In GroupOrga, common sense information is the kind of information that is not contained explicitly in the GEIMM, but which helps the user to understand the structural entity. In addition to the concrete entities (*person*, *role*, *workgroup*) and their attributes, additional comments and information also belong to the infrastructure model. Such additional information can be stored with every entity type in GEIMM from either the database front-end or from the graphical modeling tools. It can then be used in any form to further describe the entity.

Lack of decentralization of technological and hardware expertise

An operational factor against decentralization and participation in organization design is the difficulty of communicating with people and coordinating them at another location. An organizational repository may fail to perform tasks as needed for a variety of reasons. If the effectiveness of the organization depends on getting the system up and running as soon as possible, the logic of having the organizational application where it can be serviced most easily is compelling. If the technological experts cannot be decentralized, then it may be risky to decentralize the application.

However, with the GroupOrga framework, the high cost of expertise in using the organization design system is invalidated. Due to the simplicity of the system at the end user's side and its tools, more people who are able to make use of the applications, become available and thus each organizational entity will have its own *technicians*. It should be noted that the standardization and simplification of these applications in GroupOrga contribute to the repeal of this counterargument, as much as the education of more employees for such tasks does.

Taking these aspects into account, chapter 6 proposes a methodology for the collaborative construction of an enterprise knowledge base. It makes the distribution of labor efficient, and minimizes the amount of overhead for coordination. The design process is modeled as a cycle with several phases.

5.3 An Enterprise Knowledge Base for WfMS

Section 4.4 briefly introduced the GroupOrga EKB and positioned it as the container for concrete enterprise models based on GEIMM. Here, the EKB is further described, mainly with a concrete view onto the GroupOrga framework and its requirements.

The overall aim of the GroupOrga EKB is the representation of knowledge about organizational structures which are relevant for the support of communication and workflow. Three main reasons advocate the development of the GroupOrga EKB:

Provision of organizational information to WfM and office systems

Provision of organizational context information: The GroupOrga EKB provides information for cooperative applications and as answers to questions of the task performers in workflows (`<orgunit1>_is_superior_to_<orgunit2>`, or `manager_of_<workgroup>`). Thus, every application which requires such information and does not require its own information base to be developed and managed.

Distributed provision of directory information: Communication requires information about how the cooperating partners can be reached. With GroupOrga EKB an integrated and distributed directory is provided to applications and users.

Integration of standardized external resources: GroupOrga EKB provides integrated access to the X.500 world. X.500 is a standard for an electronic address book and is further examined in section 5.3.4.

Flexible and quick reaction at all times

Visible and user adaptable data model: A flexible data model was integrated in the EKB that allows the system to quickly and easily adapt to various organizations. This was required since organizations change and with most existing environments it is impossible to find a fitting representation based on their rigid data model.

Quick retrieval from the data model: Simultaneously with the conventional organizational hierarchy, the GroupOrga EKB allows for other forms of structuring, such as projects, workgroups, and teams. This results in multi-dimensional profiles of employees in an organization. In other words, they fulfill more than one function or task. With the EKB, it is relatively simple to quickly find the responsible employee in the organization, no matter how complicated the connection between workflow and task performer may be. Example: Assign the task to the secretary of the manager of the superordinated organizational unit to the unit, of which the initiator of this workflow is a member.

```
Initiator_of_<workflow>
     ↳ organizational_unit_ of_<person>
          ↳ superordinated_organizational_unit_of<org_unit>
               ↳ manager_of_<org_unit>
                    ↳ secretary_of_<person>
```

Figure 5-13: Complex determination of a task performer

With the EKB, this complex determination is assisted through a browsing mode, which makes the difficult definition of such retrieval formula superfluous.

Support of distributed modeling

Support for scalability: It is of particular importance to the EKB to aggregate organizational information in large geographically distributed organizations from the participants themselves. Therefore, it provides scalability as a crucial issue. From the administrational viewpoint, it is easy to extend the number of users or the number of locations where the EKB is used. Since it is based on an underlying distributed service environment, it is a fundamental component in cooperation settings. The market research studies in chapters 3 and 4 have shown a lack of such functionality in existing knowledge bases.

Facilitating the new competitive environment: More and more firms are allowing their employees to make their services available to other projects, workgroups, organizational units or even to other firms on a contractual basis. With its distributed modeling approach, the GroupOrga EKB facilitates this development strongly. The EKB contains information about a competitive network of properly and sufficiently qualified partners. Krulwich [1997] puts this as the development from the good old boy network to "The Good Ol' Bot Network", in which he positions the Ol'(d) roBot (the computer) as the information source that knows who and where the experts are.

5.3.1 Design Criteria for an Enterprise Knowledge Base

For the design of the GroupOrga EKB, criteria such as central accessibility at all locations, simple and easy adaptability, an adequate intuitiveness, clear structuring, aspects of data protection and security, performance and multi-user usability were defined. Tracking the history of an organization's development has not yet been taken into consideration. The EKB has been implemented as the *main knowledge base* for various application systems and end users. *Simple and easy adaptability* implies a flexible presentation of the information with the effect of easy maintenance and servicing. The criteria of *adequate intuitiveness* means that the structural knowledge saved in the organizational database and the rules belonging to it is immediately intelligible and clear to the end user. In close connection with simple adaptability and adequate intuitiveness stands the criteria of *clear structuring* of such a knowledge base: Structural and procedural knowledge are distinctly separated and are represented in different ways. *Data protection and security* relate to the fact that information stored in the EKB is highly confidential in an organization and is thus secured accordingly from unauthorized users. Under the criteria *performance*, various aspects are summarized: fast reaction times to queries through a simple query language, short replication cycles due to compressed information storage, and quick responses to application queries by means of an effective interface. *Multi-user usability* comes along with the requirement of a distributed modeling approach and indicates that a logical coherence among the entities of an organizational model in the EKB is ensured at all times.

Interfaces to GroupOrga EKB

The two most important *interfaces* between the EKB and related application systems are those for workflow and office management and for personnel administration. Another interface may be that of training applications. Standard training that is scheduled for employees in certain positions are matched with concrete training provided by the organization.

Personnel administration:

Data about an organization's employees is considered highly sensitive and is thus subject to specific data security regulations. Therefore, most of the employee data is kept away from the project managers and from the operative positions in the organization.

Up-to-date employee data which is relevant for organization design can be manually copied from the employee database into the EKB. However, this procedure has problems with consistency. With each such transfer of employee data into the organizational database, the validity of the integrity rules defined in the GEIMM is checked. Such an integrity violation occurs for example, if newly hired employees are in the employee database before their roles are defined in relation to other entities in the organizational model. For instance, this may be true for a 1:1 relation between *position* and *person*. In contrast, the relations *workgroup* to *person* or *authorization* to *person* may be lacking. In addition, for evaluations and analysis, it is compulsory to assign *knowledge/skill* to a *person*, since an employee without relevant qualifications is unthinkable in a real world organization.

Another solution was implemented for the GroupOrga EKB. Simple, yet not fully satisfying is the creation of external transfer data whenever changes are made in the employee data that affects the infrastructure information. This in turn requires a coordination of the frequency in which the two data are accessed. In this case, it must be determined how often the employee data is made topical and how current the infrastructure information needs to be. A better solution presents complete or partial integration of the data into the enterprise knowledge base. Employee personnel is maintained by the human resources department, while other infrastructure information is updated by the distributed participants as demonstrated before. Advocates this solution since its distributed setup naturally makes the solution the best to implement.

Computer supported processes/Workflow management:

As far as manual processes in an organization are concerned, the GroupOrga EKB serves as a pure information base for the employees. Only a user interface that provides access to the structural information needs to be defined. This is different with queries from applications.

In practice, there is no homogeneous architecture for workflow processing systems. According to the age of the application, its origin (individually programmed in-house or external, standard software from the shelf) and the history of its use in the organization, different forms of query formulation and access to external data will exist. Similar to the integration of human resource applications, the creation of external transfer data is a choice. The EKB reads and then evaluates such queries via its importing functionality. After the evaluation, a returning external transfer data is produced and is then imported from the processing application. In addition, the GroupOrga EKB provides access via an application programming interface. Via this interface, ideally (at least with younger applications), an application can directly be connected to the EKB and based on standardized queries (section 5.3.4).

Time factor and versioning in organizational databases

It goes without saying that an enterprise repository stores data about the current organizational situation in order to equip users and applications with up-to-date information. But existing directory systems and organizational knowledge bases have no notion of time. Thus, recording dynamic changes of the organizational situation, which may be the consequences of actions of the participants, is not facilitated.

In addition to the current structure, the planning and simulation of possible future structures makes sense on the corporate level and the bottom level. This planning can be temporal in nature, but the storage of accepted, future structures (*valid from ...*) can also be useful. Hence, there is a need to clearly discriminate between such temporal and future organizational structures. In the latter case, a transition from one version to another must be consistently realized. For instance, no existing *position* should be unrelated without intention in the new form.

Moreover, organizational structures should be analyzable at a certain point and period of time, such as "What structure was valid at point of time t_1?", "Who was the substitute of person A in period t_1 to t_2?". Such investigations require a time factor and a versioning of entity descriptions in the organizational repository. However, this may result in a deficiency in performance since depending on the frequency of changes, many versions must be sustained. This, in turn, contradicts the performance design criteria mentioned above.

While the first aspect of a time factor was conceptualized for the GroupOrga EKB, the past of an organization(al model) and its analysis was realized in the prototype in a rudimentary form. In this case, versions of each entity description are stored with:

- the name of the person who performed this last modification
- a date/time stamp of when this past modification took place
- the data (attribute names and values) as it existed in the latest version
- a connection to the currently valid entity description in the EKB

In order to cope with the performance problem, the user can decide which period of the past is maintained. In order to maintain consistency in the older versions of the infrastructure model, the specification of this cut-off date cannot be made distributively. It must adhere to a centrally reached decision and synchronization. Otherwise some versions may have been deleted at some locations, while other version were not deleted at other locations. By means of periodical reorganization, historical documents can be either transferred to an archive EKB or deleted.

5.3.2 Provision of Awareness about Organizational Changes

Awareness of individual and group activities was identified as critical to successful collaboration in chapter 3. It is supported in the GroupOrga EKB by an active, information generating mechanism which is separate from the shared organization database. Section 5.3.2 presents the message distribution model that provides mechanisms for increasing awareness about organizational changes between the users.

The mechanism in the EKB is based on a context dependent notification of users about modifications to organizational entities which directly or indirectly concern their work environment. This mechanism realizes and extends the initial ideas for provision of awareness in cooperation support systems presented in [Dourish/Bellotti 1992], [Bowers 1994], [Prinz 1996] and [Fuchs et al. 1996]. General requirements about the message distribution mechanism are presented in the following, followed by the outline of the algorithm.

Notifications about changes in the organizational repository are distributed in the context of the organizational structure for which they are relevant. For example, the notification about a new or modified entity in the structural model should be distributed to the people who are connected with that particular entity, for example, members of that workgroup, and players of that role. It is useless to distribute information about such action by user-produced global broadcasts with very general content. In order to release users from the explicit production of such broadcast messages whenever a change is made, the EKB supports an automatic generation of a message. Its generation is caused by the processing (storing) of an entity description as well as its distribution in the appropriate context (the organizational boundary).

Information about organizational entities can be differently important for different working contexts and these entities can have different significance in each. For example, an entity description can be a major element in one work setting (for example, the *workgroup* 'project leaders' being very relevant to the members of the project) and the same entity can be a background element in another context (for example, the organization's CEO). Thus, information about modifications to the entity must be distributed differently in both contexts based on the different significance. For this reason it is possible to modify (summarize or reduce) the information content of the message. For example, the detailed information about a modification that might be delivered directly to the involved people is reduced to a short notification with a link when it is delivered to an associated or cooperating workgroup. This case may occur when a workgroup consists of smaller sub-workgroups. Then the subgroup members may want to be informed in detail about modifications in their group's structure, while they might only be notified shortly about actions happening in other groups.

A set of implementation requirements were derived from these considerations.

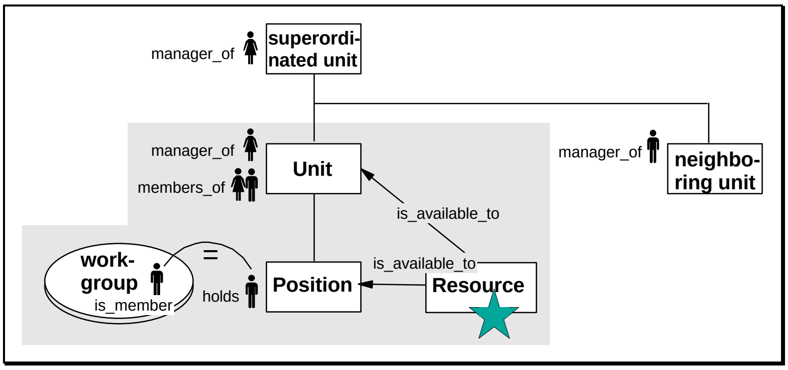


Figure 5-14: Example for the awareness-context of a modification

First, a specification of the work context and organizational boundary of each person is found. This is easily achieved with the GEIMM. Organizational characteristics and the set of people who are directly or indirectly connected with these characteristics are represented by organizational entities. The connection that exists between the structural entities and the persons to be notified is described by the relationships that link the entity description, such as a *workgroup*, *organizational unit*, and *knowledge/skill* group to the relevant persons. Besides these mere membership relations, other relations, such as subordination, role-playing, or position-holding, that exist between the organizational entities can be taken into account when determining whom to make aware of a modification in the model. The administration responsibilities (read, insertion and design access) in the GroupOrga EKB which are assigned to various persons are also a source of such automatic decision making. Figure 5-14 presents an example of such a net of connections from which to determine whom to notify.

The gray shaded area indicates the people who are directly affected by a modification in a *resource* description. Through its explicit `is_available_to` relation, this change affects the manager and the members of the related organizational unit and possibly a position within the organization this resource may directly be assigned to. While the managers of the superordinated and neighboring units may be informed, their members need not to know what actions have been taken. However, for a workgroup, in which the affected person is member, this change may be quite important. Such a workgroup is indicated to the left of Figure 5-14.

Secondly, the type of relationship by which an entity is bound to another entity to be informed determines how often messages are distributed. For example, if an entity is in direct relation (such as the *resource* to *unit* and *position*), then the persons concerned with the related entity are notified of all changes. But, if the entity is just in the background of other entities (such as the neighboring units), then only sporadically new versions of the entity may need to be notified. Thus, message distribution depends on the type of direct or indirect relationship that links one entity to another. The relationships can be regarded as the medium by which the messages flow within a working context.

Since relationships spanning various levels of hierarchy or various forms of grouping and sub-grouping may exist, such relationships can be interpreted as the representation of a boundary of interest for a modification. In other words, the further away another entity is from a modified entity, the less important is it to make this remote entity aware of a modification.

Concretely this means that messages do not need to be transported indefinitely, more likely they should be deleted after they are passed repeatedly from one working context to another. Assuming that an organizational model based on GEIMM includes the definition of a large set of organizational entities and relationships, a determination of when a message has to be passed on and when to be deleted appears to be difficult. However, since with the GEIMM an entity-relationship model is applied to the modeling of organizational relationships, this problem can be solved with computation. Following all relations between concrete entities in an organizational model with an algorithmic approach, the degree of relationship between two entities can be determined. Based on this information it can later be decided whether or not the event should be distributed further (for example, from a unit to its superordinated unit).

A possible distribution algorithm which incorporates the considerations presented above would run as follows (see [Prinz 1996]):

1) Modifications to an organizational entity trigger a message carrying the following information:
   - The name of the entity that was modified
   - The data (attribute name and new value) that was changed
   - A date/time stamp
   - The name of the person who performed the modification
   - A link to the original entity description in the EKB
2) The message is delivered to all directly connected entities which are determined via the relationships they have with the entity. If there is more than one related entity, the message is copied.
3) If the receiving entity is a person, then the user is notified about that event.
4) At each informed entity, it is decided whether the message needs to be further distributed. This decision is based on whether there are further entities in relation with this entity and on the distance of the message to its origin. Consequently, when the message has reached a distance to its origin which is equal or greater than its threshold, it is deleted. Otherwise the message is propagated to connected entities. The distribution continues with step 3.

With this model, information about organizational modifications is automatically distributed among those concerned with the participative organization design, both in the same working context and in other distributed locations. Since the distribution of messages is stopped after a predefined number of relations is traversed, this algorithm allows the modeling of distances between the place where the action occurred and where it is received. Currently, the awareness messaging functionality in the GroupOrga EKB is implemented with a threshold of *one*, as is shown in section 5.5. Further refinements are possible by additional tuning, if necessary. An advantage is that it meshes with the presented GEIMM and no further modeling techniques and other actions are necessary.

5.3.3 The Enterprise Knowledge Base as Organizational Library

Organizational structures that were designed as an organizational model can be used as a pattern for the planning and installing of new organizational structures. Therefore, the GroupOrga EKB can be used as *a repository of the past* which collects examples of how different parts of an organization (or different organizations in general) set up similar functions. The result is an on-line organizational library. Employees interested in improving organizational structures can consult it to find alternatives for laying out particular parts of an organization, along with experiences and guidelines about which alternatives work best in which situation.

As an organizational library, the GroupOrga EKB is intended primarily for three uses:

- **Inventing new organizational (sub-)structures:** An important goal of a library is to provide a structure representation database that helps to make systematic, empirical suggestions about new organizational structures.
- **Redesigning existing organizations rapidly and effectively:** In the context of today's *reengineering* and *total quality management* efforts, users of the organizational library include: designers (either internal or external) who take part in redesigning their organization, managers who design the units and workgroups they supervise, and employees at all levels who design their own working environments. Note that no consultant or external *professional* is listed in this enumeration. The library is intended to serve those who work in the organization and who participate in the continuous design process.
- **Sharing ideas and *best practices* about organizational structures:** A library is a useful source for information and examples for people at many organizational levels—from business newcomers first learning about organizational functions of their employer, to experienced employees moving to a new job where they want to learn quickly *how things are done*. Because the library contains various organizational structures to compare with each other, it also provides a particularly valuable resource for collecting and analyzing *best practices*. However, the library does not assume that there is a single best way. Instead, it assumes that there may be a number of alternative structures, whose relative advantages under different circumstances need to be compared.

There are two main advantages for reusing organizational structures: Since the design is not starting from scratch and is based on predefined and (ideally) tested concepts and architectures, the expenditures for designing the organizational structures decrease. In addition, the quality of the design process and the resulting model increases while the error rate decreases. This statement was proven true when the quality of the patterns provided improved during the process of continuous reuse and design.

For this approach to be feasible, it must be scaleable in editing and integrating information from the library into the current organizational structure. For instance, a top-down approach to design with the EKB library is to start with similar examples already in the library, create a new specialization, and then modify the specialization as needed to describe the given circumstances. An alternative bottom-up approach is to start by designing various descriptions of new distributed structures and then connecting them to existing structures from the organizational library that are generalizations of the whole process or its subactivities. In the course of adding this new specialization to existing structures, the existing patterns are modified and may serve as yet another pattern in the organizational library.

Generic building blocks may thus exist for each modeling level in an organization. A library may consist of these blocks for the enterprise and business process level. The intermediate level introduces middle-sized enterprise entities like larger organizational units reflecting the whole organization's function (production, R&D, marketing, and so on) and the overall structure of project teams to bring a new product into the market or to launch a marketing campaign. The bottom-level includes the generic building blocks like workgroup structures for accomplishing a singular task.

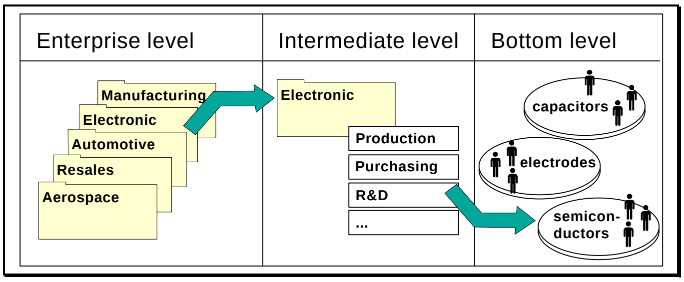


Figure 5-15: GroupOrga EKB generic building blocks

These generic building blocks can be applied for designing or redesigning the appropriate concrete parts of an infrastructure model for a specific enterprise. To support the building of an infrastructure at the enterprise level the building blocks may have the widest application, and build the framework for the intermediate level. At the intermediate level, the blocks are applied to more specific but still generic areas of an enterprise. It is an open set which is filled by standards and internal structuring.

As with every form of reuse, this concept has some problems. Whether a chosen building block fits directly or in an adapted form into the considered organization must be examined very closely. Errors during this selection procedure may be costly later. On the other hand, due to its continuous character, the design process allows for corrections at any time. Secondly, each organization should be treated as unique and thus generic principles should not be applied to it.

In the project, the most visible outcome of these considerations is a functionality within the graphical modeling tool for storing and manipulating structural designs. The GroupOrga EKB manages and stores the infrastructure descriptions, and the modeling tools display and edit selected building blocks. Section 5.5 shows implementation insights and examples.

5.3.4 Standardization of the Enterprise Knowledge Base with X.500/LDAP

Using standards is important for an enterprise repository, especially in a case where external partners with heterogeneous environments intend to participate in the design. Therefore, the partners can limit the number of inter-company formats they support in order to exchange the infrastructure information they need. With the use of standards, an organization can broaden its choice of future cooperating partners. Additionally, in the absence of a uniform data and communication standard for the organizational database, no single directory service can access the information that is contained in the database. Whether set by the government or by major computer vendors, standards are critical to the operation of the EKB. A way to standardize EKBs is to use the X.500 Directory standard. It is used in GroupOrga.

X.500 is a standard developed by the International Standards Organization (ISO) and International Telecommunication Union-Telecommunication Standardization Bureau (ITU-T, formerly International Telegraph and Telephone Consultative Committee, CCITT) for an electronic address book (see [ISO/IEC 1993a]). A simpler description of the standard can be found in [Chadwick 1994] and in [Martin 1996].

With its potentially world-wide distribution, its methods for distributed management, and its standardized service interface, the X.500 Directory standard fulfills the requirements for a distributed address directory and scalability. However, shortcomings arise when the directory is applied to office and workflow applications and when a more detailed modeling and administration of organizational information, especially in a distributed and participative form, is required. Major problems deal with the representation and modeling of non-hierarchical organizational entities and relationships and with replication of data. Nevertheless, during the GroupOrga project, X.500 proved to be a key to integration with other applications.

The standard describes two main purposes. The first purpose is to supply an application-independent management and information service about applications and their users. The second purpose is to serve as a global name server, that is, for the management of globally unique user-friendly names for objects of the *real world* that are represented by entries in directories. In this context *real world* is restricted to the world of telecommunication (see [ISO/IEC 1993a], 6.2).

Since the X.500 standard for directory and repository systems is a common topic in literature, this section does not have an extensive introduction. (A detailed description of the features of X.500 that are important in GroupOrga is given in chapter C in the technical documentation.) Rather, this section gives an analysis and summary of the X.500 directory's organizational modeling capabilities with respect to the GroupOrga requirements. This analysis is mainly based on an extensive investigation into the technology used during the project (see [Heinz 1997] and [Heinz/Ott 1997]).

Section 5.3.4.1 discusses the requirements concerning the content of the directory. Parts of the GroupOrga enterprise model are transferred into an X.500 conform object model following suggestions from [Prinz 1989a] and [Heinz 1997]. The subsequent sections deal with requirements concerning security and distribution.

5.3.4.1 Transforming the Infrastructure Model into an X.500 Object Model

The representation of real world objects as instances of freely definable object classes defined in the X.500 directory is well suited for the discrete representation of organizational entities. In order to examine whether the information model of X.500 is capable of representing an organization, the infrastructure model described in chapter 4 must be transformed into an X.500 object model. The entities defined there must be mapped onto the standard X.500 object classes, and if necessary, additional object classes must be created. A similar mapping must be performed with the attributes of the infrastructure model.

The structure suggested in [ISO/IEC 1993b] is used as the basis for this transformation. Relations specified in the GEIMM infrastructure model are reflected by attributes of the corresponding object classes. Some entities of the infrastructure model defined in GEIMM can directly be mapped onto X.500 object classes. However, for some of them it is necessary to specify new attribute types. The list of entities that are directly transferable, the corresponding object classes, and the customized attribute types are in Table 5-2. In order to reduce complexity, the custom attribute types are not specified in detail.

GEIMM Infrastructure Model Entity	Corresponding X.500 Object Class	Additional Attributes
person	organizationalPerson	
workgroup	groupOfNames	OrganizationalAttributeSet (defined in X.521)
organizational unit	organizationalUnit	
role	organizationalRole	requiredSkills connectedResponsibilities
hardware resource	device	maintenanceInterval manufacturer
software resource	applicationProcess	
software agent	applicationEntity	
location	locality	

Table 5-2: Selected GEIMM entities and corresponding X.500 object classes.

For entities that have no direct equivalent and are thus not listed in Table 5-2, new object classes must be defined. Examples of complete additional specifications according to the subschema definition [ISO/IEC 1993a] can be found in chapter B of this study in the technical documentation.

Problems occur, for instance, when transforming the entities *actor* and *resource*. They are virtual classes that are not used to create object instances but only serve as superior classes in the inheritance hierarchy. The standard X.521 object classes *organizationalPerson*, *groupOfNames*, *organizationalUnit* and *organizationalRole* could be modeled as subclasses of actor; *device* and *applicationProcess* would become subclasses of resource. However, this requires that the definitions of these standard classes are modified, which is forbidden by the standard recommendation. It only allows new classes to be subclasses of existing ones.

There are two possible solutions to this problem. Additional custom classes for *person*, *department*, *workgroup* and *role* are created, and both *actor* and the corresponding standard object classes of the X.500 standard are used as parent classes. The same method is applicable for *resource*. If it is preferred to use the standard classes instead of creating new ones, the virtual object classes *actor* and *resource* may be omitted. Their attributes and relations have to be assigned to the different subclasses.

The object model proposed here follows the second approach and would not comprise the virtual classes. The number of new object classes is smaller than in the first approach. This facilitates the integration of the model into a network of distributed directories.

The extended *directory information tree* (DIT, see chapter C in the technical documentation) structure is shown in Figure 5-16. The connections in the diagram do not express relations between the objects, but they reflect their hierarchical position in the DIT. For example, entries which represent *persons* are located below entries which represent *organizational units*.

The X.500 data model secures the concept for identification of organizational objects by hierarchical naming. The names constructed this way are, however, not very user-friendly on account of their complexity and the types of individual name components. With X.500-centered user interfaces, this would be quite difficult for end users and an additional, simpler name concept is necessary.

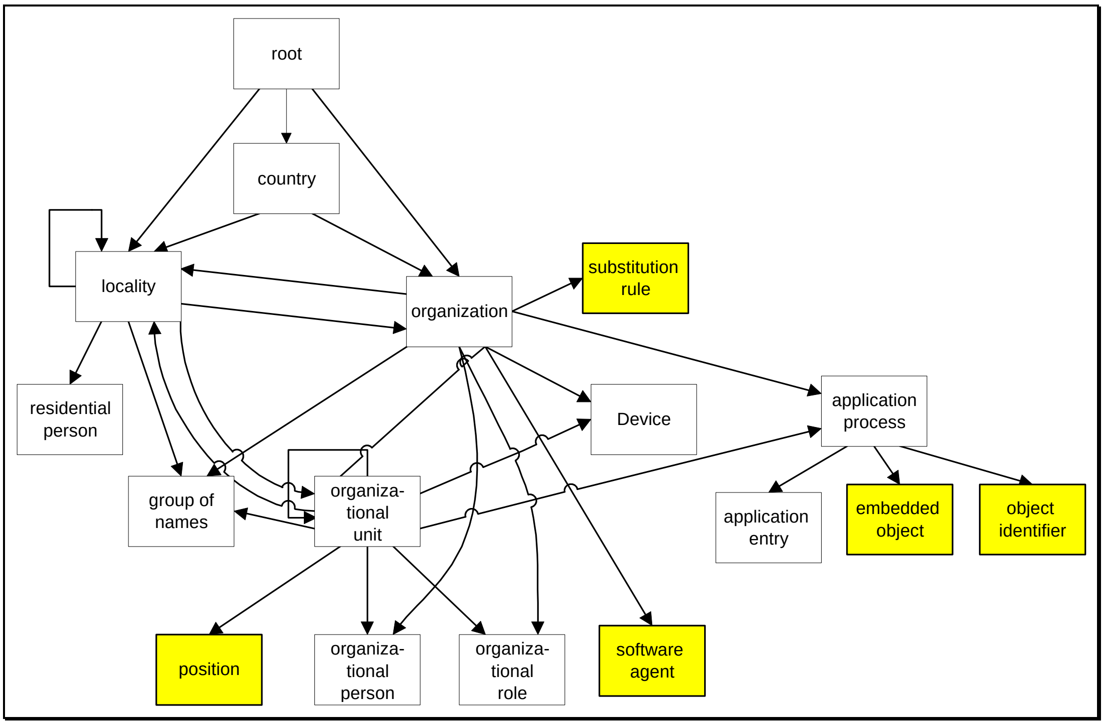


Figure 5-16: The extended DIT structure

Organizational information is mainly characterized by relations among real world objects. As Figure 5-16 illustrates, the selected representation method represents these relations by means of the DIT structure or by reference attributes. The DIT can be used to express *hierarchical* relations between entries which correspond to authority and administration relations. X.500 reference attributes are a general purpose to express relationships. Weaknesses are the missing concept of a relation as such and the restricted way to describe dynamic relations. This causes problems when the directory information undergoes frequent changes, as in the case of GroupOrga. Additionally, an improper use of directory intention and the fact that reference attributes are not administered by the directory but instead by the applications may result in inconsistencies.

In conclusion, the essential way for modeling organizational structures by means of the directory standard is that of the hierarchical DIT. This is very suitable for enterprises which exist primarily according to a hierarchical scheme. This might apply for most organizations if only the communication infrastructure (mailing addresses) is portrayed. But considering the listed requirements towards leaner and flatter organizations, more comprehensive organizational modeling becomes difficult with the X.500 directory standard.

5.3.4.2 Requirements Concerning Security

The security model described in the standard recommendation is very comprehensive and powerful. The authentication framework X.509 ([ISO/IEC 1993c]) is widely accepted and also partially integrated in modern network and groupware platforms.

All demands of a workflow management system concerning authentication are met. Storing passwords as well as public keys in the user's entry allows secure authentication to both directory information itself and external applications like office or workflow management systems. An X.500 directory integrated in a company's IT infrastructure eases password administration by enabling the storage and maintenance of the password for all involved applications in one central location.

The second important part of the security requirements are access control issues. They are also well covered by the X.500 standards. Access control is applicable on many different levels reaching from access to the directory down to access to selected values of an attribute. However, the implementation and administration of the X.500 security model may be quite complex. On the one hand, it is possible to grant groups access to selected objects instead of administering access rights person by person. On the other hand, access has to be granted explicitly to each layer down to the entry or attribute the user should access. This requires a well-designed security model for any organization.

5.3.4.3 Distribution Requirements

The distributed access to data as well as the possibility of assigning access rights allow different users at different sites to administer organizational information.

Since the X.500 standard is designed with a global directory in mind, emphasis was put on distribution facilities. The interaction among different servers to answer user requests is also important for workflow applications that operate in a distributed environment. But this model is also based on the hierarchical organization of the directory. Often, detailed administration authorizations cannot be described in a hierarchical (vertical) manner. It must be possible to describe them in a more finely tunable way.

For example, a member of the human resources department in an organization is responsible for the most recent data on training and seminars in the organization repository. These attributes appear in most *person_has_knowledge/skill* entries in the repository, but the DIT may be divided into two hierarchical domains. In this case, the domain managers would have to negotiate with the persons in human resources and vice versa. Thus, the dominating hierarchical view must be extended by a service oriented more towards participative design, which is contrasting the hierarchies.

In GroupOrga, the distributed management and maintenance of enterprise directory information plays a key role. X.500 supports the dissemination with the sophisticated concept of administrative areas. However, delegating the administration of structure management, access rights and collective attributes is difficult, and a flexible enterprise directory that adapts quickly to organizational changes is hampered.

The replication concept of the X.500 standard is not as mature as the one which the underlying groupware system of GroupOrga provides. Within the X.500 standards, replication is always performed between a master and a slave so that the information may only be updated at one location. GroupOrga's base-system treats all replica copies equally and allows modifications at all locations.

5.3.4.4 Alternatives for X.500 Integration

Sections 5.3.4.1 through 5.3.4.3 indicate that the concepts of X.500 fit only some requirements of enterprise repositories for workflow management and office systems. The X.500 information model is suitable to represent organizations which can easily be mapped into its hierarchical scheme. It may be extended to reflect custom requirements. But when considering a more comprehensive organizational modeling that includes the design of organizational relationships and loosely coupled entities, the X.500 standard shows major weaknesses. Thus, the requirements in chapter 3 are only partially satisfied.

The security features that X.500 provides are powerful. Parts of the X.509 authentication framework are already implemented in information systems (like Lotus Notes) and the defined access control mechanisms exceed the requirements.

On a large scale (organization wide), the distribution requirements are satisfied by the administrative authorization model. Particularly important in the context of GroupOrga is the potential to distribute the responsibility for administrating and maintaining directory information. Domains apply mainly to larger entities, such as whole organizations, large branches, and locations. On the smaller scale (the level of single *actors* or *workgroups*), the distributed management is not well supported, which in turn does not allow for widespread participation.

GroupOrga integration and extension

X.500 is weak in its organizational modeling capabilities. This is caused by the concept of a world-wide electronic communication directory. It does not provide a sufficient solution for an EKB as intended in this project. Following this conclusion, two alternatives for further integration of X.500 into the GroupOrga framework are possible:

- The first is to not consider X.500 any further. However, the importance of the standard and its future relevance imposes its consideration in the framework.
- The second is to fully integrate the concepts of X.500 and enhance the standard according to the requirements in chapter 3. This alternative permits the development of a directory that allows external access to an organization's repository and internal access for continuous and participative organization design.

In this project, the second alternative was chosen. It was found that when an X.500 directory was implemented into GroupOrga's EKB with justifiable effort, all the requirements were met.

With the information set specified in X.500, it is possible to completely integrate other organizational databases and connect to the world-wide directory services network. But in many cases, it is still not possible to satisfy the queries of application software. With the additional aid of autonomous software such as GroupOrga Connector, exporting structural data to and importing from any application can be achieved to the fullest extent. An organization model such as that in GEIMM serves as a master data set for synchronization. While the X.500 integration covers the part or the GroupOrga EKB, which deals with directory services, the additional GroupOrga Connector allows for the connection of external applications and their organizational databases to the GroupOrga toolset.

5.3.5 Query Language for Organization Databases

In EKBs, an ever growing and ever more sophisticated set of services and knowledge can be provided to the members of the organization. Permitting the large numbers of end users to find and use these services, which are often represented by their colleagues in other locations or projects, they must be easy to locate in the enterprise. A growing problem in computer assistance in databases, especially with large knowledge bases, involves identifying, locating, and accessing these services and resources and the people who offer them, and the application program which supports them. In some cases, the knowledge in an organization is underutilized simply because the end users are either unaware of its existence or unable to find it. Many of the difficulties arise from the lack of an adequate query facility.

Therefore, the GroupOrga EKB provides different types of search capabilities: searching based on primitive names and searching based on descriptions.

Primitive name searches in the EKB

A primitive name is understood as a character string that uniquely defines an entity in the knowledge base. Each primitive name refers to one and only one entity in the organizational model. Therefore, such a name can be used to either find a specific entity in the EKB or to determine that this specific entity does not exist. This search capability provides the same features as the "white pages" telephone directory, since the end user must know the unique name of the entity in the repository. Also, the GroupOrga EKB allows the end user to initiate a search based on a partial name. The result is a list of infrastructure entities that match the supplied partial name.

Descriptive name searches

Another type of search is one with descriptive names where an entity is identified by specifying information about the characteristics of the entity. A given descriptive name may thus refer to one entity, to various entities or to no entity. This is similar to the "yellow pages" telephone directory, because the user must only know some information about the required service or resource entity in order to initiate a search. An example of such a search may be "the secretary of the manager of the superordinated organizational unit to the unit *Sales*". Usually, the descriptive specification uniquely identifies an entity if enough characteristics are included in the search.

Descriptive name searches are potentially more powerful than primitive name searches. Moreover, for the purpose of participative design they are also more important, inasmuch as they are closer to organizational and procedural requirements, and thus easier to understand and formulate for end users. In order to provide a descriptive name search to the end users of the GroupOrga system, the GroupOrga EKB is equipped with a user-friendly organizational query language. Again, this section introduces the concepts, while succeeding sections show the implementation.

In order to address the larger segment of users of the EKB, that is, the end user who is new to the field of organizational databases, the idea of descriptive naming has been taken a step further with GroupOrga. Common sense queries in almost natural language queries are provided. This requires the enterprise repository to be able to deduce answers to questions that one would normally assume to be asked by an end user who has a *common sense* understanding of the enterprise. This does not denote knowledge of an expert, neither in the field of query formulation nor database management. The vocabulary of the language is adopted from the daily work environment and from the entities' names in the GEIMM.

The query language supports questions, for example, about administrative and functional characteristics of the organizational entities. It can ask for information about the members of an organization and it can retrieve holders of roles or knowledge/skill within the infrastructure. Such common sense queries are for example a query for geographical knowledge e.g. `Location Manager Unit ACME`, meaning "What is the location of the manager of the organizational unit ACME?". The answer could be 'Room 12c27'. A subset of this question, `Manager Unit ACME` returns the name of the manager of the organizational unit ACME. Queries about persons, such as `Person Role Hotline/Europe` ask for the responsible person who holds or persons who hold the role Hotline for Europe.

Admittedly, natural language queries include some difficulties and restrictions. The term *natural language query* is only correct in so far as a subset of the natural language can be used for possible queries. Although the EKB provides a generic mechanism to support various languages through a language switch, still only a restricted set of terminology can be used. Secondly, restrictions in terms of grammar and vocabulary apply, as well. However, compared to `SELECT location FROM room_of WHERE name = (SELECT manager FROM manages WHERE organizational_unit = ACME)`, the above listed queries are much more understandable.

The query component consists of three modular parts, allowing for a separation of the actual user interaction and the background retrieval process. Therefore, it is also ensured that the query functionality can easily be inserted into various applications. Figure 5-17 shows the components.

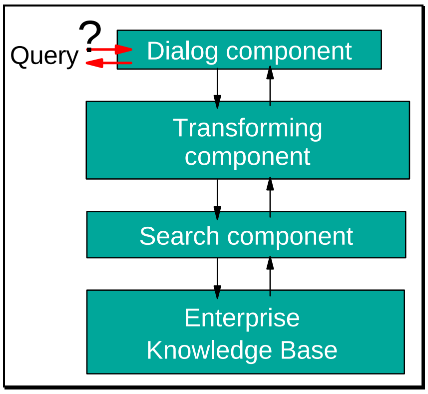


Figure 5-17: Query components

The dialog component accepts the natural language query. It provides functionality for editing and presents the result of the query to the user. A query is then transferred to the transforming component, which calls routines for performing the lexical check of the single terms and then transforms the query into the semantical representation of the organizational repository. Errors that occur during the checking are returned to the user who can then correct the query. Then the search component parses and interprets the query in order to find the answer in the EKB. Finally, the result is returned to the user.

This repository-driven method for retrieving information from the EKB is to systematically structure a descriptive name. The needs of the user are described in a formal query and the database system's search component retrieves the data. But there are some situations in which systematical retrieval is difficult or even impossible. The end user:

- is not familiar with the comprehensive data model GEIMM
- is not familiar with the contents and definitions of the particular repository
- is not proficient in the language procedures used for the definition of the required information
- has only a vague retrieval target (the user is looking for something *fitting* or *suitable*)
- has a clear retrieval target but lacks information for specification through the language

There may be many of these situations in real-world organization design procedures, especially with non-professionals in this field. A common solution is to rely on intuition and start with an exploratory search. The search often begins at an arbitrary location, and while in progress the end user learns more about the organization through the repository. At last, the search terminates successfully. This method of searching is called as *browsing* (see Figure 5-18).

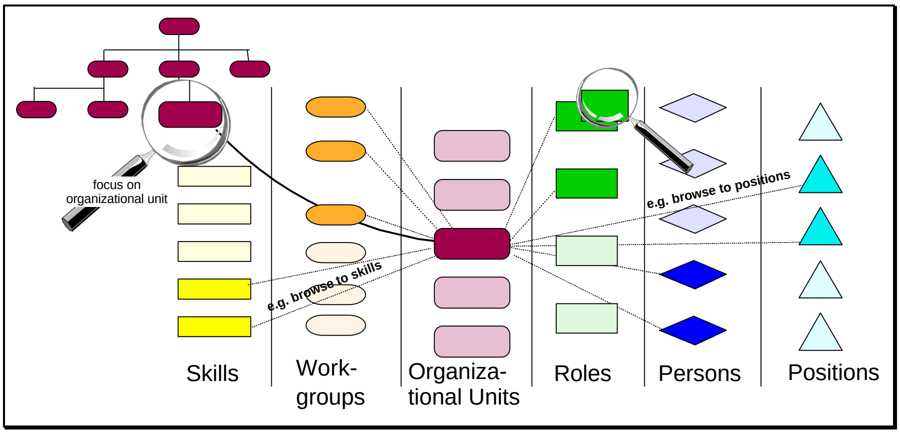


Figure 5-18: Browsing in the infrastructure model

Browsing is implemented in the GroupOrga EKB and in the graphical modeling tool of the GroupOrga framework. Both techniques are explained in the respective sections about the GroupOrga EKB and the GroupOrga OrganizationBrowser.

5.4 Layered System Architecture of GroupOrga

The main purpose of a system architecture is to explain how functionality can be provided. With the various requirements needed to provide functionality for GroupOrga in mind, the objective of an architecture is to specify the separated, required modules. Moreover, in order to describe the architecture of the prototypical GroupOrga system, it has to be considered that it is designed for two purposes: the support of office and workflow management environments, as well as a stand-alone organizational information system. Therefore, the layered GroupOrga architecture model also contains modules that are inevitable for the interaction with workflow and information management.

For the GroupOrga architecture, developing new workflow systems that allow modules to be exchanged in the architecture was considered. In addition, more and more existing applications (*legacy systems*) must be integrated. Both aspects show that a hardcoded, monolithic architecture is not helpful in this scenario. The control of distributed and heterogeneous platforms is another reason for proposing an architecture of modules, rather than large building blocks. Especially for GroupOrga the modules have been designed in a way to address the specific requirements identified in the previous chapters. Each module has been designed with a weak coupling and with independence of the other modules.

To overcome the problems of a single, complex, and unstructured attempt, a layered approach for the architecture model was adopted. The complete organizational (sub-)system is broken down into a number of layers each of which performs a well-defined function. Conceptually, these layers can be considered as performing one of two overall functions: back-end-dependent functions and front-end/application-oriented functions. This, in turn, gives rise to two operational environments: the back-end environment and the application environment. The back-end-dependent functions and front-end/application-oriented functions are then implemented in a number of layers. The boundaries between each layer were selected in order to enable exchangeability of modules. Each layer performs a well-defined function in the context of the system and has an interface between itself and the layer immediately above and below it.

This is shown in diagrammatic form in Figure 5-19.

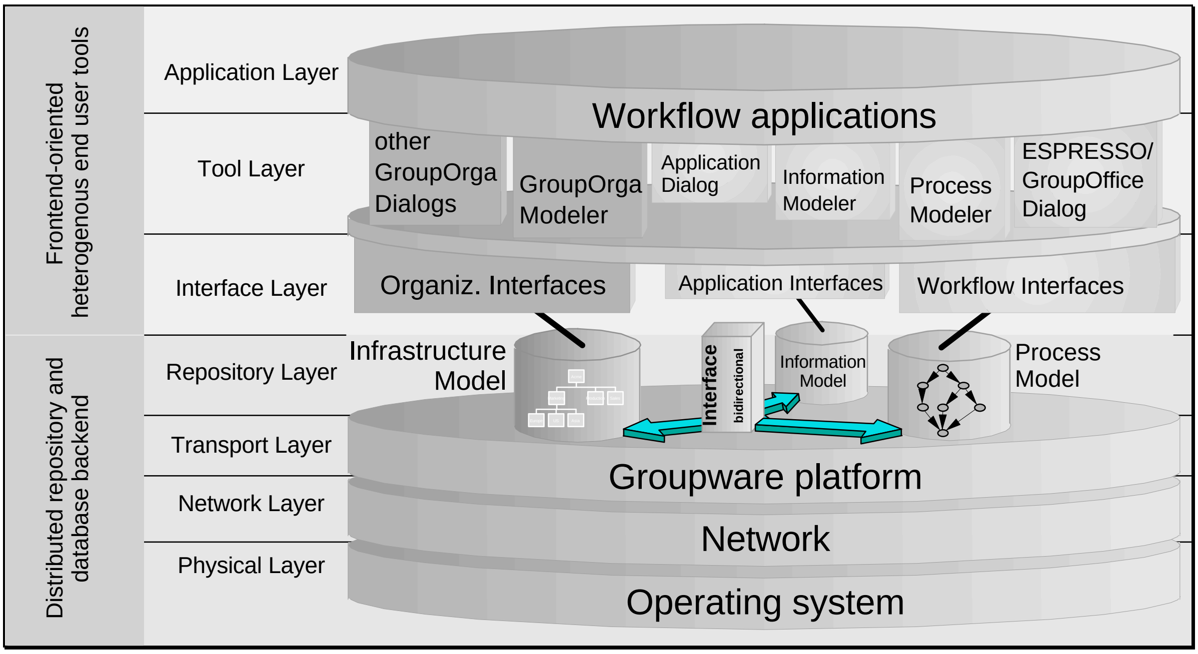


Figure 5-19: GroupOrga layered system architecture

The lowest four layers are back-end-dependent and are concerned with the data communication being used to link two interacting computers together. Apart from the repository layer, these layers are not the focus here, since their technological base is provided to the project.

In contrast, the upper three layers are front-end/application-oriented and are concerned with the interaction of the end users in their participation in the design process.

The intermediate repository layer acts as a middleware. Middleware are services which operate in-between the technological platform and the end user applications. In GroupOrga, the middleware provides the complete enterprise model information via interfaces to the respective office and workflow applications in a system- and application-independent manner. This layer is realized on top of the distributed groupware platform Lotus Notes.

From a technological perspective, GroupOrga is a system of synergistically integrated modules and specialized tools on top of this distributed platform. For that purpose, relevant products with relevance on the market were combined via interfaces with components developed in the GroupOrga project.

Furthermore, the components in the front-end area can be separated by function and model, for process support and control, application design, and organization design, information and administration. Figure 5-19 distinguishes likewise on the tool layer from right to left.

Section 5.5 introduces selected components for organization design from the tool layer, from the interface layer and from the repository layer which have been conceptualized and developed during the course of the GroupOrga project. Due to the specification of the *complete* GEIMM—in contrast to only the infrastructure model—as a core component for office and workflow systems, the GroupOrga project has also radiated into the information and process modeling fields. As a result, two GUI modeling tools for these fields were developed. To complete the picture, they are sketched briefly at the end of this chapter.

5.5 GroupOrga Prototype System: Selected Components

Besides the introduction of the partial models and the project concept, one of the main deliverables of the GroupOrga project is a computer-based toolset that will help in the organization design process. These integrated tools support predefined functions in the organization design.

The use of such this toolset offers two important advantages. First, the tools guide the end user designers through the whole design process, allowing them to gain time and improve the quality of the infrastructure being modeled. Secondly, the use of the tools controls the manner in which different modeling tasks are performed, ensuring consistency between the initial requirements and the resulting concrete enterprise model described in the EKB.

The toolset, which is graphically oriented, provides the end user designer with a set of graphical user interfaces for the different tasks with menu selection or inputs. Some examples: designing a new workgroup structure starting from the list of existing workgroups, defining the resources held by an organizational unit according to surveys in the organization, and updating the role assignment between employees.

This section presents single tools and modules of the application framework of GroupOrga. The constituting front-end components are the GroupOrga OrganizationModeler, externally connected applications (including GroupOrga Connector), and the EKB on top of the distributed repository architecture.

The organizational information browser, both in its graphical and knowledge base front-end realization, supports the daily information retrieval and assignment of entities to tasks.

For the ex-post coverage and analysis of organizational data, the graphical GroupOrga analysis tool provides adequate functionality. It assists in the creation of comprehensive management information and diagrams based on the data about organizational features, especially for the improvement of organizational infrastructures.

The subdivision of the front-end components into modules follows three basic motives. First, the single components provide heterogeneous functionality. Secondly, the various modules operate partly on different subsets of data from the overall enterprise model. While, for example, the graphical modeling tools access the entities in an infrastructure model, the organizational analyzer only retrieves statistical data from the repository and causes no modifications of the single entities. Third, the modules are provided to give differentiated user groups adequate support in their specific work.

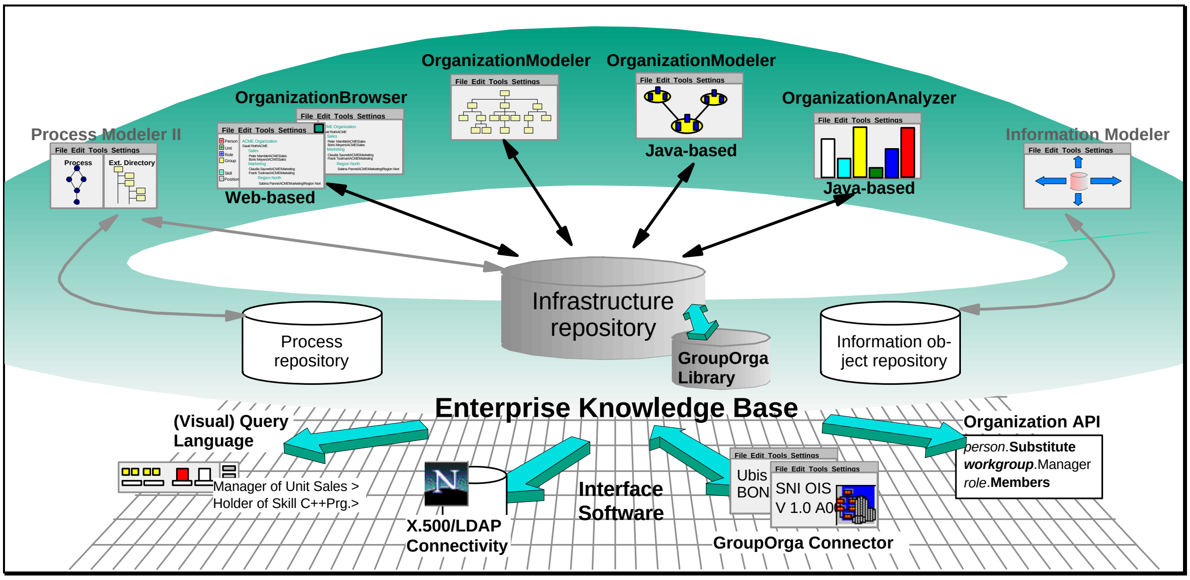


Figure 5-20: GroupOrga prototype runtime system and tools

The tools presented in this section are displayed in Figure 5-20, which shows the distributed repository architecture of the GroupOrga EKB and its modeling tools, the GroupOrga OrganizationModeler and the Java-based Organization Modeler. The GroupOrga OrganizationBrowser and the graphical GroupOrga OrganizationAnalyzer are a further aspect, together with interface software such as the X.500 integration and GroupOrga Connector. Other important building blocks present the Organizational Library and the GroupOrga Query Language, as well as the Organization API. Two minor tools (with regard to GroupOrga concepts) are also presented: the Process Modeler II and the Information Modeler.

5.5.1 The Enterprise Knowledge Base

The decision to implement the GroupOrga EKB as a Lotus Notes groupware database was mainly driven by the fact that the main requirements (distribution, provision of programming interfaces and security) can be fulfilled with this technology. Although the GEIMM is a model that was defined using the EER modeling technique, the following gives reasons for implementing the GEIMM in a groupware database structure.

First, to install the GEIMM in a Lotus Notes database in a strongly relational fashion is not a problem, as was proven with the creation of the GroupOrga repository. Moreover, relational consistency in the model is guaranteed by the organizational modeler, and the consistency validation formulas in the database. Secondly, the database structure is intended to be a back-end repository and not a planning system. It serves informational and documentation purposes, rather than the functionality for heavy computation in itself. For these purposes external tools are in existence or were developed during the course of the project. Third, purely relational database management systems usually do not allow for extensive distribution and for participation of the end users.

Functionality of the GroupOrga EKB

The knowledge base was set up to assist in a wide variety of information needs. Specific elements were implemented to take the various aspects into account, four of which are explained:

- Documents serve to represent the organizational entities and their attributes
- Relations create the network-like connections and assignments between entities
- Views are used to list the entities according to their type and attributes
- Navigation functionality allows for easy navigation among the views and documents

While the latter two are tackled in the next section, the first two are the focus in the following discussion.

Entity documents and relations

Lotus Notes is a document-oriented database architecture, which is why each entity specification is stored in a document. A document of the GroupOrga EKB contains flat information and actions which can be carried out on the information. Two selected examples of EKB representation for GEIMM entities are definitions of *person* and *workgroup*. The illustrations are taken from the German EKB prototype.

Person document

The *person* document contains all information about an instance of the *person* entity. The document takes all relevant information specified in the GEIMM (see section 4.3.2.1.1), such as name, location, phone number, and so on. The document consists of seven logical sections:

- **General.** Information about the name of the person
- **Teams.** Assignment to organizational units, workgroups, and roles
- **Further information.** The location of the person and the contact information
- **Costs.** Cost information that is necessary to use the EKB with project management systems (like GroupProject)
- **Employment.** Current availability of the person in terms of being assigned to tasks

- **Profile.** Description of the person's knowledge/skill attributes
- **History.** List of past assignments to projects, workgroups, organizational units

Figure 5-21 shows the general design and functionality of a *person* document in read mode.

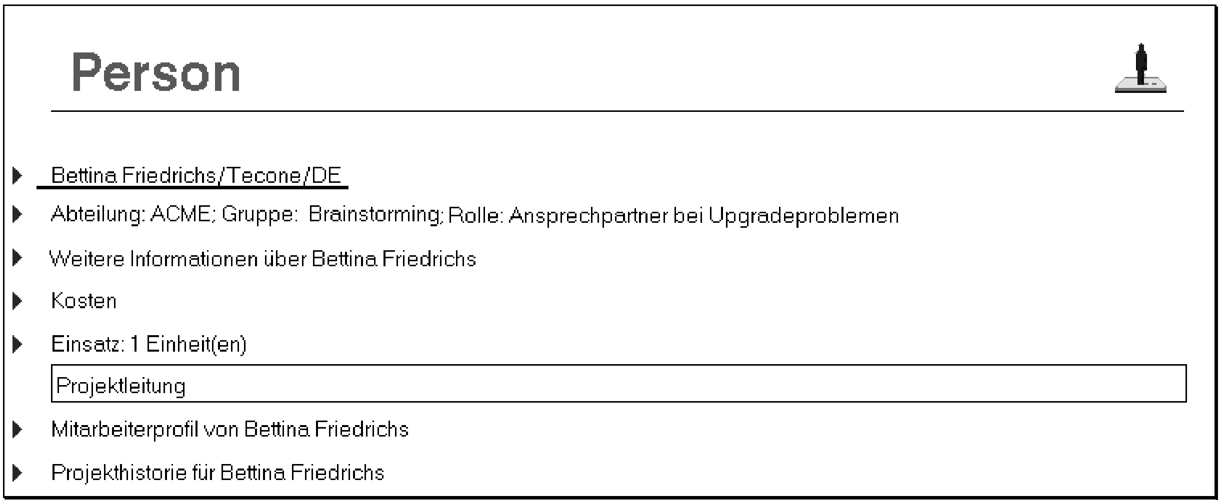


Figure 5-21: Person document with collapsed sections

All other entities in the EKB relate to the *person* document. That is, all existing relations between a *person* entity and other entities in the way they are defined by the GEIMM are invalidated when a *person* document is deleted from the EKB. However, since the other components of an organization are defined as single entities themselves (and not as attributes of a *person*), the deletion of a person's entity description does delete the other entities.

Relations can be established in the affected entity documents (for example, *role*, *organizational unit*, or *workgroup*), as well as from inside the *person* document. In both cases, the relation is saved in the *person* document in order to retain the consistency throughout the EKB as specified in the GEIMM. The *Teams* section gives information about such relations.

The section with *Further Information* contains data about the *location* at which the *person* resides and how the *person* can be contacted (e-mail, telephone, fax, an do on). The *Cost* section is intended for the planning of projects with external project management tools in connection with GroupProject (see [Ehlers 1997]) and was integrated for this reason.

Workgroup document

A workgroup document specifies a grouping of persons for a specific purpose. It consists of five sections:

- **General.** The unique name of the workgroup
- **Teams.** This section allows the designer to engage persons in this workgroup and to identify a workgroup manager. It is also specified whether this workgroup is long lasting or short lived. Moreover, it can be noted if the workgroup is currently in a planning/set-up state, is already active or was dissolved. In consideration of a

learning organization, documents of dissolved workgroups should not be deleted but conserved, so that a later analysis can still access the information.

- **Description.** This section leaves space for a more detailed description of the *workgroup's* purpose and goal
- **Skill management.** The collected skill of all members represents the *knowledge/skill* of the *workgroup*
- **History.** Information concerning the workgroup's past, activities and engagements

Figure 5-22 shows how this is realized in the GroupOrga EKB.

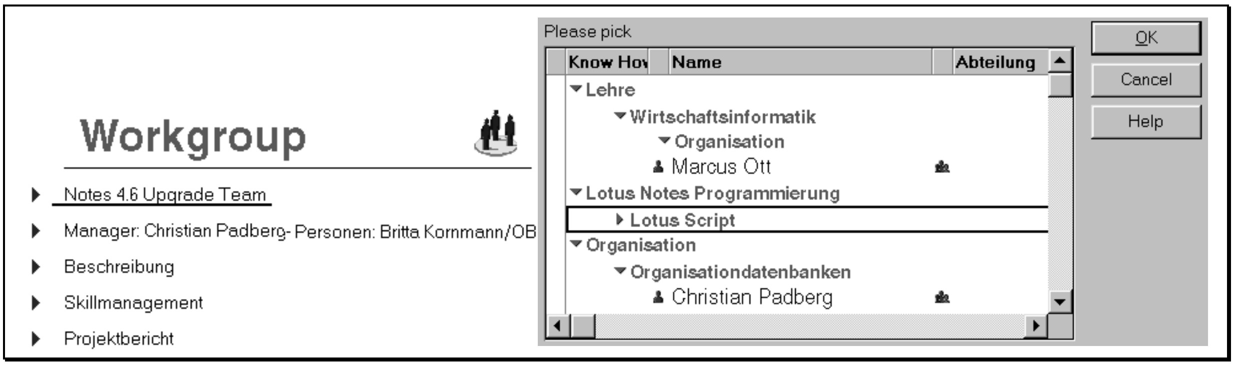


Figure 5-22: Workgroup document with pick-list for workgroup members

Since it is the intention to form a *workgroup* based on the members *knowledge/skill*, the members can be picked from a list of bearers of various *knowledge/skill* entities. This is shown at the right of Figure 5-22.

Further documents are provided for all other entities specified by the GEIMM. To name a few, there are documents for the detailed specification of *position*, *resource*, *knowledge/skill*, *role*, *organizational unit*, *authorization*, *location* and more.

Besides the specification of organizational information in the described documents, another aspect of the EKB is to retrieve this information and to provide it to other applications. For this purpose, the end user is provided with various user interfaces for navigation. This is the focus of section 5.5.1.1.

5.5.1.1 Retrieval Functionality and Query Language

The EKB provides retrieval functionality to two different types of questioners: the human user who accesses the EKB directly, and external applications which use interfaces to the repository to retrieve information. A third form of information retrieval, organizational browsing through the graphical user interface, is discussed in section 5.5.4.

Navigation and query language for end user searches

The user has four different ways of searching the knowledge base:

- Using navigators and views which preselect and categorize parts of the EKB

- Browsing the entities in the EKB by traversing relational links between them
- Primitive name searching via built-in type-ahead and search functions
- Making common sense queries from user oriented front-ends

Navigators and views

Graphical navigators are available for each entity type of the GEIMM. The user can select specific views of the infrastructure data in the repository. An initial navigator serves as orientation for the user, who can decide whether to create new entities or to browse the EKB. The browsing brings about other navigators for the various entity types. Figure 5-23 shows a set of navigators for this browsing functionality. On the left, the initial navigator is displayed, which, when selecting the *person* function, results in the navigator displayed in the middle. If the *organizational unit* function is selected, this in turn displays the navigator on the right.

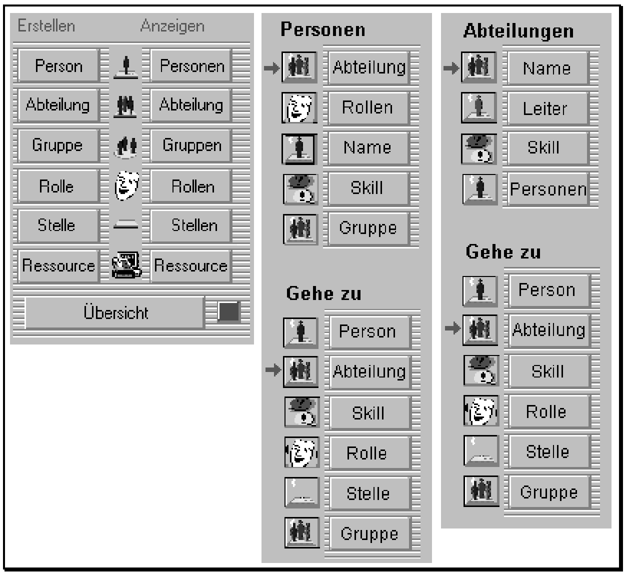


Figure 5-23: Navigators for efficient browsing

At the top, each navigator contains various functions to categorize entities of the currently selected entity type (*person* or *organizational unit* in this example) in the view pane to the right. The lower functions change from one entity type to the other without having to go back to the initial navigator. Thus, the infrastructure model can be viewed and explored effectively and quickly from many viewpoints.

Once an entity is located with this browsing functionality and selected from a view, the end user can open the document with its detailed description. From there, the EKB can be explored with the functionality of navigation and browsing.

Browsing and navigation

The aim of the overall concept is to foster the development of a non-hierarchical, network-like organizational structure. This goal is supported by the EKB. Once entered into an entity description, the user can traverse the relations between the specific infrastructure entities and can navigate and search within the network structure, without having to return to the hierarchical layout. Not only does this browsing does restrict the user to navigating within one entity type (for instance, only within the tree of *organizational unit* entities), but it also spans the diverse entity types as far as relations exist. Each entity document contains a navigation bar that gives access to all other related entities. Figure 5-24 shows the possible browsing directions for the *person* document.

Figure 5-24: Navigation bar in the person document

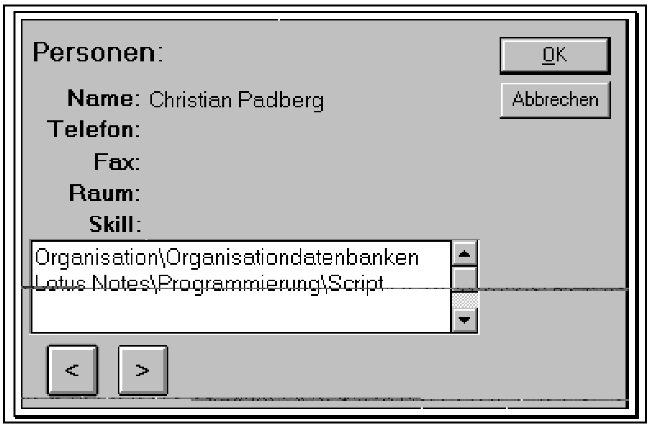


Figure 5-25: Dialog for browsing between entities

Technically, each entity in the EKB has a unique identifier (the name). When activating one of the above browsing functions, the information about this relation is retrieved from the document of the source entity and the related entity is opened. When there are multiple target entities, the user can make a choice via an enhanced pick-list dialog-box (see Figure 5-25) and navigate to the desired target entity.

Primitive name search

Lotus Notes supports two very comprehensive technologies to search for character strings that uniquely define an entity in the knowledge base. In connection with an appropriate view that lists all entities in the EKB by name, the *quick search* function jumps the user to the document that describes the entity. The *search functionality* goes one step further and provides a search engine for full-text retrieval of all documents in a repository. Although both capabilities are very powerful, they provide no understanding of how the entities are related, what the existing network looks like, or what other entities exist besides the one that was found. (For more information on primitive name searching in the EKB (see section 5.3.5), refer to the Lotus Notes documentation.)

Common sense queries

The conceptual section (5.3.5) about the EKB mentioned common sense queries, which are also implemented in the knowledge base. Each end user application can be connected to the EKB via the provided interfaces, and its content can be used, for example, to specify actors in a manually forwarded workflow, select actors who are given access to documents in an office management system, or identify the receiver of an e-mail. In the third example, instead of entering a specific name, for instance, `Francis White/TECONE/DE@TECONE@NET`, in the "To:"-field of an e-mail dialog component, the user can specify the natural language query `Manager Unit ACME` (see Figure 5-26). The GroupOrga EKB contains a standard function that needs to be copied into any end user application to enable a lookup query onto the knowledge base in order to *resolve* the natural language query into a person's name.

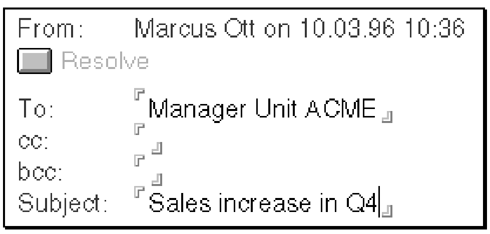


Figure 5-26: Mail dialog with natural query

For end user requirements, this process is even more simplified. This is why a function is provided by the EKB to *compose* the query rather than to formulate it "freehand". In the GroupOrga EKB, the metaphor of a filing cabinet and other visuals is used. Each file represents a type of query, or a predefined section from the knowledge base and is labeled in this context (function, *organizational unit*, *workgroup*). The single files are context sensitive. By selecting an entity type, another list (file) is displayed with pre-selected details of the search.

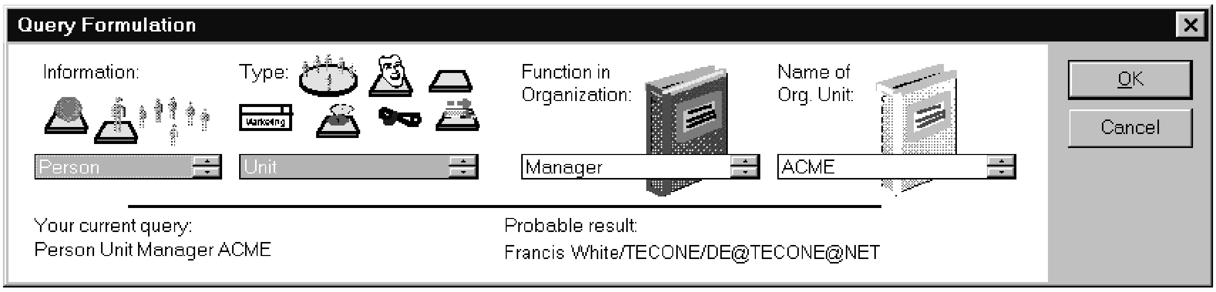


Figure 5-27: Query composition via graphical dialog

The query dialog box (Figure 5-27) can be closed after use, and the resulting query is transferred into the application from which it has been executed.

In summary, different forms of interaction are provided for the end user based on the different conceptual types of query identified in section 5.3.5, such as navigation, keyboard input or graphical forms of query formulation. The two latter forms can be combined by selecting the type of query and keying in the parameters for it.

Application Programming Interface For External Applications

The query services provided by the EKB application programming interface allow external applications (like a WfMS) to direct requests to one of the entities in the EKB. The API is a Lotus Script-based interface. Table 5-3 presents an overview.

Query service provided	Description
person.**ManagerOfWorkgroups**	The workgroups that the person manages.
person.**Roles**	The roles that are held by the person.
person.**Substitute**	The substitute of the person.
person.**Authorization**	The authorizations that the person has been assigned.
role.**Members**	The persons who hold the role.
workgroup.**Manager**	The Manager of the workgroup.
skill.**Bearers**	The persons who bear this skill.
Software.**ComputerType**	The type of machine the software agent needs to be run on.

Table 5-3: Selected services of the Organization API

These query services do not yet provide any further function (for example, creation or deletion) for the modification of a particular entity. To do this, the corresponding modeling tools or the EKB front-end must be used. In contrast, the GroupOrga Connector provides this functionality as a *specific* external modeling application, as is shown later in this chapter.

Members property
Persons who belong to the workgroup.
Defined in
Workgroup
Data type
Array of strings.
Syntax
To get: *memberNamesArray* **=** *workgroup***.Members**
Return value
memberNamesArray
The names of all members of the workgroup.

See examples

Related Topic

Figure 5-28: Definition of API service *workgroup*.**Members**

Figure 5-28 shows an example of the definition of an API query service.

5.5.1.2 Versioning and Time Consideration through Archives

For versioning and provision of structural data that records past forms of an organization, the EKB uses a functionality provided by the underlying groupware platform. Each entity description can be modified. When it is saved, the older version is archived as a *response* of the original. Thus, version tracking maintains the history of changes to an organizational model.

Since the updated and current form of an organization is always the most important and most frequently read, the latest version of each entity document is accessible via the navigation and browsing mechanisms. Previous versions and the original are accessible via views that are not intended for day-to-day use, but for analytical purposes. Moreover, the graphical modeling tools can access this view to provide prior structures in libraries (as a base of new design), as well as the analyzing tool (which compares the prior designs).

Older versions of an organization model can be archived periodically or manually from the productive EKB to an archive repository. Archiving organizational models that represent the past by regularly moving data out of the working repository and into an archive significantly improves the performance and conserves disk space on the end users workstation.

An archive frequency must be determined for this process. Relatively static and infrequently modified designs may be archived infrequently. Heavily used repositories, such as those used during a major reorganization, may be archived more often. The structure of a GroupOrga archive is the same as that of a regular EKB. But since it is the goal to analyze statistics within an archive repository easily, the creation of additional views in an archive copy for this purpose may be considered.

For simple archiving, a functionality was developed in the EKB to define when and how prior versions are to be automatically archived. It can copy entity documents based on search criteria to an archive repository, then remove those documents from the working repository. The example in Figure 5-29 archives all documents that were not modified in the last 60 days.

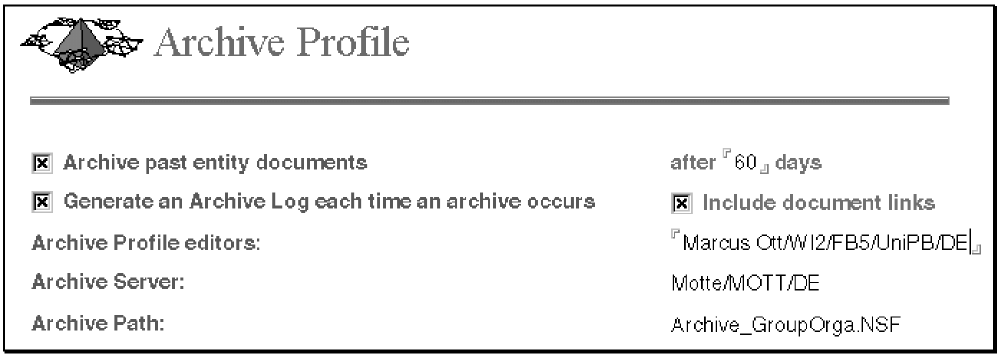


Figure 5-29: Archive profile for the GroupOrga EKB

All entity documents can be viewed in the archived repository as if it were a current EKB. Moreover, the archived entity documents may be accessed via the archive log, but only if it is specified in the archive profile to create a log and include document links to the archived entity documents. Using the archive log helps in identifying the versions and dates of entity documents in different archives.

5.5.1.3 Creating Awareness

Creating awareness about infrastructure changes was realized in the GroupOrga EKB similar to the versioning functionality. Each time major or minor changes to the infrastructure model are made, the affected end users are notified of these changes automatically. Based on the attributes of the entity and its administration responsibilities, the EKB launches a message each time an entity document is modified.

As a first choice of related entities to be made aware, the list of persons with administration responsibilities is checked. Every responsible person within an organization that has design or insertion access (except the actor who is making the modifications) is notified about any structural changes. If the modifications to the entity are fundamental, the awareness message is also sent to the holders of read access. Since the concrete vehicle for such awareness messaging is e-mail, the persons with design and insertion are addressed directly, while persons with read access receive a carbon copy of the message.

With the Organization API, it is further determined who is also related to the entity in question. When a *workgroup* definition is modified, for instance, query services such as *workgroup***.Manager** or *workgroup***.Members** are evaluated in order to specify whom to make aware. In this case, the resulting list is compared to the list of administration responsibilities to prevent sending messages twice. If the threshold were at a level of *two*, other relations would be traversed as well, such as notifying the managers and members of superordinated workgroups whose names are determined by means of the *workgroup*.**Arrangement** query service. Because the implementation in the GroupOrga EKB currently corresponds to a threshold of *one*, only direct members of an entity are notified of changes.

5.5.1.4 WWW Access to the Enterprise Knowledge Base

While many work settings would clearly benefit from cooperation, unfortunately the lack of common computing infrastructure within a group often prohibits deployment of dedicated technology. In addition, this causes serious problems for system developers who must pay close attention to issues of heterogeneous machines, networks, and application software. The scale of different user type classes in Table 5-1 termed the requirements of this user type as *"Push-button" information needs*. For more integration of these end users into the structural information and design process, a common infrastructure that addresses problems of integration is required. In its current form, the WWW provides a simple and effective means for users to search, browse, and retrieve information.

Because of its cost-effective communication and its internationally wide-spread availability, the WWW a highly attractive platform for the initial integration of end users into the structural design process and for those who *only want to be informed*. Because of the complete integration of the groupware system Lotus Notes into this global network (especially since version 4.6 and the Domino Server), the information in a GroupOrga EKB is fully accessible from the WWW.

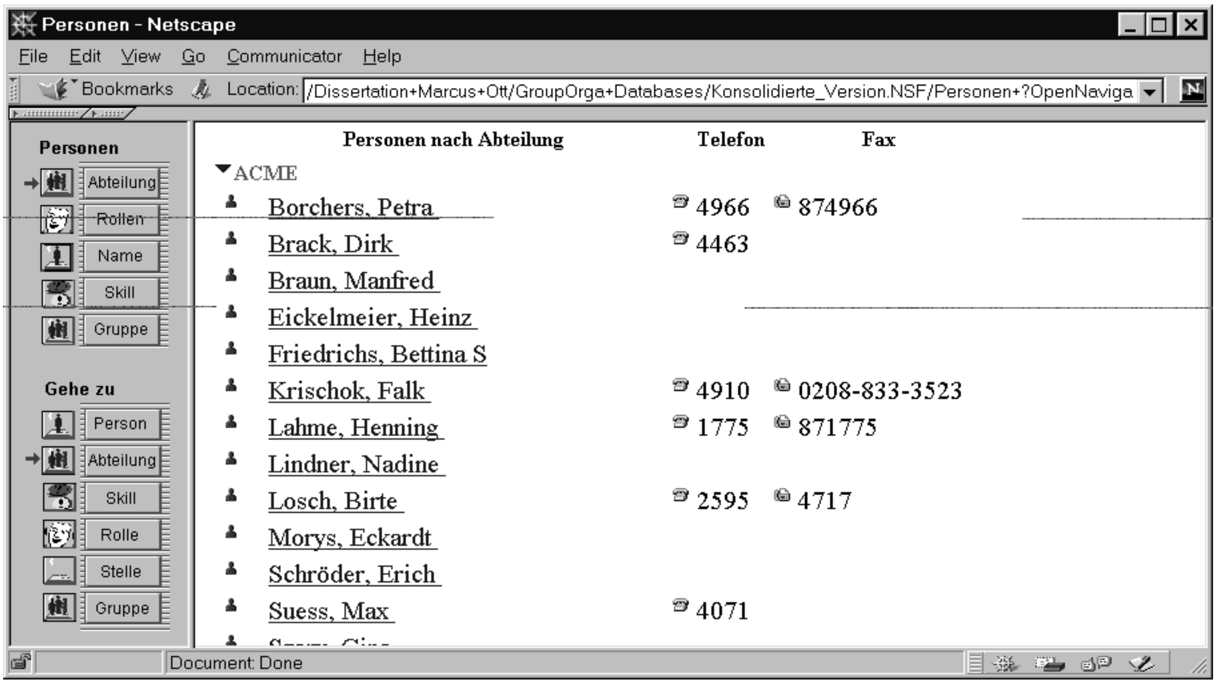


Figure 5-30: The GroupOrga EKB accessed via WWW

Figure 5-30 shows the access to a GroupOrga EKB from the WWW with the Netscape browser. The WWW-based distribution of structural information, as well as the sporadic modification of infrastructure data, is realized in the GroupOrga prototype in standardized form. Information about the organizational model can be retrieved via any WWW client any available Internet browser application.

Users access the knowledge base server using a standard user-name and password scheme, and the server responds with a navigation functionality to the knowledge base similar to that of a Lotus Notes client. New users are added to the server by completing (or having an existing member complete) a simple registration form. The information is the used to check if the user is already a member of the knowledge base.

Figure 5-31: Registration for WWW users

This is usually the case since each user is registered in the groupware environment. If not, an initial password is issued. Once logged on, users can change the password. The registration form is displayed in Figure 5-31.

5.5.2 The Graphical Organization Modeler

The GroupOrga OrganizationModeler is positioned as a graphical infrastructure design tool on top of the EKB. It supports the simple and intuitive creation and modeling of an organization's infrastructure. It was developed as a scientific subproject to the GroupOrga research. The following sections relate to the structural and conceptional aspects of the application. More detailed discussions and explanations can be found in [Meyer 1996] and [Huth 1998].

The design of the organization modeler is guided by requirements derived from the application domain of supporting users in cooperative design of organizational structure. Although other similar tools have been examined ([Ishii/Kubota 1989], [Prinz 1993], [Grabowski et al. 1996]), it was essential to draft and built a tool with enlarged functionality.

Among other reasons, the investigation revealed that existing modeling environments do not meet three major requirements:

- They do not have a programmable interface to access the API in Notes
- They are not based on a large enterprise model geared towards network organizations such as the GEIMM
- They do not support distributed and participative modeling

These requirement scenarios lead to a set of functional requirements that are used in the following to present the design concepts that are mostly illustrated by screenshots of the GroupOrga OrganizationModeler.

5.5.2.1 Graphical Representation of the Infrastructure Model

One of the aims of this research project was to build a user interface that would make repository content more accessible to native users. Therefore, the GroupOrga project accepts previous research in this field (see [Batini et al. 1992]) and their justification of the need for a graphical interface. The following statements concentrate on the essential features for such an interface. In the organization modeler, the two areas of database handling and graphical access meet to form an intuitive, user-friendly environment.

Graphical methods used to diagram organizational facts impart, clarify, and modify information. To be used meaningfully, the graphical representation of the GEIMM's infrastructure model must adhere to some criteria. In regard to its objectivity, the representation must be complete and in accordance with the facts. Completeness refers to the fact that the graphical representation is only useful if all data is displayed. It would make no sense to only display some infrastructure entity types and not the rest. In terms of accordance with the facts, the model should displays the information unchanged.

In contrast to existing graphical database visualization (for example, with pure EER-diagrams), the GroupOrga OrganizationModeler uses terminology that is problem specific and relevant to business processes. Moreover, the OrganizationModeler displays semi-formal model representations of the underlying meta-model. This significantly enlarges the expressiveness from the viewpoint of the unskilled end user who would generally want to ignore the structure of the accessed repository.

Besides merely displaying the infrastructure and giving access to it, the organization modeler also provides a visual design language. This visual design language handles the elements of the model (the entity types, the relations, their attributes) that do not necessarily have a visual representation, and provides interaction by direct manipulation. For example, the organization modeler illustrates the definition of a relation in a diagram by linking, rather than representing it in a table. To overcome a lack of formalization in defining the visual interaction mechanisms, the operations on the enterprise model were directly applied to the graphical operations (such as the selection of nodes and drawing of links/edges).

The question of which forms of representation for the enterprise model are supported is an important one for the tool aspect. The concrete form of the supported graphical elements is subjective, while the number and semantical meaning of the graphical elements is important for the diagram's usefulness. Thus, an icon semantics partly with dual representation was used (see [Sheng Liu/Wei Tai 1989]). In a dual representation, the *image* is the external graphic representation and *text* is added for further description. Examples are the graphic representation of an *organizational unit* together with the unit name and unit manager, and the representation of a *workgroup* with the *workgroup* name as depicted in Table 5-4.

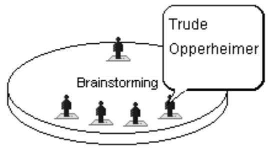
Image		
Text (meaning)	*organizational unit* 'ACME' managed by Ludewig Schöbelmann	*workgroup* 'Brainstorming'

Table 5-4: Examples of images with text

While, *organizational unit* and *workgroup* are abstract, for natural things, such as *person*, the natural shape was used for the image (see ). For the entities that are not easily expressed, creative license came into play. The *role* icon (see ) was borrowed from a theatrical background. Since the two entities shown in Table 5-4 are abstract, artificial images that have been standardized or have become conventions as representation were chosen.

Another concept in the organization modeler is the representation of relations between entities.

A widely-used technique that is standard for hierarchical infrastructures is that of decompositional diagrams, better known as organizational charts. Various forms of decompositional diagrams can be found in practice. Two of the most commonly used, the vertical and the horizontal tree structure, were implemented in the organization modeler. The charts are interesting and easy to understand. For quick guides, they are effective, simple, and unequivocal. This, plus the fact that they are still an accepted means for organizational understanding, was the reason why they were sustained for a transitional stage in the GroupOrga project. Meyer gives a detailed insight into the considerations about the realization of hierarchical charts in the GroupOrga OrganizationModeler (see [Meyer 1996], pp. 51ff.).

In terms of the goal of the GroupOrga project, the representation of organizations in hierarchical charts is insufficient. The pictorial design of workgroups illustrate richer concepts than those represented by charts and more precise symbolic elements than those in charts. This is presented in the following example.

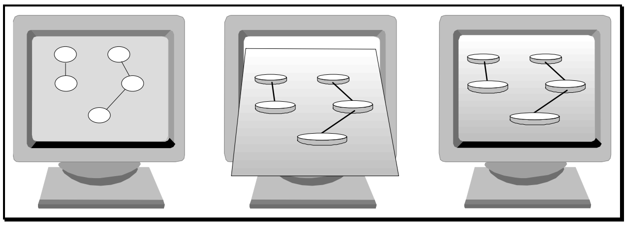


Figure 5-32: Two- and three-dimensional perspective

Workgroups do not comply to a hierarchical structure, and if structural connections exists between entities of that type, they are considered of the same rank which interact in single or dual directions. Because of this, workgroups are best represented on a single level of display (in contrast to a tree structure). But since the computer screen is a vertical plane, a two-dimensional (flat) representation of workgroups implies a hierarchy nevertheless. This brings about the idea to display the workgroups and their relations on a three-dimensional plane, which simulates a three-dimensional work space. (For an investigation of the three-dimensional display for modeling purposes, refer to [Fahrwinkel 1995].)

From the attributes and relations that further describe a workgroup, only four were chosen to be graphically displayed or shown as the text/meaning part of the image: name, members (via relations), amount of members (computed from relations), and manager (via relation). The long-text description would take too much space, and network relations to other workgroups are indicated by edges. The size of a workgroup image grows with the number of members, and the manager is displayed at the rear of the workgroup image. Touching the image with the mouse pointer causes a context sensitive dialog to appear and reveal further details. A workgroup image is shown in Figure 5-33.

5.5.2.2 Modeling of Structures and Central Concepts for Design

The creation and modification of organizational infrastructures takes place on top of a graphical user desktop. Graphical representations of entities are created, which are differentiable on various levels of the model. The representation occurs in a form where a graphical image is assigned to each entity or in tabular form. Each entity can be further specified by defining attribute values and setting up relations.

The organization modeler is implemented in C++, an object-oriented programming language in connection with platform independent StarView functions from Star Division. This was chosen because it somewhat guarantees hardware independence, making the tools available on Windows, OS/2 and some UNIX platforms. The GUI meets the design guidelines of the operating systems MS-Windows and OS/2 Presentation Manager, and provides the standard functions for easy structural modification, such as *Cut*, *Copy* and *Paste* or *Drag&Drop*. Intuitive handling mechanisms support the design process with various functions like creating and deleting entities, connecting them or modifying their attributes, as well as selecting any structure or sub-structure available from the EKB. The design space made available is virtually unlimited, that is, the infrastructure design is not restricted by the organization modeler. Figure 5-33 shows a concrete layout of an imaginary organizational infrastructure displayed in the graphical user interface of the GroupOrga OrganizationModeler.

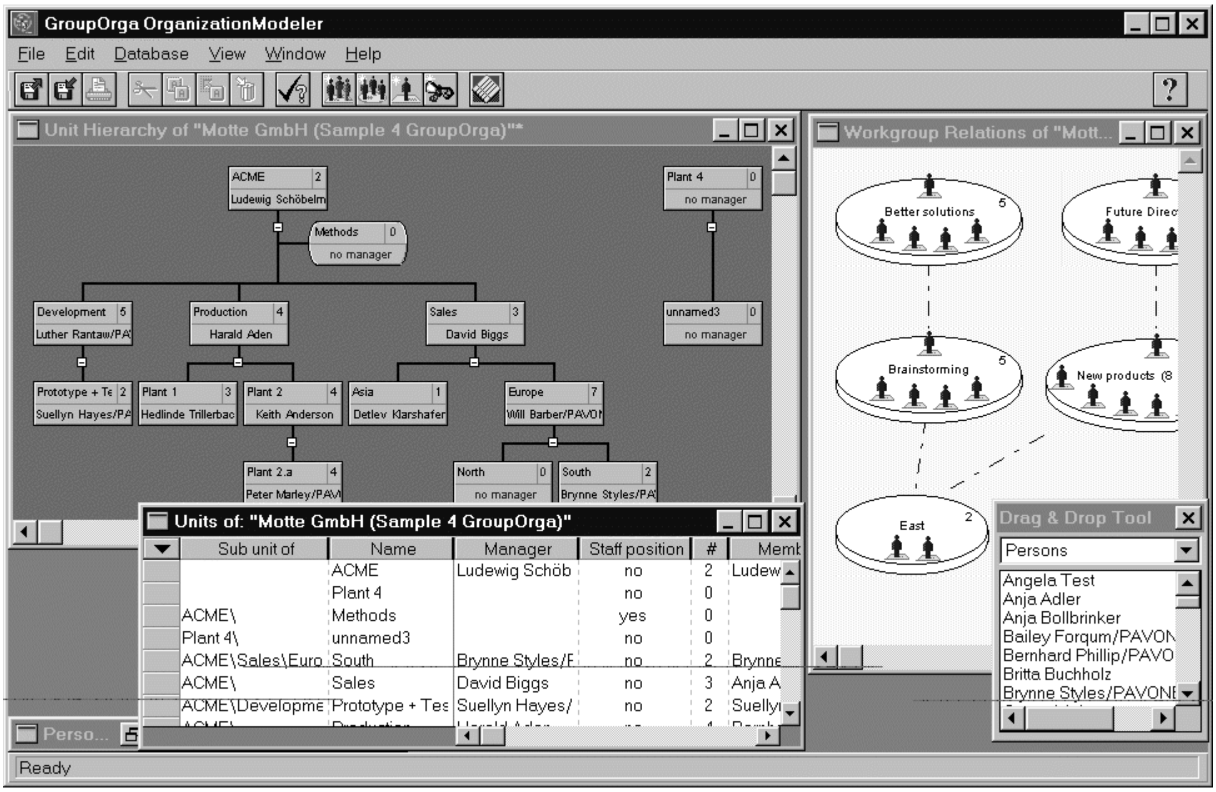


Figure 5-33: GUI of the GroupOrga OrganizationModeler

Of central importance, especially in large infrastructures with many organizational members, are concepts for transparently and lucidly displaying *very large and complex* EKBs. Planning these trusts combining thousands of entities requires powerful solutions. The GroupOrga OrganizationModeler provides these solutions. First, the user can scale the model and make it fit onto a computer screen for overview or navigation (see Figure 5-34). Second, the user can display the model with a varying degrees of detail. For instance, it can be distinguished if and when attributes in the graphical images for workgroups are displayed. Third, the user can switch parts of an infrastructure on or off in order to reduce the complexity of the diagram.

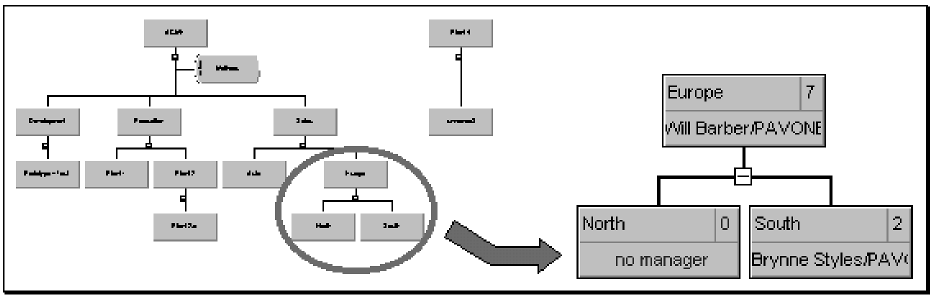


Figure 5-34: Overview scaling as presentation means

In addition to the graphical representation of entities, a foreground attribute window provides additional information that is not graphically represented in any of the views. The data entries in this window come from references to the highlighted enterprise entity and its relation to other entities.

5.5.2.3 Management of Organizational Libraries

Usually, in a repository, there are some forms of an organizational model that have been tested in reality. These past versions, which may be archived in an EKB, represent part of the overall organizational knowledge which the enterprise has gained through its lifetime. A new or modified infrastructure is usually better than the one before it, which is why old versions are a great resource.

Generally, the structures in knowledge bases that reflect established organizational structures can be used as the model for the planning of new infrastructures or as building blocks for modification. Similarly, patterns that were used in the past or those that were conceptualized but not yet implemented may serve the same purpose. For this reason, the GroupOrga OrganizationModeler allows the user to open another design space parallel to the active one, which then provides infrastructures or building blocks from a library (see [Meyer 1996], pp. 86f.). Using Drag&Drop functionality, these building blocks or single entity descriptions can be transferred into the organization model that is currently on the desktop.

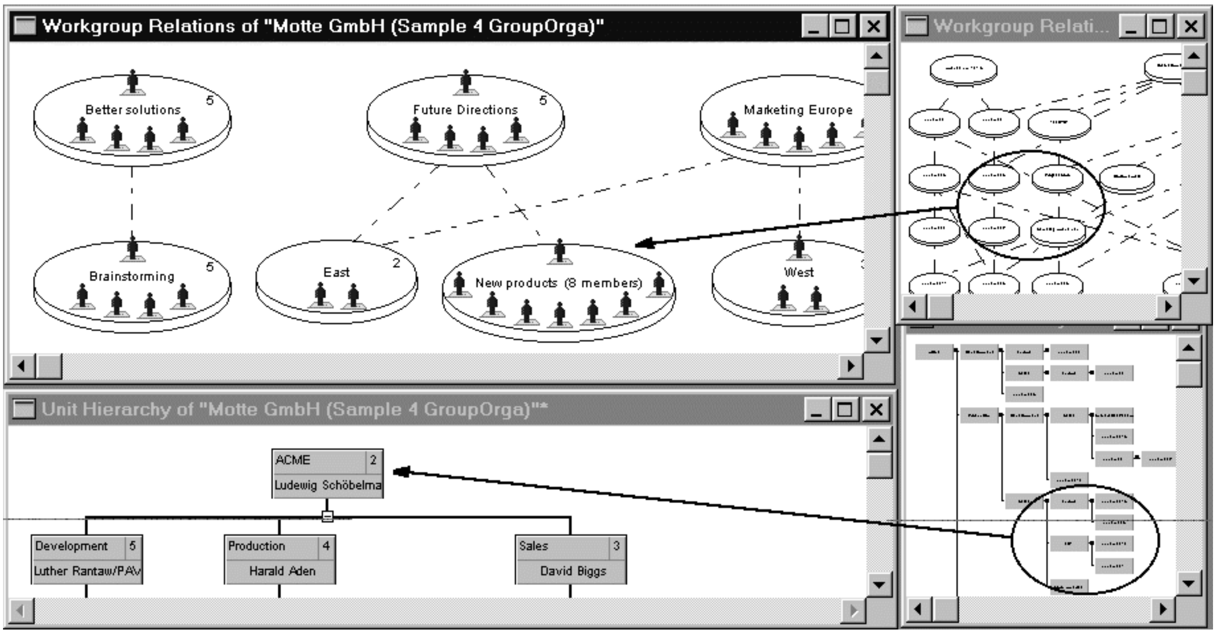


Figure 5-35: Library concept for organization design

Figure 5-35 depicts the design space for the creation and design of an organizational model, as well as a library (to the right) from which patterns and content can be copied into the newly set up infrastructure.

In an evolutionary and dynamic design process, various substructures were proven very efficient while others were created after much consideration. These substructures can be identified and reserved as modules. Therefore, this transfer from a library into an open design is a bi-directional concept, allowing the user to directly store topical structures and settings into the library for later use.

The transfer between the library and the current EKB can be parametrically defined according to what is intended by the user. Afterwards, a transfer (by cut/copy and paste or Drag&Drop) can be initiated as illustrated in Figure 5-35 as a *copy* or *move* operation. As a result from the internal consistency check (section 5.5.2.4), formal restrictions take affect when there are identical entity names in both models. Then, the GroupOrga OrganizationModeler warns the user and offers a dialog for changing the name.

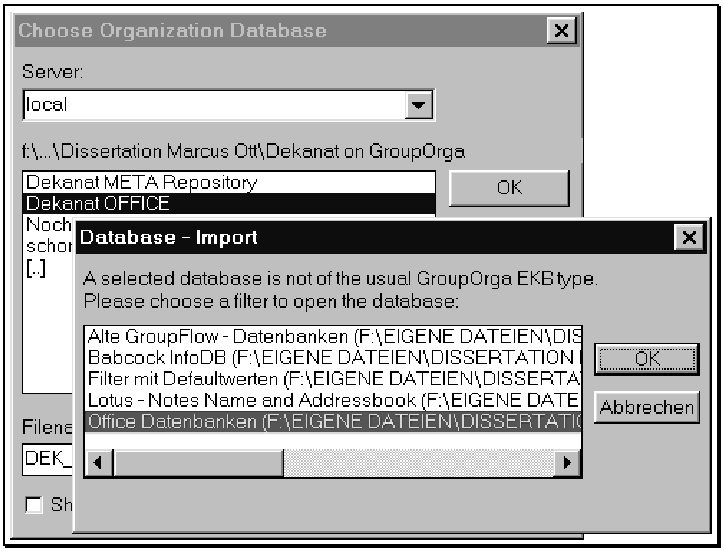


Figure 5-36: Use of other repositories as library

The Notes Name&Adress Books can also be used as libraries. Persons and workgroups defined here are transferable using Drag&Drop in the organization modeler. Similarly, other groupware databases can be used a libraries as well. Examples of such databases are office management applications such as ESPRESSO, workflow repositories or any other database with the nature of a repository. The GroupOrga OrganizationModeler provides a filtering functionality to import these groupware databases (see Figure 5-36). By predefining a mapping table between the data containers in the source database and the necessary information for the organization modeler, each repository can be used as a library. The library structure must provide information which is somewhat transferable (in content) into the generic structure of the GEIMM.

5.5.2.4 Modules for Verification of Organizational Models

The GroupOrga OrganizationModeler actively supports the process of design with algorithms that verify the practicality, advisability and consistency of enterprise models. Backed up by the definitions of entities and relations in the GEIMM, the design tool provides functionality that avoids redundancy and anomalies. It checks for internal consistency of the designed organizational model and creates errors or warnings depending on the type of anomaly found.

For instance, when defining relations, in large structures this can lead to contradictory linkages. In networked (or nested) workgroups, infinite reference loops are easily established in large and complex infrastructures. If not, the verification constantly checks for such inconsistencies. The verification functionality of the organization modeler also ensures the correct modeling of the overall design in conformity with the EER-format of the GEIMM. Lotus Notes does not per se incorporate relational forms of database management. Setting up the EKB as a GEIMM-based architecture with the verification functionality addressed here fully ensures relational consistency.

Two different types of inconsistency are distinguished. *Errors* are the inconsistencies that completely destroy the enterprise model's correctness, such as the infinite reference loops described above. *Warnings* are minor inconsistencies that contradict the rules of the enterprise model, but are accepted during design phase since do not interfere with modeling. Distinguishing errors and warnings is reasonable, since during the design phase not all concrete entities and relations comply to the GEIMM. In this stage, for example, empty organizational units or workgroups may exist.

Each interaction of a user is constantly checked for inconsistencies. When errors occur, a message is created and the intended design action is revoked or offered for correction. In addition to this *on-line* checking, a module for detection of discrepancies, which produces a list of warnings, is provided. While the checking can be initiated at any time, it can also be automatically executed when an organizational model is being loaded from or stored to an EKB. Since all inconsistencies should be resolved *before* putting a model into action, using this automatic option is advisable. Moreover, partial checks can be invoked on selected entity types and their relations, such as the workgroup relations and role assignment.

The following are examples of inconsistencies that create errors:

- Loops in the subordination of organizational units
- Loops in the nesting of workgroups
- Duplicated names for non-hierarchical entities (persons, workgroups, roles)
- Duplicated names for organizational units that are subordinated to the *same* unit of higher rank
- Entities referenced in relations of other entities but do not exist on their own

The following are examples of when warnings are created:

- Organizational units have no manager
- Organizational units or workgroups are empty
- Persons are not assigned to any organizational unit
- Roles are not held by any person

While this functionality concentrates on checking inconsistencies and model verification, another aspect of graphical organization design is that of analysis and statistical information. It was partially conceptualized in the project and is examined in section 5.5.6.

5.5.2.5 Providing Modeling Functions via the Web

The different GroupOrga user classes have different modeling needs. Those who need powerful technological support should be equipped with comprehensive (and proprietary) design technology, while those who require less technology can simply use a standard Web-browser with an applet running on it.

Technically, the Java-based GroupOrga OrganizationModeler is integrated as a Java-applet into the GroupOrga EKB. This repository is then published onto the WWW (as was shown in section 5.5.1.4). From there, the applet can be started on any desktop with a Web-browser. Hence, the role of the groupware platform is enlarged, and thus is used as applet-server for organization design in teams, as well.

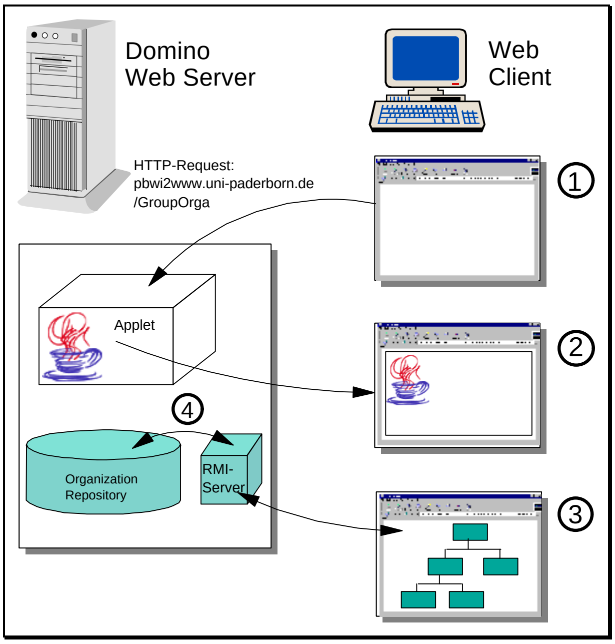


Figure 5-37: Applet invocation for organization design

Figure 5-37 illustrates the following example: After a request from the client browser is sent to the server, the applet is started at the user's desktop where the user logs on. Data about the organization structure is then requested from the server. This data is internally processed and then displayed graphically. After this first interaction, a bi-directional exchange between applet and host (the underlying GroupOrga organization repository on a web-server) is established. The GroupOrga architecture pays great attention to questions regarding data security. Since applet-security does not allow direct access to local resources, the exchange of infrastructure data between the EKB and the applet is carried out via a Remote Method Invocation (RMI) functionality running on the Domino Web Server. Thus, the Domino server handles and observes all accesses to the repository and sends data over the network to the requesting applet. The same procedure is used when data is modified in the applet and needs to be written back to the EKB.

The argument for using Web-based technology only to a limited extent in GroupOrga was based on organizational reasons. For *"Push-button" information needs*, users would not *want* to have more functionality than what is provided by Java-applets, for instance. But even if more functionality at a Web-client desktop was wanted, today's versions of Java or other Web technologies still have drawbacks. The minuses of Java-based software that were encountered during its most recent use in the GroupOrga project will disappear quickly in the future. The minuses were examined in [Ott/Huth 1998b] and in [Huth 1998].

5.5.3 Management of the Distributed Repository Structure

The GroupOrga repository structure is based on an architecture of distributed databases. The distributed repositories can be replicated in LAN, WAN, occasionally connected or mobile environments. This characteristic ensures the availability of infrastructure information and supports decentralized information management.

For organization design purposes, the innovative security concept was realized in the enterprise repository. Each document for the description of an entity in the organizational model was equipped with three additional fields, *design*, *insertion* and *read*, which are in connection with the groupware platform *$read* and *$access* fields that are contained in each document. Each time a document is opened, the contents of these fields are checked and access to the document is access or denied. These fields are also checked when other actions are started, such as designing new entities or inserting new or additional members to groupings (*organizational units, workgroups* and so on).

The functionality of manually or automatically assigning read, insertion or design access to selected entities in the organizational model (see section 5.2.4) is part of the entities' document containers in the organizational database.

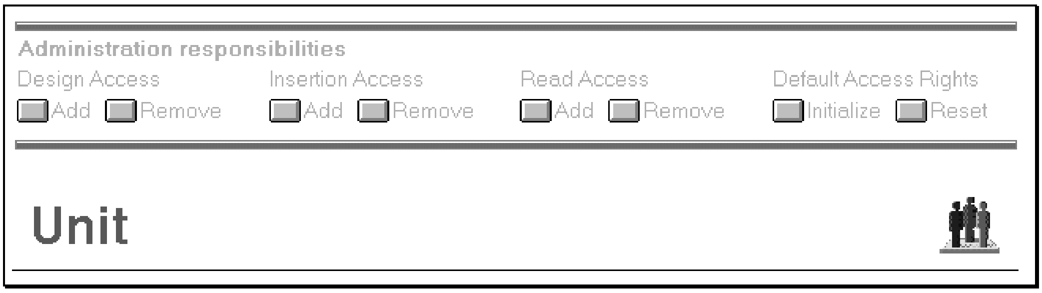


Figure 5-38: Functionality for assigning administration responsibilities

Figure 5-38 shows an example of how this functionality is integrated into the document design. The *Add* and *Remove* actions each add or remove actors to and from the respective access lists. The action *Initialize* executes the initialization of the default read access list for the whole subordinated infrastructure model according to the defined default rule. *Reset* puts the default list back into its native, blank form. For other, non-hierarchical entity types, initialize and reset do not exist.

So far, the administration responsibilities for distributed organization design are implemented in the EKB. The next step involves the graphical modeling tools that assist the end user designer in the specification of access rights. Since the end user has to log on the knowledge base via the ID file, the tools also rely on the concrete administration responsibilities of the end user who is using them. Hence, some organizational information may be not accessible at all or accessible only in read mode, as shown in Figure 5-39.

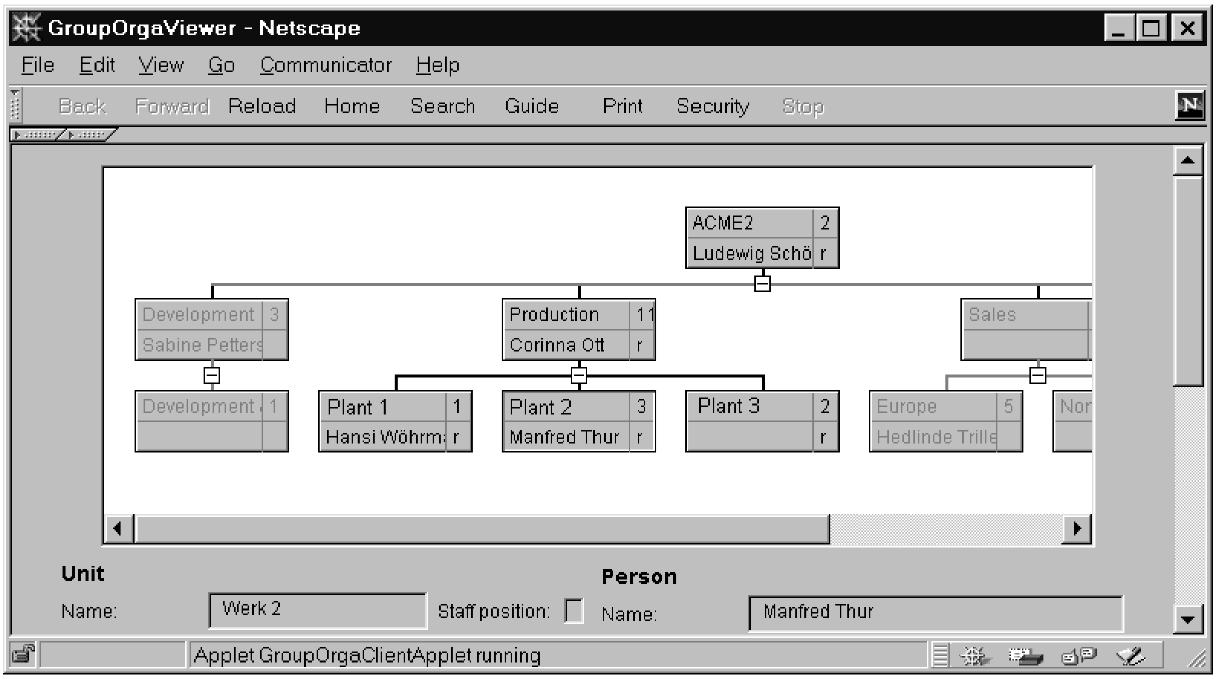


Figure 5-39: Hidden and inaccessible entities due to access rights

Some of the entities are not accessible in this view of the prototype OrganizationModeler (for example, Development and Sales), while the other "r"-marked entities indicate read access only.

5.5.4 The OrganizationBrowser

Basic requirements for organizational browsing in the EKB were mainly discussed in section 5.5.1. Therefore, this section only outlines the functionality of the graphical browsing tool, as opposed to browsing in the EKB via its own user front-end, to draw a complete picture of end user support with the framework.

The project identified that the browser interface must support two typical methods by which users retrieve data from the repository. The first was browsing through semantic relationships, picking the found entity, and selecting it for use in a office or workflow application. This functionality was coined *Point-and-Shoot* in the GroupOrga project. Additionally, it should be possible to easily switch from the productive application into the browsing facility when appropriate, for example, searching for an entity, modifying this entity in course of the participative design, and then browsing via relationships to other entities.

Once the browsing facility of the organizational modeler is launched from an operational application, the user can navigate between organizational entities. Its starting point may initially be defined as the hierarchical tree, the workgroup network or any other GEIMM entity accessible via the graphical tools. This initial starting point depends on the particular user's preference. Once an entity is selected, another dialog box displays the attributes and relationships in which the current organizational entity is involved. By selecting any entity that is a peer of a particular relation, it can be made the new current entity. Thus, users do not need to return to the central browse space (for instance, the hierarchical unit tree) for further browsing. It is assumed that the start dialog box is only used for the initial step into the organizational model.

The sequence in Figure 5-40 illustrates the browsing action from a workgroup starting point to the description of the manager of a group member's organizational unit.

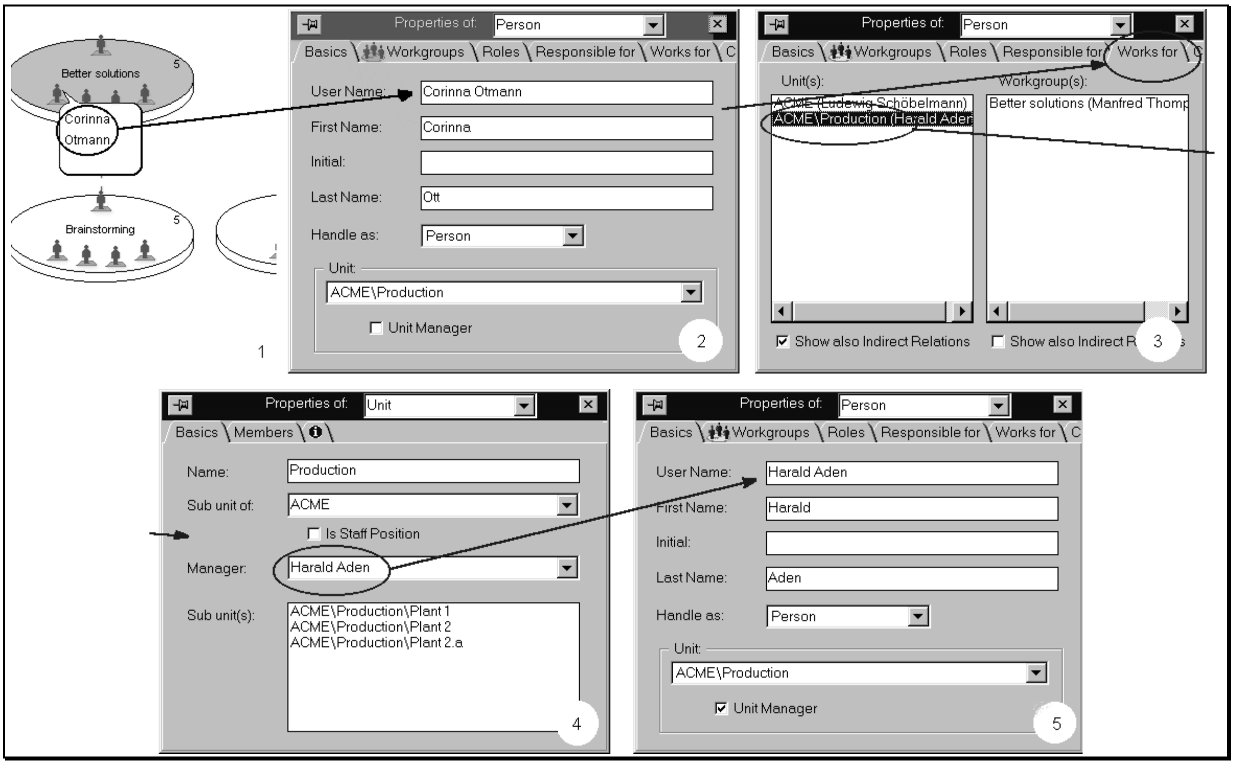


Figure 5-40: Browsing through the EKB by graphical means

A real-life scenario for a browsing action would be a case where the manager of the workgroup "Better solutions" needs to discuss the stronger integration of the workgroup member "Corinna Otmann" with her unit manager from "ACME/Production". The two managers would have to negotiate the percentage of involvement of the employee in the workgroup or in line. Otherwise the workgroup manager would initiate a design change to the infrastructure model himself and remove Corinna from the project group. In the background of such a browsing, the respective entities (the organizational unit and the unit manager) are displayed graphically.

5.5.5 External Connectivity in GroupOrga

Besides the core modules of the architecture that were introduced so far, more integration-oriented components were conceptualized and realized. This includes implementing external connectivity for the framework. The status of the framework is the focus in the following discussion.

X.500/LDAP Conformity

The GroupOrga requirement list suggests information access to the X.500 directory. On a technical level, this requires instructions in the entity documents for the mapping of GEIMM entities onto objects in the X.500 directory structure. For that purpose, the corresponding X.500 object class name can be provided for each GEIMM entity type document. Such linking is then used to map information that is retrieved from a X.500 directory to the corresponding entity type in GEIMM and vice versa. With this method, a clear mapping and presentation of X.500 into a GEIMM model can be achieved.

The concrete solution for the GroupOrga EKB is to set up a mapping in a field in each entity type document that specifies `x-500_Object_Class`. For a person entity, this mapping reveals the value `organizationalPerson` (see Table 5-2).

A Domino server can be set up as a Lightweight Directory Access Protocol (LDAP) server. LDAP ([Kille 1996] and [Gordon 1997]) is an Internet protocol that lets any LDAP client (Netscape Communicator, Microsoft Internet Explorer, Soft-Switch Directory Explorer, and most importantly any LDAP-enabled application) search directories on any server running LDAP. Originally, it was designed to replace the X.500 DAP in order to make directory access less complex. According to Casonato from the GartnerGroup, this simple protocol will gain immense importance over X.500 ([GartnerGroup 1997a], p. 8). LDAP should be understood as a lighter version of the original X.500, since a LDAP client communicates to a LDAP server which may in turn use the X.500 protocol to access directory information. In the same way as X.500, LDAP defines a means for external clients to query and manage a repository database based on entries.

A combination of LDAP on the Domino server and the mapping functions of the EKB makes the repository information in the GroupOrga EKB available to browsers, Internet clients and LDAP-enabled applications. Thus, the Domino LDAP server uses the GroupOrga organization repository under the GEIMM model.

A standard LDAP client connecting anonymously can by default query and retrieve only a restricted number of fields, such as `FirstName`, `FullName`, `LastName`, `MailAddress`, `ShortName`, `Organization`, and `Organizational Unit` from person documents as well as `Name` and `Members` for workgroup documents. Figure 5-41 shows such a query of a standard Netscape Communicator client onto the GroupOrga EKB. The query result lists all members of the organizational unit *Sales*.

However, enhancing an LDAP client application beyond adding the results to the local address book or composing a message (as suggested in Figure 5-41), could include LDAP functionality in external WfMS or office management environments that are bases on a GroupOrga EKB.

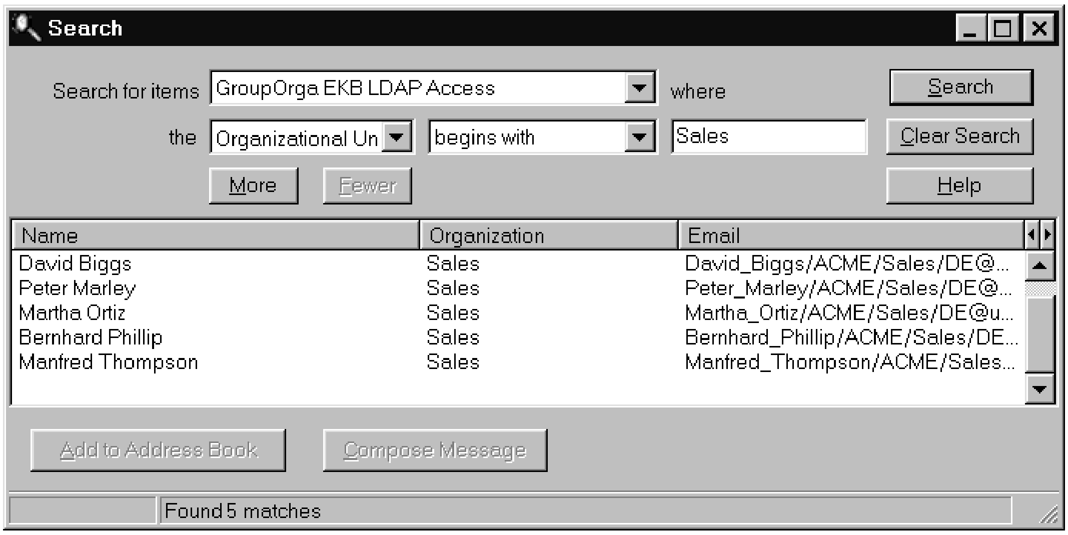


Figure 5-41: Standard Netscape Communicator LDAP client querying GroupOrga EKB

Integration with Non-GroupOrga Applications: GroupOrga Connector

The interface software GroupOrga Connector synchronizes infrastructure data from a GroupOrga EKB with that of other organization design applications. From the two concrete cases of integration with the GroupOrga framework via the GroupOrga Connector, the solution of the Siemens Nixdorf Informationssysteme AG's (SNI) product OIS V1.0 A00.05 is presented as an example.

The organizational information system OIS provides a component for unified user administration in the area of SNI's ComUnity concept (see [Siemens Nixdorf 1997b]). OIS can provide platform-independent and uniform user data to a set of applications. It is also used for administration and authentication of users in networks. Entities known to OIS are users, organizational hierarchies (with position, unit, role) and security aspects (in the form of competencies). The emphasis is on organization design, while security aspects are of minor importance. OIS offers an interface; however, it is restricted to applications which run in similar operating system environments.

The Organization Resource Model, the object model of the strictly relational and centrally administered OIS database, shows isomorphism with the GEIMM, which is why during a research cooperation the two applications were connected via GroupOrga Connector. For this reason, the two models were thoroughly examined and compared. For information on the comparison of these data models during the GroupOrga project, refer to [Huth 1998].

In general, the compared systems have similar data models but quite different forms of concrete representation for these models. In comparison to what characterizes the GroupOrga framework, the OIS system relies heavily on the relational scheme, both conceptually and technically. It is based on a rigid, normalized model that is difficult to configure or individualize.

At this point, a detailed description of specific technological characteristics is less important than the different concepts in data management, administration and storage. These fundamentally different paradigms put high demands on the interface and implementation, especially in comparison to Export-Import-Filters, OLE or other such technologies. More importantly, organizational contexts must be considered and a generic copying of scenarios stands in the foreground.

According to these considerations and the goal that interfaces should be generically useful, a concept and architecture that makes the integration of external design applications almost entirely configurable was devised. While the current realization is based on static assignment, the concept for GroupOrga Connector intends to make the synchronization completely configurable.

This interface falls back on a number of generic setting documents that describe how the mapping from one enterprise model into the other has to be executed. These documents are stored in the organization's EKB and present a set of parameters for transfer of enterprise model entities. The concept of a clearly defined and configurable interface description prevents it from being directly coded into the GroupOrga OrganizationModeler or any other tool within the framework. For of updates of the external environment, only one part of code is adapted, instead of adapting the many different applications in GroupOrga. Moreover, because of this architecture, the solution can easily be integrated into newly defined tools, such as the analyzer, in the project framework.

5.5.6 Graphical Organization Analysis

The regular or sporadic analysis of the current structural situation also needs to be considered for a complete framework. With an EKB and its archives as comprehensive sources of infrastructure data about present and past, an analysis can quickly be carried out by computational means. While the overall achievement would be a comparison of the current infrastructure with a nominal organization design, at its current stage the GroupOrga analysis is confined to the documentation and visualization of the analyzed infrastructure. For instance, it can be examined what form of subordination exists in an organization, how the employees are spread over the various hierarchical levels and within the workgroups, which roles and authorizations are occupied, and which ones are still necessary. (For more information on which aspects are and will be analyzed in the GroupOrga framework, refer to [Gieffers 1995].)

To judge the efficiency of organizational structures by analyzing the current infrastructure is a difficult task. This is especially the case, since there is another problem: What are the exact organizational goals and how can these goals be measured? General goals, such as *Make more profit* or *Better service for the customer*, are not direct indicators of how organizational structures should be laid out. Therefore, for the time being, the project offers means for documentation and analysis of infrastructures, but no recommendations about good or better infrastructures.

For the purpose of analysis, a general enterprise model provides two types of information: *latent variables* and *observable variables*. Latent variables are not directly visible in an enterprise model based on GEIMM. They are analyzed by documenting corresponding observable variables to which they, the latent variables, reference. For example, computer literacy may be thought of as a latent variable—it is not directly accessed with a single entity in the model. However, it may be judged by other entities that are observable and can be documented. In this case, the corresponding observable entity is that of knowledge/skill with instances such as *word processing proficiency* or knowledge of a *programming language*.

But if the information used by the enterprise model is different from the information used to run the actual organization, any analysis of observable variables in the model is of dubious value. This is a great advantage of the GroupOrga analysis in contrast to most other analysis applications. Due to its continuous and participative modeling approach, it analyzes data provided directly from the working environment. No time consuming and error prone process of supplying data to the analysis tool is necessary.

The GroupOrga analysis aims for :

- Transparency about the degree of hierarchical structuring and division of work
- Transparency about the current distribution of role assignments to employees and thus the distribution of functions within the organization
- Identification of the degree of teamwork and networked interdependencies (in contrast to hierarchy)

The result of an exemplary role analysis shown in Figure 5-42, reveals that two roles in the enterprise (model) are currently not referenced. This may be solved later by querying the EKB for employees with knowledge/skill in Asian and South American languages and the products of the Asian and South American markets.

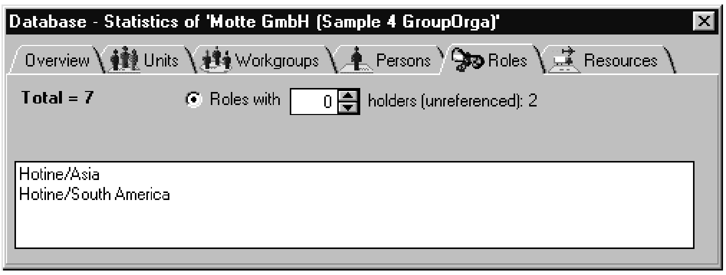


Figure 5-42: Result of role analysis

From the GroupOrga point of view, there is more significance in analyzing the informal interdependencies between the employees and within workgroups and knowledge groups than in the hierarchical structure. This difference also reflects the shift away from subordination towards networking, and must be the focus of further analysis features in the framework (section 7.3.2). Moreover, a documentation (and analysis) of the organizational situation cannot be the final aim. Its results must be the starting point of another organization design cycle (chapter 6).

5.5.7 New Concepts for Graphical Process Modeling

In addition to the infrastructure oriented view supported by the tools presented thus far, the GroupOrga project introduced two more modules to cover the threefold architecture of the GEIMM. The process modeler briefly presented here provides additional functionality to a previous version (see [Ott 1994]) for graphic-oriented process modeling.

As far as the GroupOrga framework is concerned, the overall concept of process modeling, its tool support and related topics is found in [Ott 1994]. [Meyer 1995] provides detailed information regarding the new concepts in this section. References on process modeling and WfM of any kind are too vast to mention here. An overview can be found in [Georgakopoulos/Hornick/Sheth 1995]. The process modeler fulfills the following requirements:

- It uses formal syntax and semantics based on GEIMM
- It is based on intuitive concepts
- Is has a visual user interface
- It has an enactable result (i.e. process model)
- It is able to express a variety of processes that can be any combination of formal to informal and automatic to manual
- It is able to express concurrent, parallel and sequential routing
- It supports comprehensive abstraction and clustering
- It is able to support dynamic change of process descriptions

Requirements 1 - 3 are important in the context of this project. They ensure that the form of specifying the process model part of the GEIMM is easy to learn and handle. A modeling tool with formal syntax and semantics has the consistency that is necessary for an EER-based model such as the GEIMM. Similar to the EKB, the process repository is a groupware database which follows a different architecture than that of purely relational database management systems. Like the GroupOrga OrganizationModeler, the process modeler ensures a clear and understandable result of the continuous modeling process within an organization. The *visual* requirement means that the formal properties of the enterprise model are hidden from the end user. Hence, one does not have to be a process modeling expert to use the application.

Requirement 4 focuses on the fact that *living* descriptions must be created, which have enactment semantics and can be exchanged with an enactment environment. Requirements 5 - 8 are necessary to support the description and enactment of real world processes that consist of a mix of process steps that are informal, formal, and automated, and are interconnected.

One of the novelties realized in the process modeler during the project was a conceptual and practical approach of how different processes are performed similarly and how these examples can be organized in an on-line *process handbook* or library. The handbook is intended to help redesign existing organizational processes, invent new organizational processes, and learn about organizations. It was very important that the process models in the library actually worked in real life, since processes can always work on a drawing board. Figure 5-43 shows a workflow scenario where templates or operational workflows have been copied or linked from a process handbook into another operational process model.

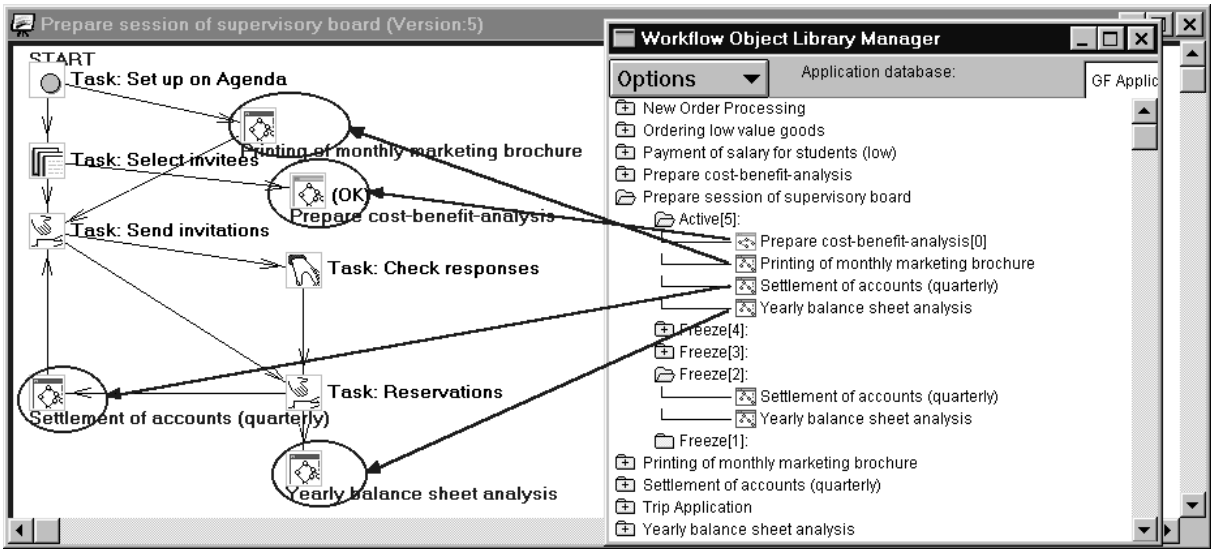


Figure 5-43: Using a process handbook in the process modeler

Since process models that are handled in WfM environments grow larger over time, a means for abstraction was added to the GEIMM process modeler during its GroupOrga development phase. In this project, abstraction is the process of temporarily suppressing irrelevant details to establish a simplified model. The clustered nature of the proposed modeling construct supports the decomposition of a single business process into multiple layers of sub-process. Each business process is made up of lower level processes that support bottom-up and top-down approaches. In Figure 5-43, the three linked workflows (*Printing*, *Accounts*, and *Balance Sheet Analysis*) as well as the copied *Cost-Benefit Analysis* example are represented as clusters in the original workflow instead of a complex sets of single tasks, indicated by an appropriate icon.

5.5.8 A Graphical Information Modeler for Groupware Applications

In order to optimize the whole range of business aspects reflected in the GEIMM, it is also necessary to take a closer look at information objects and applications. Therefore, the third partial model of the GEIMM, the information model, has an early prototype of a graphical information object modeler in the GroupOrga project. Its development was conceptualized and carried out in cooperation with a parallel research project. The information modeler aims at performing information and application modeling on top of the groupware platform Lotus

Notes. For more information on the information object modeler, refer to [Liebrand 1995] and [Hinrichs 1996].

This information object modeler uses the GEIMM meta-model to reflect the information structure of any Lotus Notes application. This part of the model is visualized by the graphical user interface of the information modeler which generates new information objects and applications or modifies existing applications from newly created or modified designs of the information model. The design process regarding the information model is supported by a data dictionary that allows the user to reuse standardized components. During this early sub-project, an object-oriented API and several controls for the graphical user interface were discussed in coordination with other research fields within the WfM area.

A consideration of different API approaches to the Lotus Notes information architecture showed that, with the exception of the Notes API, there is no tool that supports the design of the information model to the extent needed for the end user.

Hence, the structure of the GEIMM information model and the Notes database system was the basis for the development of the interface software AccessNotes, shown in Figure 5-44 (see [Liebrand 1995]). This object-oriented interface was then implemented into a graphical user front-end to support easy construction of the information model and of Notes applications based on the concrete model.

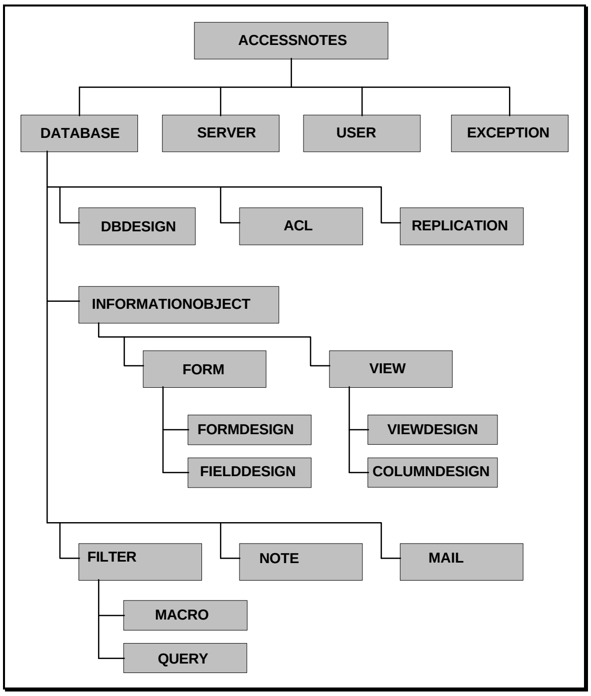


Figure 5-44: Class Hierarchy of AccessNotes

As with all GroupOrga tools, the information object modeler puts a strong focus on visualization and direct manipulation. A user-friendly desktop application allows the user to define, move, modify or delete information objects that may be used in a workflow or office management application as specified in the description of the information model in chapter 4.

5.6 Integration of GroupOrga with ESPRESSO, GroupOffice and GroupProject

Support of office and workflow management systems is a central aim of the GroupOrga framework and its applications presented in this chapter. This concluding section introduces the realization of the support for particular office and workflow cooperation support systems, namely ESPRESSO and GroupOffice (Pavone Informationssysteme GmbH). The integration with the project management support environment GroupProject (see [Ehlers 1997]) is also mentioned here. This integration was conceptualized and partly realized in the GroupOrga project and in cooperation with the Pavone Informationssysteme GmbH. Since ESPRESSO and GroupOffice were introduced in chapter 2, only GroupProject is introduced here.

Task descriptions in the two process-oriented applications represent an abstract specification of how a task can be carried out in the enterprise during a process execution. The term *abstract* indicates that the task description, for example, does not refer to any particular employee of the organization and that the binding of a task to an actor is declared by generic expressions. For instance, the responsible actor that carries out a task is described as an entity from the GEIMM-based organization model or is generically defined by the GroupOrga organizational query language. So, in general, the GroupOrga framework provides its services to these applications via its various service-interfaces from the organization repository.

The GroupProject system ([Ehlers 1997]) integrates traditional project management software, such as CA-SuperProject (Computer Associates) and MS Project (Microsoft), with the groupware platform Lotus Notes. By doing this, it combines the advantages of these different application types into a comprehensive environment for flexible and distributed project management. As with every project management system, GroupProject necessitates a resource database from which resources (human, financial or material) can be assigned to projects and tasks. The GroupOrga EKB provides modeling functionality to the GroupProject system by offering this required resource information from the organizational repository.

In conclusion, Figure 5-45 clarifies how the GroupOrga system can be the base technology for various kinds of office-oriented applications with the example of its integration with the product range of the Groupware Competence Center Paderborn (Pavone Informationssysteme GmbH and the Institute of Business Computing, University of Paderborn).

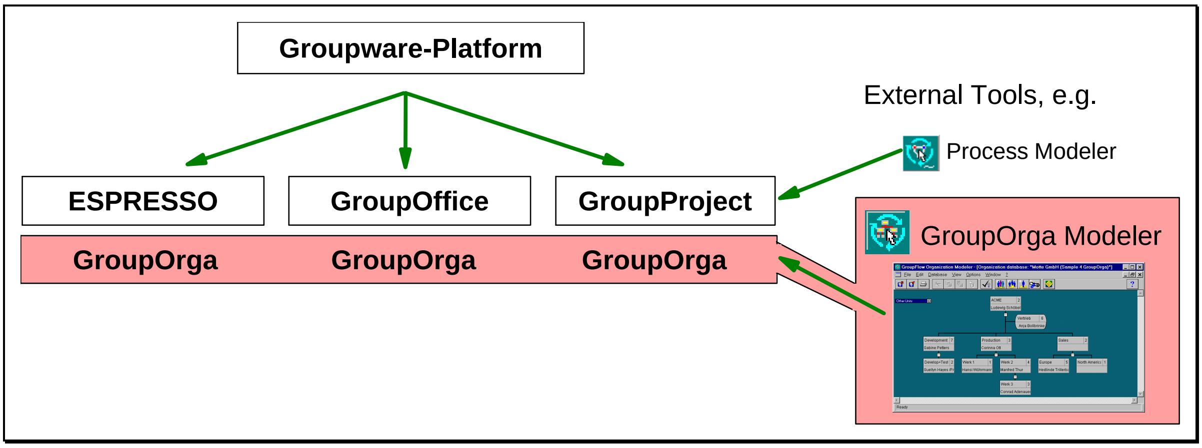


Figure 5-45: GroupOrga as the base for workflow, office and project management products

Chapter 6
A Meta-Process for Groupware-supported Organization Design

This chapter introduces a meta-process for organization design. It explains how organization design is carried out in the GroupOrga vision, outlines the stages in which particular parts of the toolset can be put to use and, building on the focus of chapter 5–the concrete system and its tools–turns the original approach of traditional organizational methodology into a systemic approach.

With this meta-process, chapter 6 presents a design methodology that integrates design techniques and tools to assist the participating organization designer. A methodology usually depicts: (a) the activities to be performed, (b) the relationships and sequence of these activities, and (c) the various evaluation milestones to be reached. Thus, a methodology is a comprehensive procedural framework directed towards accomplishing a particular change in the object system. In this case, the object system was identified earlier as the organizational model being developed.

Hirschheim [1985] notes that a methodology is conceived between *vision* and *technique*. *Vision* is seen in this context as a non-specific and broad indication of action. *Technique*, in contrast, is a precise and specific program of action that produces a standard result. Hence, this methodology is a firmer guide to the action of organization design than the vision laid out in chapter 5, and it also tells the "how to" by introducing the GroupOrga tools and their techniques into the methodology.

By presenting two types of nested design cycles in the methodology, section 6.1 more clearly distinguishes between the "what" of the visionary evolutionary design and the "how" of the precise, tool-centered life-cycle. Section 6.2 presents the four steps of a change-organization process, and section 6.3 concludes with summaries of case studies from GroupOrga.

6.1 An Ongoing Design Process

The organization design process is variable and thus adaptable to concrete situations. In order to clearly distinguish between it and common models of software-design, the terms *phase* and *phase-model* are avoided in favor of *steps* in the process. Due to the complexity of the problem, it is advisable to fully exploit the steps of the organization design process. In other words, the better the results of the initiation step, the better the concept and outcome of the transformation.

The GroupOrga vision favors an ongoing and evolutionary design process. But in such an approach, time is also a critical factor; therefore, the steps must be able to flow into each other. Simultaneous and recursive procedures are necessary as well. This design process covers all angles of organization design, considering anticipation and feedback, as well as the possibility of correcting previously made decisions.

In fact, the GroupOrga meta-process consists of two nested design and redesign cycles. Due to its nature as a participative and evolutionary approach, it would be inappropriate to solely define a traditional life-cycle model for the whole evolutionary design process ([Conroy III 1986] and [Heinzl/König 1993] or the algebraic approach of [Baligh/Damon 1980]). Such an *automated* approach would specify a defined start of a design process as well as a defined end for the whole enterprise. Moreover, for *every* organizational member this approach would have to be in the *analysis phase* at one point in time, then move on to the *design phase*, later to the *implementation* and *testing phases*, and so on. Such a centrally stipulated approach would imply a planned and strictly guided procedure that lasts for a certain period of time and is considered to be finished afterwards.

In contrast, in GroupOrga a life-cycle concept is nested into an evolutionary procedure without a precisely defined start and end. The participative organizational members or groups follow the redesign life-cycle that is presented in sections 6.1 through 6.2. However, at any point in time, a variety of such redesign cycles may be in existence in different organizational locations and groupings, each in a different stage of completion. In other words, while a particular group is finishing its internal organization design, another group may be starting, and yet another is in the midst of its participative design process. Thus, the organization is in constant flux and adapts to environmental changes when necessary.

Hence, the GroupOrga project suggests a nested methodology that lays out the methods and necessary steps by which enterprises can reshape their organization design. One part (the *outer* organization design process) describes and supports the creation of the overall vision in order to successfully drive innovative structural design efforts. The other part (the *inner* life-cycle) supports the design of very specific details which are directly translated into the enterprise knowledge base. It enacts definitive action in order to implement a new organizational substructure or to deploy specific changes.

Sections 6.1.1 and 6.1.2 explain these two types of redesign cycles. Then the discussion returns to the broader view, and steps of the overall procedure are examined.

6.1.1 Developing the Model - A Long-term Procedural Approach

The methodology to develop the model is a matching of the enterprise's organization design criteria against different organizational forms, and thus progressively refining the selected form at multiple points until a temporarily satisfactory closure between criteria and characteristics of the organization is attained. In order to gain a complete model of an organization, the tools and modeling elements from the preceding chapters have to be used for developing a model of the processes, a model of the information objects and a model of the infrastructure. The methodology presented here arranges the steps to be carried out. In contrast to other such approaches, the order of the modeling tasks (i.e. infrastructure design, process and information modeling) is not necessarily sequential. Rather, it is a parallel procedure of process modeling and infrastructure design at the same time. A modeling result that meets the requirement of an integration of workflow and organization design IT is thereby ensured (see section 5.1.2).

Some approaches to organization design distinguish between two types of modeling: that of simply depicting an as-is-infrastructure and that of designing a future to-be structure. In the first case, these approaches advise to start with the infrastructure design, since such a documentation is mostly existent in an organization and simplifies the entry into the modeling. In the latter case, it is suggested to start with the critical processes and later adapt the infrastructure in a fitting manner. For GroupOrga, no such decision must be taken, since it is intended as an ongoing process where process modeling and infrastructure design coincide and influence each other. Only when first realizing this methodology, it has to be decided whether a radical reengineering of the processes is desired or if a gradual progression is the better choice. In the latter case it is recommendable to start with the structural design.

The infrastructure model can be set up top-down or bottom-up. This design procedure can be applied to change at any level of granularity. If the procedure is applied on a coarse level, this can be referred to as the top-down strategy from section 5.2.2. Moreover, this implies that the design methodology must be used recursively until the finest level of granularity has been reached by means of decomposition. The GroupOrga vision also acknowledges that organizational change can begin in an organizational subunit or a workgroup and that such grass-roots design can influence the whole enterprise. Consequently, regardless of the type of design initiation, several design procedures will be underway at different locations and under various forms of supervision. This corresponds with the results from section 5.2.2, where a *middle-out* strategy was advocated, which links the top-down design with the bottom-up modeling approach.

The overall idea of this modeling approach is (similar to [Rein 1992]) that the participative design process brings about organizations that are self-organizing systems. This procedure is a conscious, ongoing problem-solving and planning-process. It results in an organization design that is in constant motion and that is continuously developing and changing.

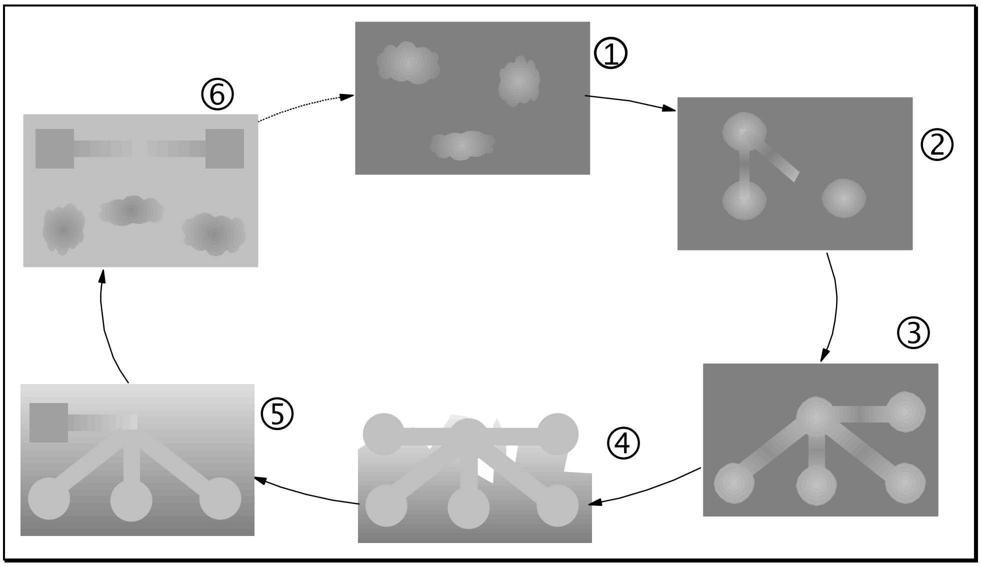


Figure 6-1: Outer GroupOrga organization design process

Figure 6-1 depicts this perpetual process of design and structuring as a form of a constantly evolving organizational infrastructure. During this process new infrastructure islands may evolve inside the existing design (1). Over time, such informal islands stabilize and are formalized in order to become fixed and accepted components in this alternatively structured environment (2). While these new components grow and manifest themselves (3), other older structures dissolve and vanish (4)—later in this stage it becomes difficult to distinguish the old from the new (5). Some of the old structures gradually break up into islands in the new organization design (6), but not all of them will necessarily change. It is more likely that some structures will remain stable and coincide with newer ones, while others dissolve completely (state 1 in a new cycle).

The described process involves everyone who shows interest in the organization design, and each participant can decide how much integration into the process is wanted. The larger the number of participants, the better the result will eventually be.

This *outer* organization design process can be referred to as global BPR. Among other steps, this includes rather global tasks, such as problem identification and the creation of divergence. Based on this, design alternatives are to be compared and evaluated before implementation takes place. A reshaped organizational structure is then used as the starting point for the transition into a continuous improvement environment that enables recurrent refinements in order to cope with required organizational changes.

Before going into detail on the various overall stages of such an ongoing process in section 6.2, the following section will more closely examine where the various GroupOrga tools come into use during the bottom-level design processes of the *inner* life-cycle.

6.1.2 A Tool-supported Life-Cycle for Organization Design

The *inner* organizational life-cycle as shown in Figure 6-2 reflects a more concrete endeavor to set up new or redesigned bottom-level organizational structures based on a set of design tools. Its intention is to integrate the presented tools and to explain their practical use in the participative design approach.

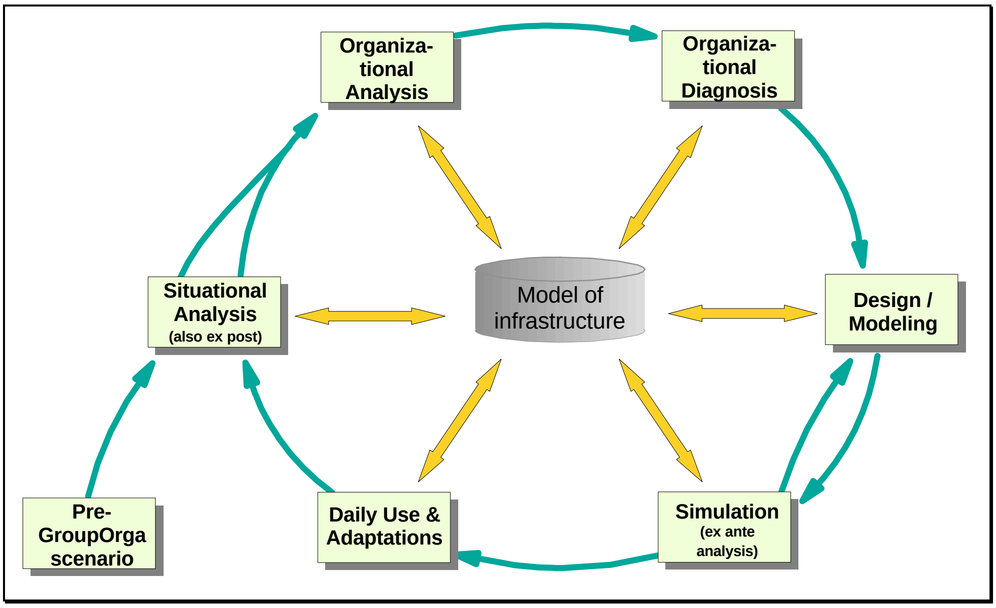


Figure 6-2: Inner organization design life-cycle

There are various entries into this cycle that are to be selected based on the particular scenario of an organization in question. Existing organizational structures may have evolved in an enterprise scenario over time in operative databases according to the actual task performance or based on an "organizer's" advice. Such existing structures may show symptoms of inconsistency or inefficiency that cause the employees to initiate a redesign of their particular working situation. Hence, these structures can be analyzed, as suggested in section 5.5.6, by means of the generic and graphical infrastructure analyzer tool GroupOrga OrganizationAnalyzer in order to find the reasons for the symptoms. This tool provides a graphical representation of the analytical findings from different viewpoints. It enables to gain a structural understanding of the current situation by aggregating detailed information which is otherwise somewhat hidden in numerous entity documents in an EKB.

The selected and sorted out results of the organizational analysis cycles may be diagnosed and refined in order to be used as input for the design of refined organizational structures. Such an organizational diagnosis provides the base for a better understanding of the social organizational system. But although both, the EKB itself and the GroupOrga

OrganizationAnalyzer provide with statistical data from its daily use and from the analysis, it is not intended to give computed reasoning of how to structure an organization. This decision is supported by the GroupOrga tool set but the affected employees will not be overruled by the system.

During redesign, the supporting tool is the GroupOrga OrganizationModeler. It is now used on the various technical platforms and for different user types in form of a comprehensive tool or as a Java implementation. Depending on the degree of the symptoms found in the analysis, the redesign has a rather revolutionary or rather evolutionary character. In the GroupOrga vision the latter will more likely be the case since even small inefficiencies are eliminated in the ongoing process which makes a drastic, revolutionary redesign somewhat improbable.

Before the derived structural specifications in a model are subsequently used in order to support the workflow interaction they may be tested and evaluated. In the GroupOrga approach, this can be achieved within the context of an interactive and graphical simulation environment. Potential inconsistencies in the model can be avoided before they actually occur in the organizational model. Such improvements can be accomplished by direct changes to the structural layout in an interactive fashion. This part of the organizational life-cycle is not yet realized as a prototype application and is thus introduced in the concluding chapter. The refinement procedures for the bottom-level organization design and the accompanied testing via simulation will usually be recurrently applied until an appropriate design has *ex ante* been determined, which will then provide the base for the following utilization of the model.

In parallel to these dedicated life-cycle steps, the participating employees will use the groupware EKB and its query and navigation tools to support their daily work. Occasionally while they do so, minor adaptations to the organizational model may become necessary and if their access rights allow it, they will carry out these modifications in their own working environment on-the-fly. The users may continuously validate the organizational model and correct errors interactively. So, during this utilization stage, manifold modifications must be tracked which will become important in an adjacent stage of post-evaluation.

Organizational structures may thus later be reviewed again or *monitored* in multiple *ex post* analyses in order to be reflected against the *ex ante* planned situation using the same methodology and GroupOrga tools. The results will seamlessly be used as the basis for the ongoing redesign effort within the architecture which supports continuous improvement and refinement. Here the inner life-cycle is closed in form of an iterative procedure.

All steps in the described life-cycle have a more or less intense connection to one of the GroupOrga tools and they are all based on the organizational repository, i.e. the EKB (see Figure 6-2). Again, the use of this repository ensures, that the real organizational structure is not different from the newly modeled and optimized one. The relationship between the various steps within the two nested parts of the organization design methodology is to be further refined. Some steps have a rather loose relationship with those on the other cycle. But there are steps in both cycles that bear a close association. However, they are not completely equivalent as in general the stages on the outer process tend to have a broader scope.

6.2 Four Steps of a Change Organization Process

This section describes what general stages will continuously be run through in a process of organization design.

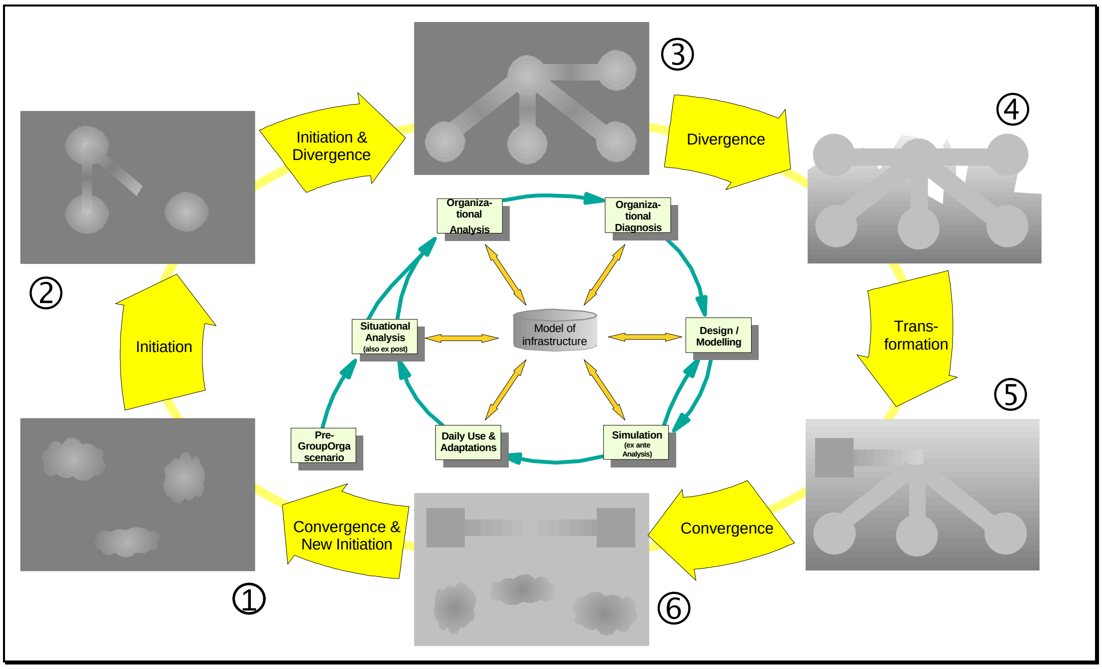


Figure 6-3: Nested organization design process and inner life-cycle

The outer organization design process depicted in Figure 6-3 is initially specified by organizational members of upper management levels. The process is started top-down and at this point the organization is now conducting its evolutionary modeling process according to the procedure explained above. As a next step, the participating designers choose to either change parts of the organizational structure or to carry on with the current situation.

Such a design process has slightly distinctive stages, such as initiation, divergence, transformation, and convergence. Each such stage produces some output which in turn is the input for the next stage in the process. For example, in the initiation stage a brief may either be officially set up or implicitly be agreed upon of what to focus on in the modeling process. It would not make sense to enter the transformation stages with the members of a workgroup or a unit, if no preliminary statement of the proposed modeling has been agreed on. The stages generally occur in the order named above, although at a later stage in the design process several such modeling processes may have been started and they may overlap at a certain point in time. Figure 6-4 visualizes this overlapping at an imaginary point of time t_3. In this graphic various design teams are in the midst of their participative design at different stages of the life-cycle. At time t_3 the workgroup *Project South* is currently conducting the organizational analysis, the group of role holders *Sign contract* are in the midst of the situational analysis while the skill group *C++ Prgr.* is redesigning its own setup and structure. The members of the unit *Sales* are already simulating a new design to put it into practice soon. This simple example shows how different participants in the design process can be engaged in different design steps at the same time.

In fact, in the transformation and convergence stages, sub-structures are better understood and modeled by the employees on the various levels. These modeling procedures of sub-structures are then dealt with by going through all steps of the inner life-cycle . This pattern of ongoing modeling can be visualized as an infinite nesting of the design process which at ever lower levels examines further details of the design, as shown in Figure 6-4. With such an approach changes in organizational structures emerge from an iterative series of modifications. Each such modification may be incremental, but all of them add up to strategic redirection.

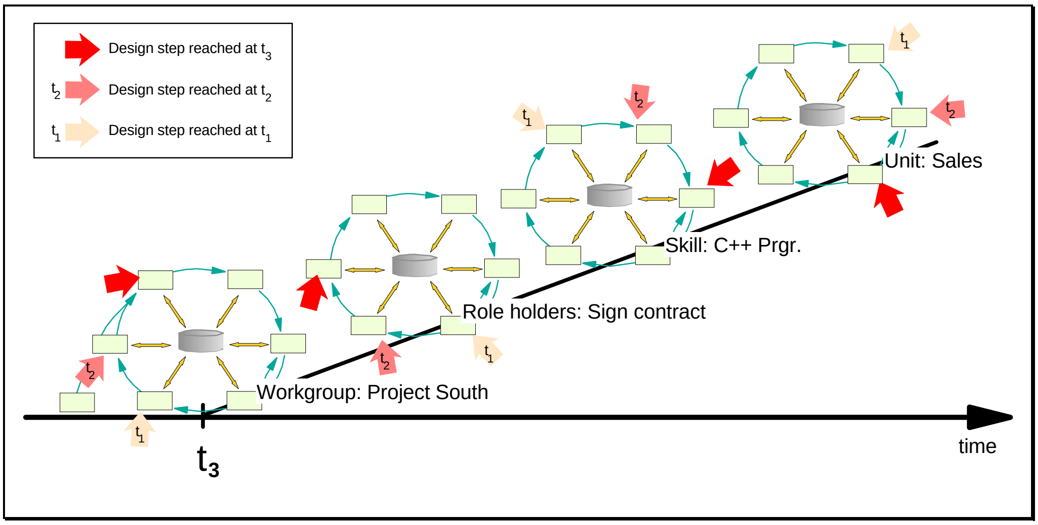


Figure 6-4: Looped cycles of organization design stages

Initiation of change

This initial step is mostly driven from external stimulation. Here the participators implicitly agree on a brief statement about where the inconsistencies in the current structure were found and about what goal to reach with the mutual redesign. Some informal discussion between those who are affected may occur until further informal support is mobilized and the minor structural changes are about to be tackled. At this level of the modeling process there is a *sponsor* of the redesign whose backing ensures that the modification will be endorsed in the organization. A *stakeholder* is someone who is not directly involved in the organization, but is significantly affected by it. In an example, the members of a midsize organizational unit would want to restructure their workgroup patterns in order to meet some ISO 9000 reporting requirements and have thus a higher value as internal profit center. In such a case, the stakeholder might be the ISO auditor whose interest is to see that the unit meets the reporting requirements, while the manager of the organizational unit would be the sponsor of the design process.

Creating divergence in the existing structures

The purpose of such a stage is to establish a common understanding of the operations of the enterprise and to determine the assignment of labor. This stage gives time to discover what in the organizational structure is susceptible to modification and what parts are to be understood as fixed. Thus, this phase is fact finding, analyzing and diagnosis. The goal is to start to resolve the existing sub-structures. In this stage, the structural boundary is unstable and undefined.

Organizational transformation from old to new

The transformation stage is the pattern making stage. It can occur unexpectedly in an organization, but it will always follow a divergence stage (see Figure 6-3). In this stage, a new pattern evolves on the results from the divergence stage—it is the design and development of a changed model of the enterprise. This structure is precise enough to later permit convergence to a single and consistent structure in every detail. The activity is that of turning a number of complicated restructures into a consolidated form again. Therefore, the overall organization design problem is split into several sub-tasks, each of which is tackled by itself. The graphical design tool GroupOrga OrganizationModeler can be considered a transformation support tool—it helps people work together and blend their pieces into a group-based solution.

Returning into a new state: convergence

The convergence stage can be understood as a temporary implementation. Its task is to consolidate the different structural modifications into the organizational model and then into the EKB. In other words, it is the creation of representations for the entities and relations of the particular enterprise being modeled. Problems concerning the participation, (e.g. input conventions and the like) are solved and the result is a well-integrated and consistent knowledge base. However, this state of the model is only temporary, since after this step is completed, the model returns to the initiation stage. The next run will modify the organizational model again. But due to its never-ending nature, no clear split between the finish of one run-through and the start of another can be distinguished. While the structural modeling may have just finished at one location, it may have just started at another location.

6.3 GroupOrga in Field Studies on Organizational Change

A portion of the research in the GroupOrga project was on the analysis of various business enterprises that acted as external partners to the research in order to identify and discuss critical components for the vision and framework. Another main goal of the user involvement was to find out whether the tools from the prototype system addressed the problems of distributed work management for organizational modeling, and what might improve their design. Some of these cases were examples of already effective forms of flexible and evolutionary organizations, and others were examples of organizational structures where no such concepts had yet been discussed. This section shows which characteristics of the GroupOrga approach are represented by the cases which are drawn from the project's field studies and it contains a descriptive summary of each case.

6.3.1 Field Study 1: Deutsche Bank AG

The case: A large-scale international banking environment

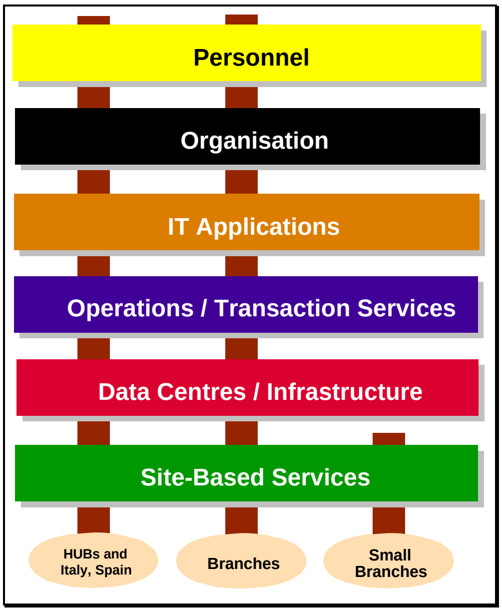


Figure 6-5: Layer-model of the Deutsche Bank AG divisions

The domain of this GroupOrga field study (see [Brunner in prep.]) is large-scale international banking environment, in this case, the Deutsche Bank AG's division "Transaction Services". The Deutsche Bank Group currently consists of five high-level divisions (as of May 21, 1998), namely "Retail and Private Banking", "Corporates and Real Estate", "Global Corporates and Institutions", "Asset Management", and "Transaction Services" and each of these divisions is managed by at least four divisional heads (see [Lilienthal 1998] and [Lilienthal/Bonk 1998]). Besides these divisions, ten staff departments exist, each being headed by its own manager and two *regional CEO's* who are responsible for the overseas branches. The divisional heads, the advisory managers and the two CEO's report directly to the Deutsche Bank AG's board of directors which currently consists of ten members of the board. In the course of a complete structural reformation the former division OuB ("Organisation und Betrieb") had been restructured an then integrated in form of the newly established division "Transaction Services". This division represents the complete back-office infrastructure of the Deutsche Bank Group and consists of six levels as pictured in Figure 6-5.

It has to be noted that the presented Six-Layer-Model is valid world-wide for the Deutsche Bank AG, and thus not only for the structures of the head office in Frankfurt/Eschborn.

Case study results

The overall, fundamental organizational structures and all steps to restructure single divisions are centrally initiated by the board of directors or by the individual divisional heads. The responsibility for the realization of these targets in the individual divisions lies with the globally and regionally responsible persons, i.e. the *global heads* or the *regional heads*. Besides these modifications of the overall macro-structure, numerous world-wide modifications occur on the micro-level, as well. For such restructuring to become documented, it must be first checked and then approved. For this process within the division

"Konzerndienste" a central authority has been nominated, specifically the department *organization* in Frankfurt/Eschborn (see Figure 6-5). This unit has six organizational subunits which are illustrated in Figure 6-6.

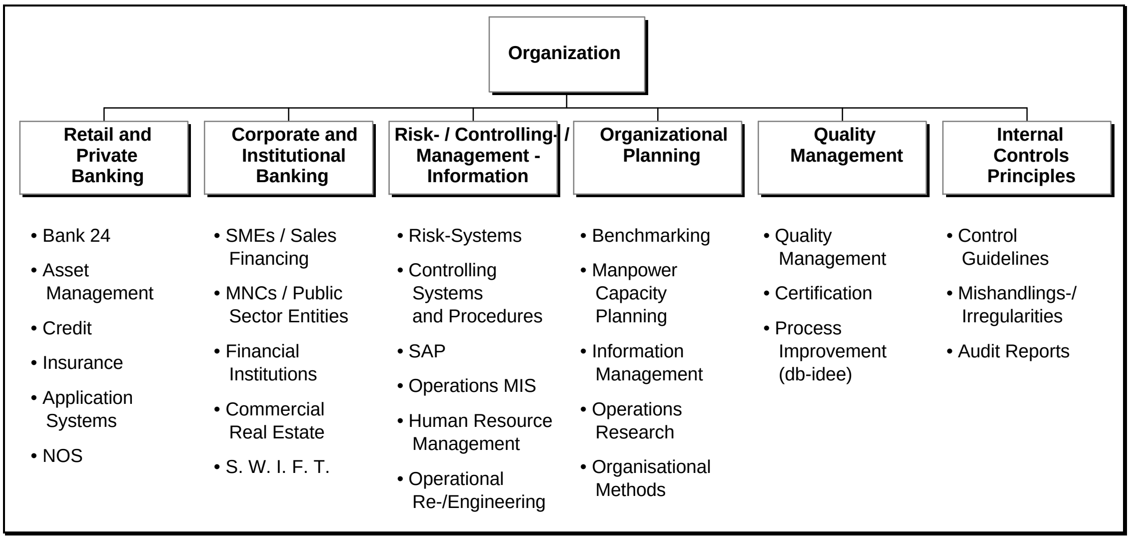


Figure 6-6: Unit "Organization" of division "Konzerndienste"

The central "Evidenzstelle" (the organizational units responsible for approving structural changes) are the subunits "Risk-/Controlling-/Management-Information" and "Organisational Planning". All modifications requested by lower level units must be coordinated by the administrator at this central position. The administrator can judge in three ways:

- If the modification request violates an overall macro-structuring rule stipulated by the board of directors, it is rejected and the requesting unit is notified.
- If the request conforms with the overall macro-structuring rules and if it does not affect profound decisions in the AG, the modification is accepted and saved for later documentation.
- If the requested modification is of great importance to the Deutsche Bank AG and if the "Evidenzstelle" cannot decide on it, the request (but not the modification) is accepted and forwarded to the responsible people (board of directors, divisional head, global head) for authorization.

Besides evaluating the requests for modification, the two organizational units responsible for the approval are also responsible for the documentation of every change. This process is currently structured as follows: Every month by means of e-mail or fax the "Evidenzstelle" actively asks every organizational unit to report modifications in their local organizational structure to the local authority. The reply is in form of organizational charts which have individually been compiled in the local branch offices and which contain the structural changes that have been undergone in the last month. In addition to this organizational chart each branch office creates a word processing file with the employees' names, telephone and fax numbers. The files (chart and employee list) are then sent to Frankfurt/Eschborn via e-mail or by floppy disk.

The locally performed structural changes are now evaluated in accordance with the process sketched above. Afterwards all approved modifications are manually transferred into another form of organizational chart, which is locally administered by the "Evidenzstelle" in form of various business graphics application files for each branch. In addition to the graphics files, the word processing files are checked and saved, as well. Together, the files are sent to the DTP (Desktop Publishing) group which in turn creates a setting copy. Finally, galley proof is checked by the "Evidenzstelle" and then send to the printing.

Besides a quarterly publication in print, the updated files (MS PowerPoint and MS Word for Windows) are also stored in a Lotus Notes Database in the bank's intranet. This allows the employees to download the updated organizational charts as well as the telephone and fax lists from an *OrgChart Application* on the division's intranet homepage. The single graphical files are linked by hyperlinks which allows for a restricted leafing through the charts.

Discussion

A homogeneous and Deutsche Bank-wide rule of organizational modeling and documentation of structures does not exist. Usually, each division has developed its own concept for a documentation and modification of its organizational structures. For example, a different approach has been chosen for the regional subsidiaries which are subordinated to "Privat- und Geschäftskunden". In this division one of the ten advisory departments is responsible for the documentation, and the results are internally published in form of a comprehensive organizational (hand)book. Modifications on the lower hierarchical levels only require the approval of the respective subsidiary, which in turn reports changes to the advisory department.

From an investigation into the GroupOrga characteristics realized (or not realized) in this field study, the following observations can be made. In the described case no integration between organizational modeling and any kind of operative application, e.g. workflow or office management, can be found. The term *OrgChart Application* is somewhat misleading. In addition, the organizational modeling in the division "Konzerndienste" does not rely on an enterprise model, in fact the only organizational entities that are modeled are that of organizational units and employees. The modeling process is somewhat computer-supported, however no dedicated organization design application but graphics software is used instead. When it comes to the participative and distributed modeling of the organizational structure, great steps have been taken already. Since the various distributed branch offices and their employees are asked to document and report their own organizational structure in a monthly procedure, an evolutionary modeling process could be a thinkable goal in this enterprise. Moreover, if the Lotus Notes database in the bank's intranet would not be used as a file storage only, but for an active distributed documentation and design approach in form of the proposed EKB, various GroupOrga concepts could be turned into reality here. In fact, during a cooperation project with the Deutsche Bank AG several aspects of the GroupOrga approach have been suggested and partly realized. This suggested framework is documented in [Brunner in prep.].

6.3.2 Field Study 2: agens Consulting GmbH

The case: A small-size management and IT consultancy

agens Consulting GmbH was established as a *society for business-oriented information processing* ("Sozietät für betriebswirtschaftliche Informationsverarbeitung") in 1977. It originated from a notary's office which explains the remnant "Sozietät" in the company's title. The core of agens' strategy is an international network of smaller, integrated and mutually complementing service organizations. The main offering of this network are business-oriented consulting services in various business fields and in organizational and IT-oriented respects. The network shows the significant characteristic that no partner owns a controlling interest in the partnership which hence fosters each partner's entrepreneurial skills.

The following Figure 6-7, which is copied from the organization's quality handbook, shows the partners of the network.

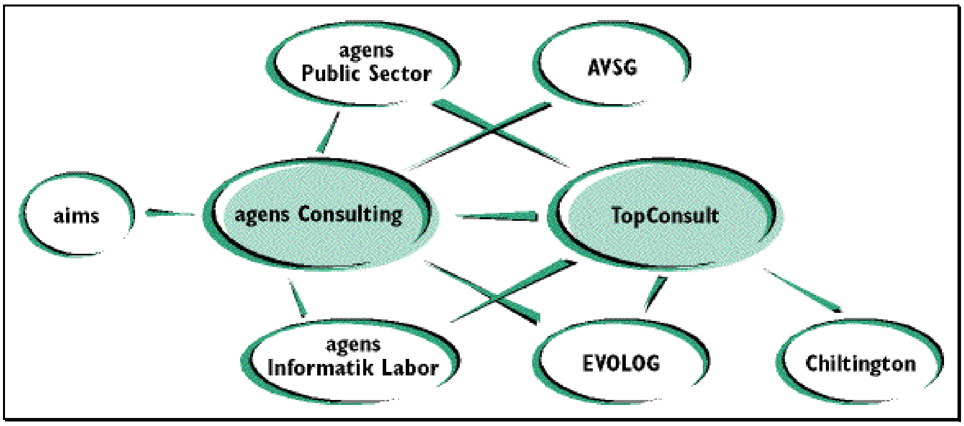


Figure 6-7: The agens Consulting GmbH network

agens' services concentrate on companies in the financial market, such as insurance companies, banks, reinsurance companies, building societies, and administrative authorities. In the technological sector, agens's core competencies are object-oriented programming, management of complex projects, multimedia and Internet, and distributed communication with groupware.

Case study results

The organization's self-given term for its form, a "society", illustrates that agens has no (or hardly any) hierarchical subordination and that organizational practices are considered different from tradition. Its infrastructure basically consists of two levels (or groupings of employees): the management and specialist solicitors ("Fachanwälte").

The *management* consists of *managers* and *managing partners*. Together with the *guiding partners*, the management forms the *enhanced management*. The idea of a society is reflected through the *specialist solicitors*. Each manager and guiding partner has—as a specialist solicitor—the full responsibility for the own field of action.

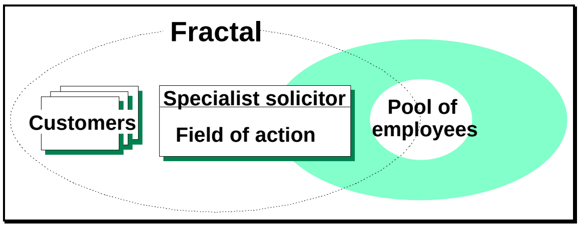


Figure 6-8: The fractal model at agens

The organization's structure is mainly characterized through its *fractal concept*. In this case, fractals are considered to be self-organized, recursive structures of an organization. When initiating a project, each specialist solicitor creates such a fractal for the project and its management. Consultants are not assigned to a particular fractal for a long term, but their are employed in the project or for single tasks at relative short notice according to their personal skills and their expertise. This selection of partners occurs in respect to the project, and in doing so the specialist solicitor sources from the entire pool of agens employees.

When composing a team for a project, the partners are selected in compliance with their qualifications based on a theoretical analysis model, the brain-dominance-instrument (HDI, Hirn-Dominanz-Instrument). A HDI profile gives record of a person's analytical, organizing, social, and creative abilities and preferences.

While agens operates numerous tools and technologies for their projects, such as Lotus Notes, project management applications, and controlling tools, the project managers and the enhanced management determine and appoint their partners by means of a paper-based handbook which contains all information about their technical competencies, their fields of action, and their HDI profile.

The aforementioned quality handbook is a pure Lotus Notes groupware application which is used as the container for process descriptions, structures, and personal profiles. This application is the main source for newcomers at agens to find out about their future workplace. In addition to this quality handbook, agens maintains a partner database application which informs about the entirety of agens employees, their skills and knowledge. Moreover, the partner database has information about the employee's resources and about their availability. Although it is open to everybody for reading, only the management and the specialist solicitors are entitled to add or modify information in it. In other words, any modification in an employee's knowledge profile or availability overview needs to be accredited and carried out by the management. A resource application gives additional information about the employees' involvement into current projects. It reveals to what percentage a particular employee is engaged in which project and what the employee's overall capacity utilization is.

Discussion

The case study at agens has revealed that a participative and evolutionary design process is feasible for all sizes of organizations, but that a large-scale use of an enterprise model in order to prepare for a computer-based integration with WfM or office systems requires a minimum size. agens has defined an organization-wide rule for setting up and structuring the organization in terms of fractals and specialist solicitors. Modification to the "structure" may be performed at every organizational level and by every partner. At agens each partner offers the own skills on an internal job market so that the best fitting employees are assigned to each project.

Several GroupOrga characteristics can already be found in the organizational design process at agens. The organizational design occurs as a participative process since all partners can be involved in it, but it is not a significantly distributed procedure. Due to its small size, agens maintains the aforementioned organizational, and quality handbooks in a centralized architecture. While the organizational database describe the possible entities of agens' structures, such as fractals, managers and specialist solicitors, its content does not yet describe the existing structure based on a comprehensive enterprise model. Hence, an integration between WfM and office systems with the organizational model was not realized. This may also be the case, since agens' size does not necessitate a full-featured workflow model and office management. The integration of computer-based tools in the design process took part as far as the groupware applications are concerned, but no graphical tools or other design aids were used.

A project in cooperation with the agens Consulting GmbH has measured the degree to which the GroupOrga characteristics were realized at agens and the other integrated service organizations. From it, suggestions were made about how to improve the organizational design processes at agens.

6.3.3 Field Study 3: Babcock Dienstleistungs-GmbH

The case: A mid-size building administration and IT services company

Babcock Dienstleistungs-GmbH (BDL) is a facility management and IT services company headquartered in Oberhausen, Germany. Its service program covers the engineering and the business part of administrating and running buildings, as well as the surface administration and complementary services. In addition, BDL has competence in network management and in the development of Lotus Notes-based applications for process management. As a direct subsidiary of the HKT Objektmanagement GmbH, the BDL belongs to the Deutsche Babcock Group, which in turn has a share in the H. Krantz TKT GmbH (see Figure 6-9). BDL has been founded in 1991 with currently 165 million DM in yearly turnover and about 200 employees—120 employees are engaged with facility management tasks, while the other 80 employees work in the IT services and Lotus Notes-oriented branch ([BDL 1998]).

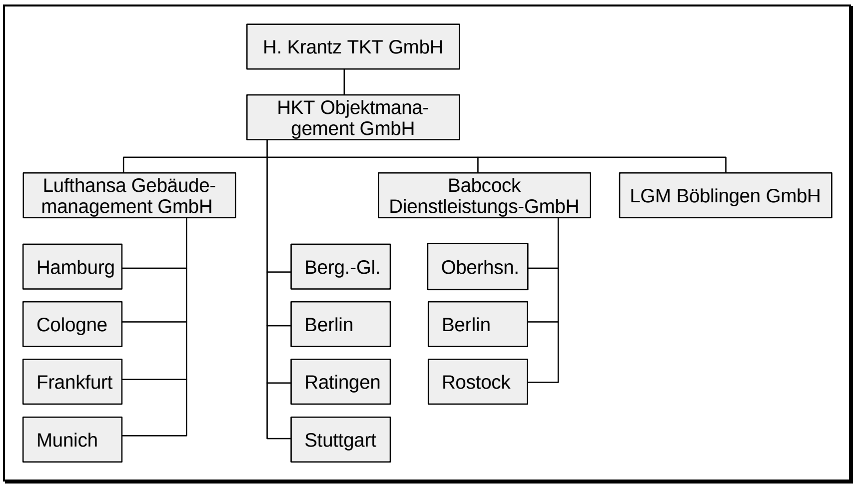


Figure 6-9: The facility management cooperation

The BDL runs all 17 buildings—offices and administration buildings, cafeterias, fleet and car pools, foundries, etc.—at the Deutsche Babcock Group's headquarters in Oberhausen, as well as the Babcock industrial area in Friedrichsfeld, Germany. Two other BDL sites are in Berlin and Rostock, Germany. The worldwide Babcock network, which connects about 120 corporations on over 100 locations with some 5.000 Lotus Notes users, is under BDL's full responsibility and management. The know-how gained in this sector is used for the process-oriented control of engineering and business task in facility management sector.

Case study results

During the realization of the case study in autumn 1998, the BDL was undergoing a major reconfiguration which practically abolished all forms of traditional hierarchical subordination. The final goal was the creation of four major product areas, namely *Intranet*, *E-Commerce*, *Facility Management*, and *Process Management* (Figure 6-10, [BDL 1998], p. 1). Although these four product areas could be understood as the traditional scope of four organizational units, at BDL they are intentionally not referred to as such. The four product areas do not show any form of super- or subordination, but are independent and self-responsible groupings in their own right. Each such group lacks of traditional positions, but has *Key Manager*, *Sales Manager*, *Product Manager*, and *Project Manager*. Key Managers establish the first contact with potential customers and present and explain BDL's core competencies to them. The task of a Sales Manager is, in cooperation with Key Manager and Product Manager, to offer concrete services to the customer. A Product Manager turns the offer into a concrete customer order. At last the Project Manager, as an *entrepreneur over time* ([BDL 1998], p. 2), is responsible for the successful completion of the project with the customer. Two internal functions are carried out by the *Personnel Manager* and the *Budget Manager*, who have the task to manage the project's human and material resources. Currently, the BDL has four Key Managers for acquisition and presentation tasks, three Sales Managers, and—at the final stage of the restructuring—about 15 to 20 Product Managers. At the moment of the investigation, 26 employees were assigned Project Managers, four employees were assigned Personnel

Manager and two Budget Managers existed. BDL's CEO is considered to be the Chief Key Manager.

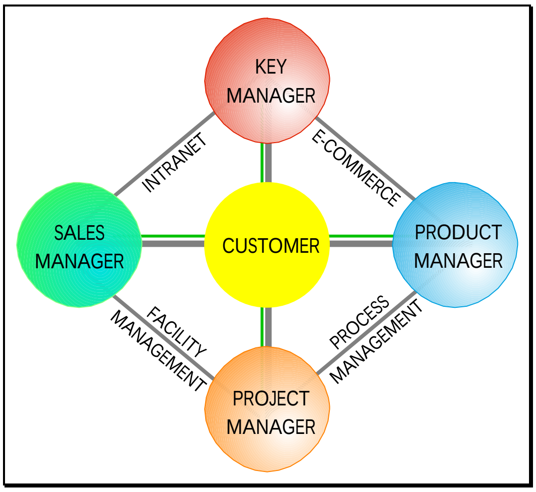


Figure 6-10: Organizational structure of Babcock Dienstleistungs-GmbH

Despite these varied functions, the large part of the 120 commercial employees fall into a category of project members. The former master craftsmen at BDL could become Project Managers at most, but usually they work as project members in the new structure. Some of BDL's projects, especially in the IT services branch, may last only over a certain period of time. Accordingly, the project members and managers in these groups are assembled only over the same time span, based on their respective knowledge and skills. Moreover, especially in IT projects, in one project an employee may be a Project Manager, while in another project the employee is only a project member. Besides these transient groups, other groups may last longer. A maintenance group, for example, might exist over years, although the tasks (projects) change constantly. The BDL organizational structure resembles a network, rather than a traditional organizational chart.

All information about this innovative organizational structure, its members, responsibilities, and functions is contained in a Lotus Notes groupware application. This application serves as a repository and at the same time as a discussion base for the managers. All managers are invited to participate in the (re-)structuring process by proposing and discussing new organizational forms in this electronic forum. The proposals are compared and discussed in order to present the basis for a temporal organizational structure. Moreover, this groupware application contains information about the expected goals to be reached by the various managers and their overall task profile. The application's entry point, a graphical navigator, is very similar to the organization's structure chart in order to visualize the resemblance between the goals and the real circumstances.

This organization database is open to every employee as a source for getting informed about the organization's vision, structure, strategy, and the four major product areas. In addition to it, a knowledge application and a human resource application are currently under development, however, no prototypes exist yet. The human resource application is projected to hold information about time spent in various projects, about holidays, and about personal interests. In a final stage this database is meant to be integrated with the knowledge base to provide information about skills and availability simultaneously (Figure 6-11).

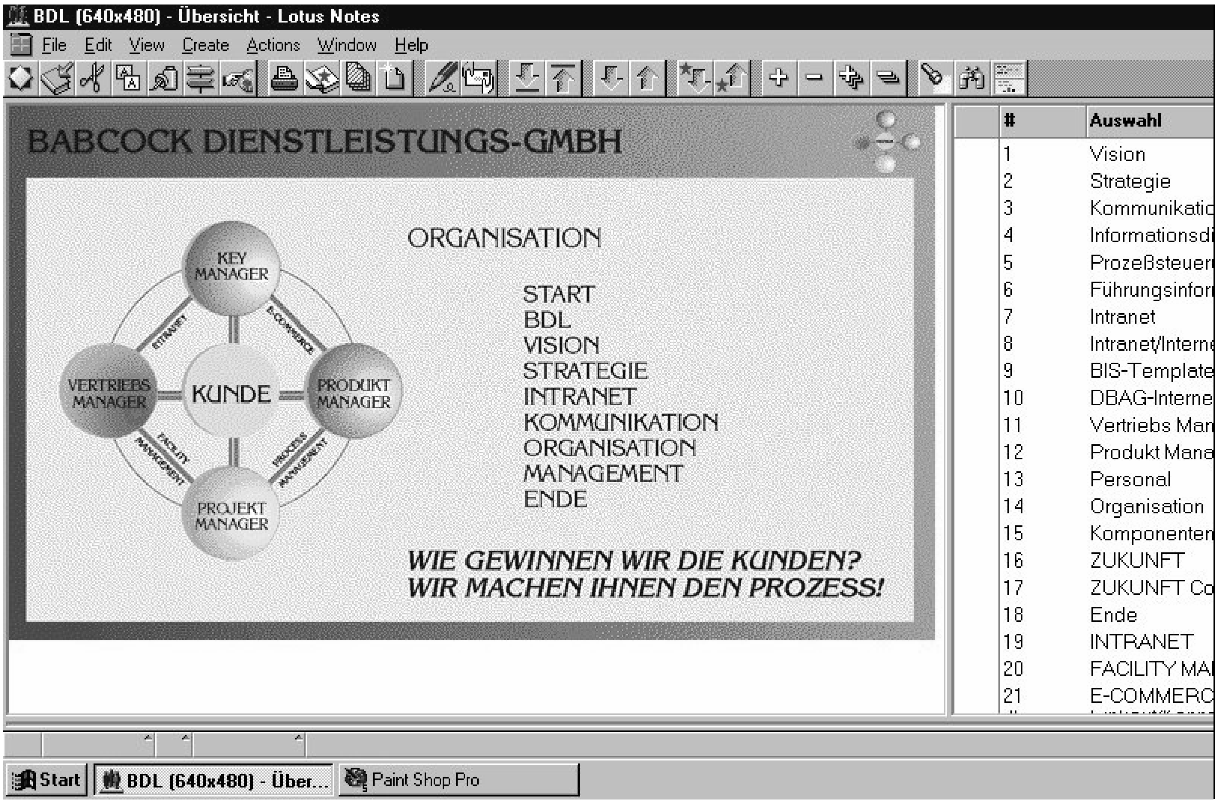


Figure 6-11: Entry point of the BDL organization database

In the process of organizational structuring, the key managers do not necessarily have a leading position. Although, they do not have a superior position in the sense of a hierarchy and they do not overrule other leading employees in a structuring process, generally the broad directions for organizational structuring are given by the higher management (the CEO). While at BDL all *managers* are entitled to participate in the discussion about organizational structures (in other words, major product areas and *manager* positions), not all project members are yet allowed to it. Participation in the organizational structuring and discussion process is free and nobody is forced to take part. Experience at BDL shows that not yet all employees feel equally responsible for the structuring of their own organization.

Discussion

The BDL completely concentrates on the groupware platform to document their structures, assets, and knowledge. While being in the change process at the time of the investigation, in the long run no paper- or file-based documentation was projected to exist. Because of this ongoing change process a unified regulation for organizational modeling and documentation did not yet exist at BDL. Some aspects of it were already carried out in innovative, computer-based applications while other aspects were still performed the traditional way. Although this was the case, the change process at BDL was strongly under way and the final result was already projected. As a result of it, all product areas will have a homogeneous concept for the documentation and modification of their organizational structures in groupware-based applications. A promising result of this change process is the BDL vision that Project

Managers and project members may belong to a specific project only temporarily and based on their knowledge/skill attributes.

The examination of the GroupOrga characteristics shows that a large portion of GroupOrga's vision is realized at BDL already. The organizational design process at BDL is certainly an evolutionary process, which is also participative. The new organizational structure was not imposed at once, but the re-organization is still going on with the participation of many organizational members. Although GroupOrga envisions that *all* employees take part in the design, at BDL a considerable step has been taken in this regard. On the other hand, BDL's organizational modeling does hardly rely on an enterprise model—be it traditional or innovative. Currently, no computable description exists about what entities may be designed and how they interact. The description of the four "Manager"-types in the organizational database helps to understand the person's profile, but is not operational for computer systems. Hence, an integration of WfM or office systems with the organizational database is not realized nor planned.

Great attention was paid to the attribute of *distribution* at BDL. All repositories, knowledge bases, or organizational applications are set up as groupware applications and are thus inherently distributed. In the case of BDL, distribution is not restricted to personal computers at fixed location, but it also includes the mobile user who can take part in the design and decision process. The design process is computer-supported since all design activities take part in the Lotus Notes groupware databases. Although, no dedicated organization design application is used, the database applications provide many features to describe the current and propose a future organization design.

The use of GroupOrga tools for defining the project groups, describing the product areas, and assigning the roles of Sales Managers or Key Managers might be advantageous in this case. Moreover, the GroupOrga EKB would be a valuable design template for the envisaged integration of knowledge/skill database, personnel database, and organization database. Such thoughts were put into more concrete forms during a cooperation project with the Babcock Dienstleistungs-GmbH.

6.3.4 Field Study 4: Siemens Nixdorf Informationssysteme AG

The Case: A European supplier of information technology

Siemens Nixdorf Informationssysteme AG (SNI) is the largest European supplier of information technology, with offices on every continent in a total of 58 countries. Siemens Nixdorf was created in 1990 by the merger between the Data and Information Technology Group of Siemens AG in Munich and Nixdorf Computer AG in Paderborn. SNI has a decentralized structure of 250 independent units throughout the world which are grouped into 12 autonomous lines of business. It follows a clear differentiation between the IT-driven

Products and Technology Services on the one hand and the *Solutions and Business Services* on the other, which are in turn based on the business processes of the customers.

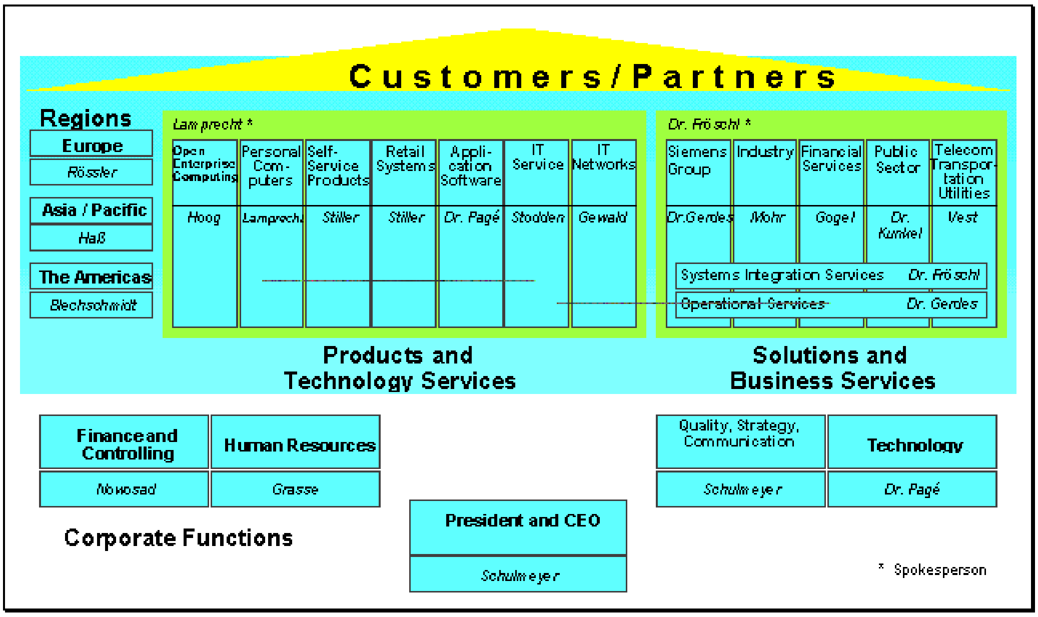


Figure 6-12: SNI corporate management responsibilities

As Figure 6-12 (see [Siemens Nixdorf 1997a], p. 26) shows, the corporate structure brings every organizational unit under one roof. The mentioned two divisions include the twelve lines of business. In addition there are regions that focus on the international market and the corporate functions that are not directly included in the hierarchical structure. *Solutions and Business Services* is structured in a matrix form, with two cross-functional *services*.

The area of the GroupOrga field study within SNI is originally located in *Application Software*. Although its investigation spanned the complete structure, the concrete structuring process itself was investigated in *Application Software*.

Case study results

The SNI intranet is the central and innovative information source for organizational information within SNI. Various pages and sites with organizational information (employees, structures, responsibilities, etc.) can be found, however, a complete organizational database does not exist. Before the intranet came into existence, printed process handbooks were available which used to be very detailed. Today this printed documentation is reduced to a necessary minimum. Organizational handbooks were not widely used and did thus basically not exist. In early 1998 the intranet operated with full functionality for about one and a half years. It is accessible from every employee's workplace within the enterprise. Its focus is the dissemination of internal and externally collected information which is considered important for the operative work. Diverse information can be found by means of navigation and with the assistance of search engines.

Amongst others, information about processes and structures can be found, working schedules are available, organizational charts can be displayed, etc. However, the information available is not yet presented in a coordinated and navigable form, but it appears to be rather unstructured. Search engines such as the SNI product *Consult Info* are still the most useful navigation tools. Every division and organizational unit is self-responsible for what and how to present information in the intranet which results in various forms of layout, linking and navigation. Organizational information is depicted differently by every unit and sometimes the employee who is responsible for the information contained in a particular page cannot be found out. The study revealed that SNI's intranet is heavily accessed, however structural information is found rather by chance.

Besides these mere presentational HTML-pages, the SNI intranet comprises a number of innovative tool-environments for organizational purposes.

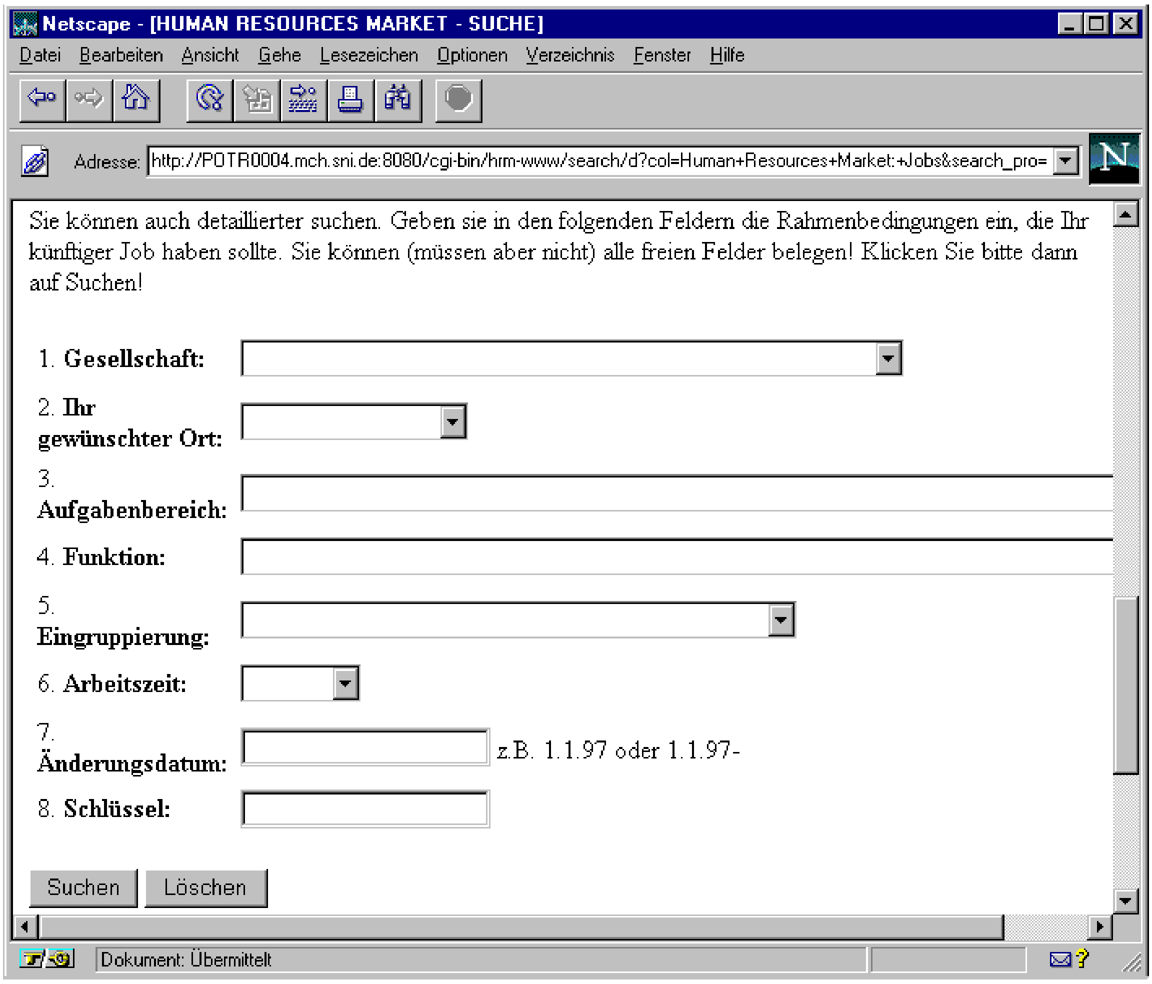


Figure 6-13: Human Resource Market (HRM) in the intranet of SNI

For instance, the intranet-based SNI job market application *Human Resources Market* (HRM) (see Figure 6-13) relies on SNI's principle that each employee is self-responsible for the own career within the enterprise (see [Kürn 1997]). It provides the possibility to be informed about job vacancies and to directly apply for a particular position or for temporary vacancies in projects. Moreover, each employee can use this application to generally apply for another job and to internally offer the own qualifications without addressing a specific vacancy.

Furthermore, a *skill-database* exists into which the capacity, the abilities and talents and the know-how of employees can be documented. This skill-database had about 2500 entries at the time of this investigation and it was constantly growing and heavily accessed. Another application, the *Who-is-Who?* allows to find the correct employee in the enterprise to turn to in case a customer needs a special consultation or information. Yet another application is an internal directory with several search options, as well as an externally accessible X.500 directory with employee names, addresses and locations. A stand-alone *Address-Tool* can be downloaded and locally installed which offers a user friendly interface for the retrieval of address information from enterprise wide employee databases.

Discussion

While any organizational restructuring within SNI has to be centrally approved, its documentation and publication is completely left to the various divisions and departments. SNI's organizational modeling takes place as a bottom-up and top-down approach at once. Modifications are requested in bottom-level departments and they have to be approved by higher management. However, the way in which such modifications are documented, how often this documentation is made topical and what form of presentation (layout, color, etc.) is chosen in the intranet remains open. In terms of the GroupOrga approach this procedure is somewhat in the reverse. With the GroupOrga toolset the documentation and actualization of structural information would be quite simple and in a uniform manner. With it, the navigation in the intranet could be made more reliable and distinct.

The presented tools are by far not all applications which can be used in the SNI intranet, however this selection already shows how comprehensive the information provided by it could be. However, since it is rather unstructured, much personal initiative is required to get the necessary information and to use it accordingly. HRM is an interesting application towards new and leaner organizational structures in which the employees tend to shift jobs and projects more often and in which hierarchies and positions are less important. An integration of the various applications for address searches, skill management, human resources, organizational charts, into a framework similar to the GroupOrga framework would be quite beneficial for SNI. With all the applications in existence, such integrated environment is not a big step away.

At last, the GroupOrga characteristics will be discussed in the light of this field study, as well. Although SNI has a comprehensive and powerful organizational modeling application with its product OIS V1.0 (see section 5.5.5), interestingly the company is not extensively using it throughout the enterprise. Hence, hardly any integration between what is modeled as organizational structures and SNI's own workflows and its office system has taken place. The workflow product WorkParty may be used internally in some departments but it is not connected with an enterprise-wide OIS organizational database. This also implies that the organization design and its documentation is not based on any kind of enterprise model such as the organization's own ORM. The documentation process is computer-supported, however the actual design process is not, since it still relies on the traditional, hierarchical approval paths. SNI design processes can partly be described as a distributed and participative process since the various organizational units and divisions take part in the structuring to a certain extend—the degree of participation that is suggested in the GroupOrga vision is not yet realized.

6.3.5 Lessons Learned

Sections 6.3.1 through 6.3.4 presented the characteristics of GroupOrga, how they were realized, and how they functioned in the cases studied. While more case studies were conducted in GroupOrga, these four cases were chosen for presentation that showed contrasting forms and realizations of the GroupOrga characteristics. Instead of summarizing the cases, this concluding section emphasizes which case showed which characteristic in the strongest form, and highlights the peculiarities of each organization.

The Deutsche Bank AG case showed a very well-defined *distributed procedure for organizational design*. There were many specifications and rules that explained how to perform the organizational design process, which tools to use, and whom to integrate. The case revealed that a large organization must have well-defined procedures and that organizational design can only work well if many organizational *experts* are integrated into the process. Participation in this case meant that lower-level organizational units (or their managers) do participate, but only a few.

agens Consulting GmbH is a contradictory example. Its small size allows many (if not all) employees to partake in the design process, which, however, is not strongly predefined by management. Thus, the organizational design is a much more *evolutionary design process* than in the Deutsche Bank AG, since the organizational members modify and design the structure whenever necessary. The selection of employees for projects occurs in accordance to their skills and only for a short period of time. This rarely happens in large, hierarchical organizations such as the Deutsche Bank.

While the Deutsche Bank strongly relies on hierarchical elements, agens chiefly depends on non-hierarchical entities, such as project groups and teams. Babcock Dienstleistungs-GmbH defines its structure similarly by means of workgroups set up and roles played. Its design is also a *highly participative process*, but less evolutionary than with agens. The BDL case shows that in the long run, entities other than leader, organizational unit, or position will gain importance. Since the BDL was in the midst of a re-organization, a parallel structure (and terminology) of traditional characteristics and innovative forms was found. In other words, although the entities *Key Manager*, *Project Manager* and *product areas* denote a new era in organizational structuring for the BDL, its purpose still resembles that of a traditional functional and layered design. This observation also applies to agens.

The case of Siemens Nixdorf Informationssysteme AG stands out because of its strong *computer support in organization design*. Being both a supplier of information technology and a software developer, SNI uses information technology to support its design process much more than the three other companies. Many applications are used within SNI to describe the structure of the organization, to list knowledge and skills, to post vacancies, and to distribute organizational changes. But while the computer-support is exemplary, the process support for modeling organizational structures is weak. Many distributed units and workgroups may describe their organizational structure on the intranet, but little is done to integrate this knowledge into an organization-wide, distributed repository.

All four cases lacked two major characteristics of the GroupOrga vision: the integration of an organization application with WfM or office systems, and the extensive use of a comprehensive enterprise model.

None of the four cases presented here used a workflow management system for all organizational processes. When WfM was implemented, the organizational information was hardcoded and did not rely on an organizational database. Similarly, in the four cases, any form of documentation of organizational circumstances was based on free-text descriptions or unspecified organizational models. Predefined organizational meta-models could not be found, which is the main reason for them not having integrated the process structures with the organizational structures by means of IT-support.

Chapter 7
Summary, Evaluation, and New Directions

Research on the concepts of distributed and participative design of organizational structures is the focal field of the GroupOrga project described here. Accomplishing the objectives of the introductory chapters, the project examined the traditional forms of organization design before the requirements of innovative workflow management and office systems. It presented an enterprise model for the representation of organizational infrastructure and a prototype framework for the realization of the distributed and participative design of such a model. Concepts for the distributed management and use of groupware applications and knowledge bases within the framework were also presented. The applicability of the proposed approach and prototype is examined by the implementation of GroupOrga in case studies that realized and adopted parts of the research presented here.

In this final chapter, the novel contributions of this work to research on organization design in the field of WfM and office systems are summarized and reviewed with the requirements that were presented. Section 7.1 gives a quick overview of the contents of the research. Section 7.2 presents the results of this research, reviews some of their implications, and discusses the transfer of the proposed concepts into a groupware product. Section 7.3 offers an outlook to future research and development.

7.1 Summary of Contributions

The findings of this research cover four areas:

- The introduction of the basics of organization design terminology, as well as the analysis and examination of traditional theories of and technological approaches to organization design

- The development of a meta-model for developing dynamically adaptable enterprise models
- The development of an architecture to support the modeling process and the use of groupware-based technology to implement prototype applications
- A presentation of a design methodology on how to use these applications and tools in a participative process

The research presented here started with the foundations for GroupOrga and with a definition of terminology in chapter 2. The team was positioned as a core element of innovative working environments, and different forms of computer support for teamwork were identified and evaluated. Groupware was shown to be more than simply a new software package—it is a supportive application environment for comprehensive teamwork tasks. Office management systems were presented, and if the distributed team is to be supported, putting existing office automation packages onto networks is not enough. Workflow management systems are another key technology in the GroupOrga research. An insight into the understanding of WfM in the GroupOrga research was also given. The less technical field of organization design was defined and literature was reviewed. Chapter 2 concluded with a presentation of the practical foundations of the GroupOrga project, such as Lotus Notes, Pavone GroupOffice, GroupFlow, and ESPRESSO.

Chapter 3 gives a problem definition and comments on the organizational situation taken as the basis for this project. The research is loosely confined to the parts of the organization that deal with information processes of average complexity, such as marketing, sales, or controlling. Since these process are of average size, coordination of the employees is necessary. The sample organization is also of average to larger size, that is, the concept is not aimed at smaller and very small organizations. Chapter 3 continued with the identification of requirements for an innovative organization design approach. This list covered a wide range of goals:

- Integration of workflow IT and organization design
- Focus on flexible organizational subsystems
- Creation of a data model for WfM and office systems
- Distribution in the modeling
- Tool support to process oriented organization design

These requirements were then used to evaluate traditional theories and related systems and concepts for BPR in this field. The review showed that hardly any of the examined approaches and systems fulfilled a larger set of requirements and that some of the requested concepts were completely missing.

The conceptual foundation for the GroupOrga framework was laid in chapter 4 through the presentation of a meta-model for organization design. Chapter 4 introduced the meta-model GEIMM, which defines the construction elements and rules for the creation of a specific enterprise model that represent a particular organization. GEIMM was refined into a process model, an information model, and an infrastructure model. A distinction between organizational entities and their relations (according to the EER approach) allow an expressive modeling of real world organizations of various types. Such constructs provide for both a modeling of hierarchical organizational structures and the modeling of more dynamic network oriented relationships. Therefore, the concepts of the GEIMM satisfy the requirement for the representation and provision of organizational structures by entities and relationships, as well as for the provision of organizational services. Chapter 4 also presented a list of references that showed preliminary projects and efforts which inspired the development of GEIMM. It concluded with a brief introduction into an organizational database, which was the storage for a concrete enterprise model based on GEIMM.

Chapter 5 considered architectural issues and the services and tools of the GroupOrga prototype system. There were two main parts in this chapter:

The first part introduced the GroupOrga vision: a thinkable form of distributed and participative organization design. First it demonstrated the basic GroupOrga concepts using the requirements in chapter 3. It named the most important characteristics and explained their possible advantages. An integration of workflow IT and organization design, the idea of evolutionary organizational subsystems and the realization of a participative, learning organization were discussed. Distribution in organization design was positioned as a main concept. It was supported in the sense of spatial distribution as well as administration distribution. In other words, the concept promoted an architecture where spatially distributed designers are responsible for different information types. This was achieved through the use of client-server architectures and administration responsibilities. A comprehensive top-down modeling approach, which combines a top-level structuring and a bottom-level design at once, was presented. Major concerns of the approach, platform independence and tool support for a variety of different user types, were tackled in this chapter. The enterprise knowledge base and its characteristics was another concern here. It incorporated mechanisms for the provision of awareness about organizational changes, which achieved two primary goals. First, information about structural modifications was automatically distributed in the organizational context of the modified entity. Second, the intensity dependent diffusion of messages tied in neatly with the introduced organizational model of the chapter 4. Since in some approaches the X.500 directory appeared to be the primary candidate for the provision of an infrastructure model, chapter 5 investigated the applicability of the X.500 standard. The X.500 analysis showed that the directory fulfills several (but not all) requirements of GroupOrga. This was mainly due to the conception of the X.500 directory as a communication directory, which reduced its usability in frameworks of distributed organization design. Solutions for an integration of X.500 with GroupOrga were discussed.

The second part of chapter 5 described the implementations of GroupOrga applications in their current prototype stage. It outlined the layered architecture, as well as selected components of the system. Several implementation aspects were considered in detail:

- Realization of the EKB
- Implementation of the graphical organization modeler
- Management of the distributed repository structure
- Browsing functionality
- External connectivity
- Thoughts on graphical analysis

Several window-based tools for the modeling, browsing and evaluation of infrastructure models were presented here. The GroupOrga OrganizationModeler, for instance, allowed the creation and modification of infrastructure entities, relationships and attributes under consideration of the access rights specified in the distribution model. The two adjacent areas of process modeling and information modeling were also examined for completeness.

In chapter 6, an enactable organization design methodology was defined. It specified how an organization designs, or redesigns itself to adapt to changing internal and external pressures by using the GroupOrga framework and toolset. This design process defined a new approach to organization design, where the group design process is the change mechanism. In life-cycles of infrastructure models, analysis, design and development were traditionally treated as clearly distinct phases. However, the GroupOrga framework integrated these phases seamlessly across an enterprise, and thus represented a significant advancement over existing methodologies and tools. As an evolutionary approach, GroupOrga used multiple perspectives, and is based on informal organizational roles and structures. Thus, the process defined "organization design by the people". It is an *in vivo* process, that is, everyone in the organization participates and contributes in a significant way to the resulting organizational structure. Case studies conducted during the project form the conclusion of chapter 6. Their main aim was to identify and discuss critical components of the vision and framework. Moreover, the testing of the GroupOrga tools and database applications was another goal of the case studies and lessons learned from the cases are given.

7.2 The GroupOrga Approach in Retrospect

Research has shown that the *traditional* approaches to organization design have been disappointing. Among the problems cited were:

- Failure to take into account the members of an enterprise
- Failure to always observe the current organizational situation
- Centralized design process that was time consuming, which thus produced an outdated result
- Output that was not very useful in today's workflow and office environments

The overall conclusion was that there must be a fundamental rethinking of organization design methodologies that focus on an integration with modern IT in business. However, before an organizational methodology may be designed, a conceptual foundation has to be built first. This foundation is the GroupOrga approach. Its ingredients are derived from organizational theory and from IT. This section discusses some of the results and suggests further research topics.

"Effectiveness is driven by the relationship among components (congruence) rather than by the inherent characteristics of individual components; thus, there are very few universally good approaches to organizational architecture. Different ways of organizing will be more or less effective for different contexts, for different technologies, and for different people." ([Nadler/Gerstein/Shaw 1992]). This finding also holds true for the approach developed and implemented in this project. It is not a cure-all technology for organizational modeling, but organizational and technological restrictions that were outlined throughout this research do apply.

A specific objective of the GroupOrga research and implementation is to address the incorrectness and unavailability of information on organizational structure. Use of GroupOrga components such as the EKB and the browsing facilities changes this, and people are now able to manage resources more effectively, and perform online performance monitoring. Two outcomes may result from this: fear of electronic surveillance and competition among the participators of the design process.

Various case studies and references revealed that with such approaches, the data was much more accurate than before ([Orlikowski 1995], pp. 18f.). An important element in this increased accuracy was the constantly updated modeling where employees dynamically monitored and corrected errors or areas of ambiguity. Meyersiek [1995] stresses that monitoring is part of an employee's job and that it is something that could be positive when used to reflect well on the organization's overall situation. Meyersiek adds that monitoring can increase productivity and that currently existing regulations in organizations which prevent such technologies have to be rethought. Österle, Saxer and Hüttenhain [1994] summarize their article on *organizational monitoring* by explaining how such approaches can help depict the real organizational structure in contrast to mechanically installed structures. While Orlikowski presents results from US businesses, Österle, Saxer and Hüttenhain focus on the stricter European market. Two investigations conducted during the GroupOrga project also revealed a positive management opinion towards these concepts in Germany (see [Walsch 1997], [Nolte 1998]).

Inevitably, the more technologically mediated the work, and the more valuable and effective the mediation, the more dependent the design work and the employees become on the technology. Such dependency may become apparent in two forms: a physical dependence on the availability of the GroupOrga tools, and a psychological dependence contained within the technology. The former is manageable with various backup systems and security hardware. The latter is more problematic because it is a state of mind. It is especially problematic for employees who have never known to work without groupware technology, and for them, work may not be conceivable any other way. Providing training that specifically offers alternative models for working with and without the organization design technology might prevent this form of dependence.

7.2.1 Limitations of the Theoretical Research Approach

The pragmatic goal of an investigation, such as the one presented here, is not necessarily committed to an empirical research conception; however, such a connection is often considered useful. Instead of relying on empirical tests and analyses, this project referred to published academic findings. Such a secondary analysis has the drawback that the way in which the results were found is not always clear. Moreover, the results cannot be veryfied anymore.

This criticism is partly true for the approach presented here. On the other hand, both organization design theory and IT have basics that are widely accepted: situational or contingency theory approach, and groupware and WfM technology. The fact that GroupOrga is based on these preliminaries, positions it as an extention of the accepted foundations. Getting to these novel thoughts from the given frameworks is based on reflections of plausibility and partly on empirical findings. Thus, making the connection between the two fields more transparent and suggesting recommendations for practical scenarios was a primary concern.

Hence, the concept presented here does not mainly depend on its theoretical cornerstone, but rather on its practical effectiveness. The practical relevance of a comprehensive organizational model rates higher than an empirical description of the current detailed problems in the field. Considering today's gap between the theoretical description and the practical implementation of IT, this procedure seems justifiable. Nevertheless, it should be stressed that the rational organization has its limits, which cannot be expressed by rules and formulas. IT can be implemented meaningfully, only if topics which can be formalized to a certain degree are considered. In the field of organization design, IT is only an instrument, but one which can improve the conditions if it is available to many people via groupware technology.

Whereas the academic audience will benefit from the advancement in the theory of organization design in this study, practicing managers will benefit from gaining an understanding of new organizational forms and corresponding management practices that could become a trend in the future.

7.2.2 Distributed Participation vs. Acceptance and Resistance

In GroupOrga, anyone in the enterprise can be a member of the organization design process. However, logical questions to ask are:

- Why would everyone in the organization be part of the process?
- What about the existing work overload?
- What is the incentive for a worker to participate in the process?

In the light of these questions many researchers point out that employees are underutilized and underchallenged, which leads to dissatisfaction and poor performance. The participative approach has shown to be the more effective and popular approach in today's business. So far, this section and sections in chapter 3 have focused on the personal benefits to employees. There is also an important organizational benefit: The design process is highly multi-disciplinary because the modeling can be bound to anyone from top to bottom.

Markus and Collony [1990] argue that possible failure of a participative approach may also be due to *interdependence* in the recompense derived from the use of concepts such as GroupOrga. Interdenpendance is when the benefits and advantages of cooperative applications to one user are contingent on the behavior of other users. On the one hand, interdependence can lead to the failure of such application frameworks. On the other hand, interdependence is intimately linked with the benefits derived from cooperative concepts as opposed to those derived from single-user, central concepts. But in contrast to experiments conducted by Markus and Collony, in GroupOrga, users participate in the modeling in order to improve their direct working context (their workgroup, their organizational subunit, or their knowledge/skill entry). For them, a better and more accurate modeling of the *whole* model is of minor benefit. Therefore, few would rely on the modeling (and participation) of others since this would not significantly improve their own situation. A recommendation beyond that given is that further basic social research will provide long-run solutions to the problems arising from interdependence in cooperative application frameworks.

Another important question in using technology to support participative design is how to provide incentives that encourage people to share (structural) information. Malone and Rockart [1993] and Orlikowski and Hofman [1996] consider this aspect. At a basic level, the employees in an organization should be allowed to bill the hours they spend learning the organization design system—an observation which the case study in the Deutsche Bank AG has revealed. A more subtle problem is that employees may be rewarded for being *experts* on something, that is, for knowing things others do not. While this topic appears to be very problematic in fields were accumulated knowledge is required to be shared, it is of minor consequence for structural knowledge—organizational structures are *public* knowledge in an EKB. Moreover, by making the possession of expert knowledge public, it becomes even more valuable since this information may now be available to a wider group.

In addition to improving and better supporting the design of collaborative systems, it is also important to consider how they are going to be integrated into organizations. This process of introducing a collaborative system can be complex, with numerous obstacles and resistance in its path. However, as far as this aspect is concerned, GroupOrga, like many other CSCW environments, is a collaborative environment. Various studies have analyzed barriers that ensue when a multi-user system is implemented. Hence, no additional thought is given to this respect, but it is referred to cited literature ([Rogers 1994], [Grudin 1994], [Bowers 1994], and others).

7.2.3 Experiences with GroupOrga

For evaluating GroupOrga, two factors were considered: its modeling capabilities, and the appropriateness of the tools and concepts in cooperative environments such as workflow and office systems.

During and after its development, tools from the GroupOrga framework were tested in various organizational structures through a cooperation with NotesWare Ltd. and Pavone Informationssysteme GmbH. Organizational structures of customers and partners were designed to represent a large set of employees, workgroups, organizational units and roles. In addition, from various demonstrations, segments of other (partly imaginary) organizations were constructed. In all cases, the GroupOrga approach of providing a toolset was proven successful. In almost all cases, GroupOrga could represent different infrastructures and their entities and relations. Such test phases resulted in a number of minor modifications in technical and organizational details (apart from identifiing and eliminating software bugs).

At a stage where the infrastructure is modeled by the GroupOrga tools, it is interesting to evaluate the applicability and merits of the distributed EKB in a cooperative environment. For the test implementations mentioned above, this usefulness has already been proven, since the GroupOrga EKB is fully integrated and operable with cooperative environments for workflow, office and project management. However, in these cases, the EKB was used only in a restricted form of distribution. For the next step, a comprehensive evaluation should be based on a fully distributed organization repository which is available to people working in different organizational domains, and its administration should be integrated into daily administrative procedures.

Although the empirical results of a widespread evaluation of GroupOrga would have paid for itself, this was out of the scope of the project. Testing in this direction, using partners in

Germany, the UK, and the USA, is planned for the future. To the general public, GroupOrga was presented at large computer exhibitions at Lotusphere '97, Lotusphere '98 and CeBIT '98. During the project lifetime, it was demonstrated in various frameworks of groupware applications in combination with the mentioned commercially available products. To the academic community, it was demonstrated at conferences of the Gesellschaft für Informatik e.V., at BIS '97, BIS '98, WI '97, HICSS '98, and WET ICE '98.

Future development aspects are application oriented and related to the technical concepts presented here. In section 7.3, a concluding outlook to proposed research opportunities is given.

7.3 Proposed Research Opportunities and Further Questions

While the conceptual GroupOrga approach is quite advanced, the implementation of the prototype is currently only realizing an mid-sized environment for experimenting with the concepts of participative and distributed organizational modeling. In order to realize the full power of the approach, various things remain to be done. Most notably the representational issues of the GEIMM in the graphical tools, the entity and relation types need to be extended. Additionally, the existing analysis functionality has to be further diversified and simulation should be taken into account.

Future research could also investigate the usability of the system in a larger outer organizational context to determine the needs when it comes to linking partners across an organization's borders. Concerning the connection of organizations to each other, there are several research opportunities in the field of *virtual organizations*. While this aspect may not yet be directed to implementation aspects, but rather to basic concepts and suggestions, a further specification of how to integrate the WWW into the GroupOrga system aims at a more technical level.

Another area of immediate research can be the discussion, modification, and possibly the enlargement of the GEIMM. It has to be examined if the proposed model is effective, or if additions and modifications have to be undergone, for instance in the course of the integration of an outer organizational context as suggested above. The use of the GroupOrga framework and system in connection with the topic of knowledge management should also be included. Future research should consider what the implications are, when knowledge management becomes a topic, and how existing human resources applications (like PeopleSoft software) can be integrated.

The following sections discuss further research topics in more detail.

7.3.1 Representational Issues in Graphical Tools

The GroupOrga OrganizationModeler and other modeling tools presented in this project visually display organizational information by a graphical user interface. These tools have been implemented to show the practicability of such modeling and to test its acceptance and usefulness.

Future research on organizational modeling tools should concentrate on excellent user interfaces, interaction modes, and the usability of such tools. For example, research could explore the application of virtual reality technologies. Rather than displaying the resources, the people, the organizational units, and the workgroups on a flat two-dimensional screen, one could imagine the visualization of an organizational infrastructure by navigating through a lifelike three-dimensional space. Such three-dimensional displays of the infrastructure can present more complex relationships in easily understandable ways. In addition, such displays would allow the user to connect various dimensions of the enterprise model, such as linking the workgroups with the organizational units. Also, a reorganization of the organizational information would be easier to maintain.

The ability to undo actions is a standard feature in most single-user interactive applications. However, for collaborative applications that allow several users to work simultaneously on a shared information storage, undo capabilities are difficult to imagine. This aspect becomes interesting in the GroupOrga framework. What if a user has modeled an organizational structure based on someone else's design and then overwrites the initial design? Is it sufficient to provide a global undo in the graphical tools, where the last change to the infrastructure made by the last user is undone, instead of allowing users to only undo their own modifications? Such research should consider the possibility of conflicts between different user's actions that may prevent a normal undo and it should propose a general framework for undoing actions in collaborative work.

7.3.2 Improved Analysis and Simulation of Organizational Structures

The concept of latent and observable organizational variables for organizational analysis was introduced earlier. Because organizational analysis was only touched upon in this project, it could present another interesting research opportunity for further developments as spin-offs of the GroupOrga project. [Lin 1994], [Krackhardt 1994] and [Unterstein 1994] provide valuable imput in this direction. Lin examines, compares, and evaluates various mathematical measures for organization design. These include structural indicators and process indicators. For GroupOrga, the structural indicators are more interesting, one of which is presented in [Krackhardt 1994] and in [Krackhardt/Stern 1988]. While the GroupOrga tools confine their analysis to the documentation and visualization of the analyzed infrastructure, Unterstein introduces how a profound database-supported organizational analysis can be performed. This valuable suggestion goes well with an examination of the EKB and its contents. While most researchers often develop new measures with little attention to previous measures, a more practical investigation should put the existing measures into practice and evaluate their usability.

The identification of new organizational forms has been identified as a goal of organization theory. Various forces (changes in the environment in which firms operate, for example) suggest a rapid alteration in the evolution of organizational structures. An important future question is how to search the space of possible organizational structures for possibly useful forms. With computer simulation, there is a choice next to a *wait-and-see* solution. Using computer simulations to search for possible organizational forms has several advantages. First, the same evaluation of an organizational structure can be performed repeatedly under the same conditions (see [Crowston 1994]), thus providing a high validity. Second, variations in organizational subforms explored in a computer simulation are not restricted by social factors or human influences. However, since computer models abstract from real organizations, the features simulated must be chosen carefully to ensure that conclusions drawn from the result can be applied generally.

The first issue of future research is what to simulate, that is, establishing the level of simulation of the organization. A simulation could represent entire organizational structures or organizational subunits. For GroupOrga, the most direct approach is to simulate the competition and cooperation between substructures in an organization to show how these interactions result in a particular overall infrastructure. However, if a simulation that models individual substructures is really workable and which substructures constitute a single organization is to be examined in subsequent research.

7.3.3 Enhanced Modeling of the Outer Organizational Context

Clark and Schiano [1996] acknowledge that the recent interest in interorganizational connectivity has been fueled by the opportunities brought about by a reduction in communication costs, particularly the linking of computers. However, they found that standard connectivity such as Electronic Data Interchange (EDI) is not sufficient to generate substantial savings and that organizational changes are also necessary. Newer approaches that focus on interorganizational connectivity, especially in the field of workflow management are: [Groiss/Eder 1997], [Amberg 1996], [Kamath et al.1997], [Kozlowski 1997], and [Adams/ Dworkin 1997]. A similar approach, which is groupware-based, was developed as a co-project to GroupOrga ([Hilpert/Riempp/Nastansky 1994], [Riempp/Nastansky 1996], and [Riempp/ Nastansky 1997]).

The research for connectivity in GroupOrga contains an examination of the X.500 and LDAP directory standards. It focuses on connectivity by means of the GroupOrga Connector and by means of database browsing and linking. Since the X.500 directory is not a suitable basis for GroupOrga concepts, it is also not a suitable basis for an interorganizational connectivity as understood in GroupOrga. Amberg, who also introduces the term of *actor* as a generic term for *machines*, *computers*, *persons*, *organizational units* and *roles*, notes that the organizational structure of the enterprise is an important fact, one that should not be part of a wide area workflow system. On the contrary, he believes that "this may hinder consistency and completeness of the workflows. So this kind of a distributed modeling—without any predefined regulations and structures concerning modeling domains—is advisable only in special cases." Amberg's (and other authors') suggestion, to integrate a comprehensive, distributed organization modeling environment such as GroupOrga with wide area workflow systems, appears to be more forward-looking than the development of an organization database as part of such a system. Of primary importance for future investigation is thus an integration of wide area workflow solutions with the GroupOrga approach (see Figure 5-45).

With the emergence of the WWW in early 1994, it must be considered as another platform to support the modeling of the outer organizational context of an enterprise. However, the first aim of the WWW is to publish hyper-linked, multi-media documents and make them available to the user. The aim of GroupOrga is the modeling of organizational structures. The emphasis in this distinction is on the comprehensive modeling aspect. Thus, instead of considering both systems as competitors, future research must regard them as application platforms that can be combined. Recommendations for extending the WWW part of GroupOrga for more collaborative work could include a richer model of access control, a better user interface, and better support for a wider range of user types. Therefore, future research could further refine the scale of user classes (see Table 5-1 and chapter D in the additional documentation). As a secondary step, the graphical modeling tools for the WWW would need to be enhanced.

7.3.4 Organizational Models and Tool Support for Virtual Organizations

Terms like *virtual organizations* or *network organizations* were used to describe how organizations are no longer entities where people work at the same location.

Based on the research on virtual organizations ([Sieber 1995]), further questions in a GroupOrga continuation could examine the notion of virtual organizations under the influence of traditional organization design aspects. Such research needs to explore the view that the classical design of organizations remains a fundamental management task for virtual organizations, as well. Can an evolutionary and distributed multiple level team-approach to design be proposed for the modeling of a virtual organization? Yes. Although *designing virtual* organizations appears to be a contradiction in terms, the indication of the bounds of action within the virtual infrastructure is a necessary focus of research in this area.

Such ongoing research might identify and define a set of additional technology-based organizational entities and relations. Some of these variables are similar to traditional design variables, while others are unique. [Ott/Nastansky 1997b] and [Ott/Huth 1998a] have taken the first steps in this direction. [Ott/Nastansky 1997b] agrees with Klein, who defines the design of an organizational architecture as an elementary management task: "Virtual, flexible organizations require a minimum of structure, too. Therefore basic organizational principles have to be determined and rights and responsibilities of organizational units and their agents have to be clarified" ([1994], p. 313). [Ott/Huth 1998a] focuses more on the technological requirements of a modeling of virtual organizations and introduces first aspects for further research in this respect.

7.3.5 Conceptual Questions and Modifications to the GEIMM

The GroupOrga project presents a concept for the interplay of an organizational modeling system with other workflow or office management environments. Although much effort has been put into this integration, an interesting further research aspect is the enhanced coupling of the system with workflow applications, and the role the GroupOrga system can play for the support and analysis of a business reengineering process. A first step into that direction is the connection of GroupOrga with GroupOffice, ESPRESSO and GroupProject, as well as with the OIS and BONAPART systems. Further research in this direction is required to identify general interfaces between the systems to generate an infrastructure library.

Another study could seek to extend the research on the effects of automating paper instruments by examining the area of organizational documentation. When paper instruments are automated, the question of equivalence between paper and computer forms occurs. At first, it might seem obvious that putting the documentation of organizational information on a computer display should result in the same responses as on paper. But the two formats can differ enough to cause significant variations in the responses. A study on format effects would be made even more interesting by the fact that the organizational modeling approach can now be performed by not only one person, but by everyone with a computer.

A further investigation could explore modifications (introducing, modifying, or deleting existing entity and relation types) to the enterprise model GEIMM. This aspect is intensified if the integration of an outer organizational context which may require new entities, such as an *external partner*, a *cooperation*, is considered. Further research is needed in this area, based on experiences with the GroupOrga system in combination with wide area workflow systems. Additional entities, such as a *strategy* entity, may also be introduced. In an overall enterprise model, strategy is considered a high level of abstraction. Therefore, interconnectivity between strategy and other organizational entities (especially the process model) is likely to include integration across several partial models.

From an empirical point of view, the success of the enterprise model can be measured in a later study by the extent to which the representation successfully models two or more enterprises.

The suggestions presented in section 7.3 are the basis for further research activities in this field. GroupOrga provided theoretically profound and practically tested data material. The respective modules and process steps are optimally adjusted to each other, and any detected errors and known problems have been determined and solved. Further efforts need to refine the framework and put it into action.

BIBLIOGRAPHY

[Ackermann 1994]
Ackermann, Mark S.: Answer Garden: A Tool of Growing Organizational Memory, Ph.D. Thesis, MIT Sloan School of Management, 1994.

[Adams/Dworkin 1997]
Adams, Thomas; Dworkin, Steve: Workflow Interoperability between businesses, in: Lawrence, P. (ed.): Workflow Handbook 1997, Wiley, Chichester etc., 1997, pp. 211-221.

[Agostini et al. 1996]
Agostini, A.; de Michelis, Giorgio; Grasso, Maria A.; Prinz, Wolfgang; Syri, Anja: Contexts, Work Processes, and Workspaces, in: Computer Supported Cooperative Work: The Journal of Collaborative Computing, Kluwer Academic Publishers, Netherlands, Vol. 5, Nos. 2/3, 1996, pp. 223-250.

[Albach/Wildemann 1995]
Albach, Horst; Wildemann, Horst: Lernende Unternehmen, Zeitschrift für Betriebswirtschaftslehre, Ergänzungsheft, Gabler, Wiesbaden, 3, 1995.

[Allweyer 1995]
Allweyer, T.: Modellierung und Gestaltung adaptiver Geschäftsprozesse, in: Veröffentlichung des Instituts für Wirtschaftsinformatik, Saarbrücken, Heft 115, 1995.

[Amberg 1996]
Amberg, Michael: Modeling adaptive workflows in distributed environments, in: Proc. of first intl. conf. on practical aspects of knowledge management, Basel, Switzerland, 1996.

[Ang 1996]
Ang, J.S.K.: A comprehensive office modeling framework for ITHACA, in: ITHACA, Bull, 89, D9, #1, Version 0, June, 1996.

[Ang/Conrath 1993]
Ang, J.S.K.; Conrath, D.W.: The ITHACA Office Object Model: Modeling and Implementation, in: Data Base, Bd. 24, H. 4, 1993, pp. 5-14.

[Argyris/Schön 1978]
Argyris, C.; Schön, D.: Organizational Learning, A Theory of Action Perspective, Reading, Mass., 1978.

[Bach/Brecht/Österle 1995]
Bach, Volker; Brecht, Leo; Österle, Hubert: Software-Tools für das Business Process Redesign, Hochschule St. Gallen, Institut für Wirtschaftsinformatik, FBO-Verlag, Baden-Baden, 1995.

[Baligh/Burton 1981]
Baligh, Helmy H.; Burton, R.M.: Describing and designing organizational structures and processes, in: International Journal of Policy Analysis and Information Systems, 5, 1981, pp. 251-266.

[Baligh/Burton/Obel 1990]
Baligh, Helmy H.; Burton, R.M.; Obel, B.: Devising Expert Systems in Organization Theory: The Organizational Consultant, in: Masuch, M.(ed.): Organization, Management, and Expert Systems. Models of Automated Reasoning, de Gruyter, Berlin - New York, 1990, pp. 35-57.

[Baligh/Burton/Obel 1994]
Baligh, Helmy H.; Burton, Richard M.; Obel, Børge: Validating an Expert System That Designs Organizations, in: Carley, K.M.; Prietula, M.J. (eds.): Computational Organization Theory, Lawrence Erlbaum, Hillsdale, New Jersey, 1994, pp. 179-193.

[Baligh/Damon 1980]
Baligh, Helmy H.; Damon, William W.: Foundations for a systematic process of organizational structure design, in: Journal of Information & Optimization Sciences, 1, 1980, pp. 133-165.

[Bannon/Schmidt 1991]
Bannon, Liam J.; Schmidt, Kjeld: CSCW: Four characters in search of a context, in: Bowers, J.M.; Benford, S.D. (eds.): Studies in Computer Supported Cooperative Work, Theory, Practice and Design, Elsevier, Nort Holland, 1991., Elsevier, North-Holland, 1991, pp. 3-16.

[Bannon/Schmidt 1997]
Bannon, Liam J.; Schmidt, Kjeld: Issues of Supporting Organizational Context in CSCW Systems, in: COMIC-Deliverable report, 1.1, 1997.

[Batini et al. 1992]
Batini, C.; Catarci, T.; Costabile, M.F.; Levialdi, S.: Visual Query Systems: A Taxonomy, in: Knuth, E., Wegner, L.M. (eds.): Visual Database Systems, II, Elsevier, North-Holland, 1992, pp. 153-168.

[Bauer 1998]
Bauer, Hans-Peter: Die Bedeutung von Knowledge - Management in der Wirtschaft, Director Consulting, Lotus Development, 1. Groupware Hochschultage, Groupware Competence Center at the University of Paderborn, May, 1998.

[BDL 1998]
Babcock Dienstleistungs-GmbH: Produkte und Leistungen, Oberhausen, May, 1998.

[Becker/Rosemann/Schütte 1995]
Becker, Jörg; Rosemann, Michael; Schütte, Reinhard: Grundsätze ordnungsgemäßer Modellierung, in: Wirtschaftsinformatik 37, Nr. 5, Oktober, 1995, pp. 435-445.

[Becker/Vogler/Österle 1998]
Becker, Matthias; Vogler, Petra; Österle, Hubert: Workflow-Management betriebs-wirtschaftlicher Standardsoftware, in: Wirtschaftsinformatik, 40, 4, 1998, pp. 318-328.

[Bender/Ficks/Bender 1986]
Bender, Henry E.; Ficks, F.L.; Bender, Judy: A Case Study of Productivity Improvement Resulting from Organizational Restructuring, in: Brown, O.; Hendrick, H. W. (eds.): Proceedings of the Second Symposium on Human Factors in Organization Design and Management, Elsevier Science Publishers (North Holland), Vancover, BC, AUG. 19-21, 1986, pp. 613-620.

[BIFOA 1993]
BIFOA (ed.): Vorgangsmanagementsysteme, BIFOA-Marktstudie, BIFOA-Selbstverlag, Köln, 1993.

[Bleicher 1991]
Bleicher, Knut: Organisation, Formen und Modelle, Wiesbaden, 1991.

[Bowers 1994]
Bowers, John: The work to make a network work, Studying CSCW in action, in: Proceedings of the CSCW'94, ACM Press, Chapel Hill, NC, Oktober, 1994, pp. 287-298.

[Bracchi/Pernici 1984]
Bracchi, G.; Pernici, B.: The Design Requirements of Office Systems, in: ACM Transaction on Office Inform. Systems, Vol.2, No. 2, 1984, pp. 151-170.

[Bradley/Hausmann/Nolan 1993]
Bradley, Stephen P.; Hausmann, Jerry A.; Nolan, Richard L.: Globalization, Technology and Competition, The Fusion of Computer and Telecommunications in the 1990s, Harvard Business School Press, Boston, Massachusetts, 1993.

[Breitbart et al. 1993]
Breitbart, Y.; Deacon, A.; Schek, H.-J:; Sheth, Amit; Weikum, G.: Merging Application-centric and Data-centric Approaches to support Transaction-oriented Multi-system Workflows, in: SIGMOD RECORD, Vol. 22, No. 3, 1993.

[Brunner in prep.]
Brunner, Kai: A Virtual Center of Competence 'Organization Structure', Masters Thesis, University of Paderborn, Department of Business Computing 2, in preparation.

[Bungard/Antoni 1993]
Bungard, W.; Antoni, E. Lehnert: Gruppenarbeitskonzepte mittlerer Industriebetriebe, Ehrenhof Verlag, 1993.

[Burnett 1996]
Burnett, Steve: How RSA Works, An RSA Data Security, Inc. Engineering Brief, RSA Data Security, Inc., 1996.

[Burrell/Morgan 1979]
Burrell, G.; Morgan, G.: Sociological Paradigms and Organizational Analysis, Heinemann, London, 1979.

[Bußler 1992]
Bußler, Christoph: Capability Based Modelling, in: Proceedings of the First International Conference on Enterprise Integration Mdeling Technology (ICEIMT), Hilton Head, SC, USA, June, 1992.

[Bußler/Jablonski 1994]
Bußler, Christoph; Jablonski, Stefan: An approach to integrate workflow modeling and organization modeling in an enterprise, in: Proceedings of 3rd IEEE International Workfshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, 1994, pp. 81-95.

[B.BIT 1992]
B.BIT CONSULT GmbH: Geschäftsprozeßmanagement, Vorgangssteuerungssysteme und integrierte Vorgangsbearbeitung, FBO Verlag, Baden-Baden, 1992.

[Calabria/Plumley 1997]
Calabria, Jane; Plumley, Sue: Lotus Notes 4.5 and the Internet 6 in 1, QUE, Indianapolis, 1997.

[CCITT 1985]
CCITT Red Book, Vol. III, VIII.7, Message Handling Systems, International Telecommunications Union, Geneva, 1985.

[Chadwick 1994]
Chadwick, David: Understanding Directories, Chapman Hall, London, 1994.

[Chandler 1962]
Chandler, A.D.: Strategy and Structure, Cambridge etc., 1962.

[Chapple/Sayles 1981]
Chapple, E.; Sayles, L.: Work flow as the basis for organization design, in: Jelinek, M.; Litterer, J. A.; Miles, R. E. (eds.): Organizations by design: Theory and Practice, Business Publications, Inc., Plano, TX, 1981, pp. 386-406.

[Charan 1991]
Charan, Ram: How networks reshape organizations - for results, in: Harvard Business Review, September-October, 1991, pp. 104-115.

[Chen 1976]
Chen, P.P.-S.: The Entity Relationship Model, Toward a Unified View of Data, in: ACM Transactions on Database Systems, ACM, Vol. 1, No. 1, 1976, pp. 9-36.

[Chrapary/Rosenow-Schreiner/Waldhör 1991]
Chrapary, H.J.; Rosenow-Schreiner, Elke; Waldhör, K.: Das elektronische Organisationshandbuch, in: Lutze, R., Kohl, A.(eds.): Wissensbasierte Systeme im Büro: Ergebnisse aus dem WISDOM-Verbundprojekt, Oldenbourg Verlag, München, Wien, 1991, pp. 295-312.

[Clark/Schiano 1996]
Clark, Theodore H.; Schiano, William T.: Seven Levels of Interorganizational Connectivity, An Examination of the U.S. Grocery Distribution Channel, in: Nunamaker jr., J. F. et al. (eds.): Proceedings of the 29th Annual Hawaii International Conference on System Science, IEEE Computer Society Press, Los Alamitos, 1996, pp. 281-289.

[Computerwoche 1995]
Marktstudie untersucht Tools zum Business Process Redesign, in: Computerwoche, 18, 1995, p. 18.

[Conroy III 1986]
Conroy III, Joseph: Systematic Organizational Design (SORD) Method, in: Brown, O.; Hendrick, H. W. (eds.): Proceedings of the Second Symposium on Human Factors in Organization Design and Management, Elsevier Science Publishers (North Holland), Vancover, BC, AUG. 19-21, 1986, pp. 607-612.

[Coulson-Thomas 1994]
Coulson-Thomas, C.: Business Process Re-engineering. Myth and Reality, Kogan Page, London, 1994.

[Crowston 1994]
Crowston, Kevin: Evolving Novel Organizational Forms, in: Carley, K.M.; Prietula, M.J. (eds.): Computational Organization Theory, Lawrence Erlbaum, Hillsdale, New Jersey, 1994, pp. 19-38.

[Cyert/March 1963]
Cyert, R.; March, J.G.: A behavioral Theory of the Firm, Englewood Cliffs, N.J., 1963.

[Damm 1994]
Damm, Andreas Wolfgang: Organisatorische Problemanalyse, Entwicklung eines Verfahrens zur Analyse und Strukturierung von Problemen in einer Organisation, Dissertation, Karlsruhe, 18. Juli, 1994.

[Davenport 1993]
Davenport, Thomas H.: Process Innovation, Reengineering Work through Information Technology, Harvard Business School Press, Boston, 1993.

[Davenport/Short 1990]
Davenport, Thomas H.; Short, James E.: The New Industrial Engineering: Information Technology and Business Process Redesign, in: Sloan Management Review, Boston, Summer, 1990, pp. 11-27.

[Davidow/Malone 1992]
Davidow, William H.; Malone, Michael S.: The Virtual Corporation, Structuring and Revitalizing the Corporation for the 21st Century, Harper Business, NY, 1992.

[Dayal/Hsu/Ladin 1991]
Dayal, U.; Hsu, Meichun; Ladin, R.: A Transactional Model for Long-Running Activities, in: Proceeedings of the 17th International Conference on Very Large Databases, Barcelona, Spain, September, 1991, pp. 113-122.

[de Greene 1986]
de Greene, Kenyon B.: Systems theory, Macroergonomics, and the design of Adaptive Organization, in: Brown, O.; Hendrick, H. W. (eds.): Proceedings of the Second Symposium on Human Factors in Organization Design and Management, Elsevier Science Publishers (North Holland), Vancover, BC, AUG. 19-21, 1986, pp. 429-492.

[Deiters/Gruhn/Striemer 1995]
Deiters, Wolfgang; Gruhn, Volker; Striemer, Rüdiger: Der FUNSOFT-Ansatz zum integrierten Geschäftsprozeßmanagement, in: Wirtschaftsinformatik, 37, 5, 1995, pp. 459-466.

[Deiters et al. 1996]
Deiters, Wolfgang; Herrmann, Thomas; Löffeler, Thorsten; Striemer, Rüdiger: Identifikation, Klassifikation und Unterstützung semi-strukturierter Prozesse in prozeßorientierten Telekooperationssystemn, in: Krcmar, H., Lewe, H., Schwabe, G. (eds.): Herausforderung Telekooperation - Proceedings of the German Computer Supported Cooperative Work Conference '96, Springer Verlag, Berlin, Heidelberg etc., 1996, pp. 261-274.

[Derungs/Vogler/Österle 1995]
Derungs, Marc; Vogler, Petra; Österle, Hubert: Metamodell Workflow, Report Nr. HSG/CC PSI/3, Version 1.0,1.5, http://www-iwi.unisg.ch/iwi2/cc/psi/abpsi3.html, St. Gallen, 1995.

[Desai 1991]
Desai, S. M.: Unification of Underlying Concepts in Different Office Models, DataBase, Winter/Spring, 1991, pp. 38-45.

[Dhar/Olson 1989]
Dhar, V.; Olson, M.: Assumptions underlying sytems that support work group collaboration, in: Technological Support for Work Group Collaboration, Lawrence Erlbaum Associates, Hillsdale, NJ, 1989, pp. 33-50.

[Dier/Lautenbacher 1994]
Dier, M.; Lautenbacher, S.: Groupware: Technologien für die lernende Organisation, Rahmen, Konzepte, Fallstudien, Computerwoche Verlag GmbH, München, 1994.

[Dodswell 1983]
Dodswell, A.: Office Automation, Wiley, Chichester, England, 1983.

[Domsch/Reinecke 1989]
Domsch, M.; Reinecke, P.: Bewertungstechniken, in: Szyperski, N. (Ed.): Handwörterbuch der Planung, Poeschel Verlag, Stuttgart, Bd. 9, 1989, pp. 143-156.

[Dourish/Bellotti 1992]
Dourish, P.; Bellotti, V.: Awareness and coordination in shared workspaces, Proceedings of the CSCW'92 (Toronto, Canada), ACM/SIGCHI, New York, Oktober, 1992, pp. 25-38.

[Drucker 1988]
Drucker, Peter Ferdinand: The coming of the new organisation, in: Harvard Business Review, 1-2, 1988, pp. 45-53.

[Drucker 1991]
Drucker, Peter Ferdinand: The new productive challenge, in: Harvard Business Review, November-December, 1991, pp. 70-79.

[Drumm 1996]
Drumm, Hans Jürgen: Das Paradigma der neuen Dezentralisierung, in: DBW, Die Betriebswirtschaft, Schäffer Poeschel, 1, 1996, pp. 7-20.

[ECMA 1985]
ECMA: Standard ECMA-101, Office Document Architecture, European Computer Manufacturers Association, Geneva, 1985.

[Ehlers 1997]
Ehlers, Peter: Integriertes Projekt- und Prozeßmanagement auf Basis innovativer Informations- und Kommunikationstechnologien: Das GroupProject-System, Referenzrahmen, Architekturen, Konzepte, Systemdesign und empirische Einsatzerfahrungen eines verteilten, prozeßorientierten Projektinformationssystems, Dissertation, Universität Paderborn, Juni, 1997.

[Eisenführ 1993]
Eisenführ, Franz: Personalwirtschaft und Organisation, Skript zur Vorlesung, Verlag der Augustinus Buchhandlung, 1, Aachen, 1993.

[Ellis 1983]
Ellis, Clarence A.: Formal and Informal Models of Office Activity, in: Mason, R.E.A. (ed.): Information Processing, North-Holland Publishing Co., 1983, pp. 11-22.

[Ellis/Gibbs/Rein 1991]
Ellis, Clarence A.; Gibbs, S. J.; Rein, Gail Louise: Groupware, some issues and experiences, in: Communications of the ACM, 34 (1), 1991, pp. 38-58.

[Ellis/Keddara/Rozenberg 1995]
Ellis, Clarence A.; Keddara, Karim; Rozenberg, Gzregorz: Dynamic Change Within Workflow Systems, in: Comstock, N. et al. (eds.): Conference on Organizational Computing Systems, ACM Press, Milpitas, 1995, pp. 10-21.

[Ellis/Nutt 1980]
Ellis, Clarence A.; Nutt, G.J.: Office Information Systems and computer science, in: Computing Surveys, Vol.12, No.1, 1980, pp. 27-60.

[Emery/Trist 1965]
Emery, F.E.; Trist, E.L.: The Causal Texture of Organizational Environments, in: Human Relations, 21-32, Vol. 18, 1965.

[Engelbart 1990]
Engelbart, Douglas: Knowledge-Domain interoperability and an open hyperdocument system, in: CSCW'90 Proceedings, ACM, Los Angeles, USA, 1990, pp. 143-156.

[Esswein 1993]
Esswein, Werner: Das Rollenmodell der Organisation: Die Berücksichtigung aufbauorganisatorischer Regelungen in Unternehmensmodellen, in: Wirtschaftsinformatik, Vieweg, Wiesbaden, 35, 6, 1993, pp. 551-561.

[Fahrwinkel 1995]
Fahrwinkel, Uta: Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des Business Process Reengineering, Dissertation, Heinz Nixdorf Institut, Universität Paderborn, 1995.

[Faidt et al. 1990]
Faidt, K.; Grünberger, H.; Heller, H.; Hinkelmann, K.; Karagiannis, D.; Salzmann, W.; Schebiella, S.: Organization Support System - OSSY-, Der Demonstrator, Projektbericht, FAW Ulm, February, 1990.

[Ferstl/Sinz 1990]
Ferstl, Otto K.; Sinz, Elmar J.: Objektmodellierung betrieblicher Informationssysteme im Semantischen Objektmodell (SOM), in: Wirtschaftsinformatik, 6, 1990, pp. 566-581.

[Ferstl/Sinz 1995]
Ferstl, Otto K.; Sinz, Elmar J.: Der Ansatz des Semantischen Objektmodells (SOM) zur Modellierung von Geschäftsprozessen, in: Wirtschaftsinformatik 37, 3/95, 1995, pp. 209-220.

[Ferstl et al. 1994]
Ferstl, Otto K.; Sinz, Elmar J.; Amberg, M.; Hagemann, U.; Malischewski, C.: Tool-based Business Process Modeling Using the SOM Approach, in: Wolfinger, B. (Ed.): Innovationen bei Rechen- und Kommunikationssystemen, Springer, Berlin, Proc. 24, GI-Jahrestagung im Rahmen des 13^{th} World Computer Congress, Hamburg, 28.8.-2.9., 1994, pp. 430-436.

[Ficks 1986]
Ficks, F.L.: Inserting Design into the Organization Process, in: Brown, O.; Hendrick, H. W. (eds.): Proceedings of the Second Symposium on Human Factors in Organization Design and Management, Elsevier Science Publishers (North Holland), Vancover, BC, AUG. 19-21, 1986, pp. 621-628.

[Fikes 1982]
Fikes, Richard E.: A Commitment-Based Framework for Describing Informal Cooperative Work, in: Cognitive Science, 6, 1982, pp. 331-347.

[Fox 1992]
Fox, M. S.: The TOVE Project: Towards a Common Sense Model of the Enterprise, in: Petrie, C. (ed.): Enterprise Integration, MIT Press, Cambridge, Mass, 1992.

[Fox 1993]
Fox, M. S.: Issues in Enterprise Modelling, in: IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, 1993, pp. 86-92.

[Fox/Barbuceanu/Gruninger 1996]
Fox, M. S.; Barbuceanu, M.; Gruninger, M.: An Organisation Ontology for Enterprise Modelling, Preliminary Concepts for Linking Structure and Behaviour, in: Computers in Industry, Vol. 29, 1996, pp. 123-134.

[Fox/Grüninger 1997]
Fox, M. S.; Grüninger, M.: On Ontologies and Enterprise Modelling, in: Proceedings International Conference on Enterprise Integration Modelling Technology 97, Springer-Verlag, 1997.

[Fraser 1994]
Fraser, John: Managing change through Enterprise Models, in: Milne, R.; Montgomery, A. (eds.): Applications and Innovations in Expert Systems II, SGES Publications, Cambridge, 1994.

[Frese 1992]
Frese, Erich: Organisationstheorie, Historische Entwicklung - Ansätze - Perspektiven, 2. Aufl., Gabler, Wiesbaden, 1992.

[Fuchs et al. 1996]
Fuchs, Ludwin; Sohlenkamp, Markus; Genau, Andreas; Kahler, Helge; Pfeifer, Andreas; Wulf, Volker: Transparenz in kooperativen Prozessen: Der Ereignisdienst in POLITeam, in: Krcmar, H., Lewe, H., Schwabe, G. (eds.): Herausforderung Telekooperation - Proceedings of the German Computer Supported Cooperative Work Conference '96, Springer Verlag, Berlin etc., 1996, pp. 3-16.

[Gable 1991]
Gable, J.: Workflow Software, 1991.

[Gaitanides 1983]
Gaitanides, Michael: Prozeßorganisation, Entwicklung, Ansätze und Programme prozeßorientierter Organisationsgestaltung, Vahlen, München, 1983.

[Gaitanides/Scholz/Vrohlings 1994]
Gaitanides, Michael; Scholz, Rainer; Vrohlings, Alwin: Prozeßmanagement - Grundlagen und Zielsetzung, in: Gaitanides, M. u.a. (eds.): Prozeßmanagement. Konzepte, Umsetzungen und Erfahrungen mit Reengineering, Hanser, München, Wien, 1994, pp. 1-19.

[Galbraith 1973]
Galbraith, J. R.: Designing Complex Organizations, Addison Wesley, Reading, Mass., 1973.

[Galbraith 1977]
Galbraith, J. R.: Organization Design, Addison-Wesley, Reading, MA, 1977.

[Galler 1995]
Galler, Jürgen: Metamodelle des Workflow Managements, in: Scheer, A.W. (ed.): Veröffentlichungen des Instituts für Wirtschaftsinformatik, Heft 121, 12, 1995.

[GartnerGroup 1997a]
GartnerGroup: Benefits of Groupware from a Business Perspective, Key-Note-Speech at the 7th DNUG conference by Regina Casonato, Vice President and Research Director, GartnerGroup (Mailand/Italy), Frankfurt/Main, Germany, November 13, 1997.

[GartnerGroup 1997b]
GartnerGroup: Get Ready for Enterprise Work Management, Workshop Presentation at the 7 th DNUG conference by Regina Casonato, Vice President and Research Director, GartnerGroup (Mailand/Italy), Frankfurt/Main, Germany, November 13, 1997.

[Georgakopoulos/Hornick/Sheth 1995]
Georgakopoulos, Diimitrios; Hornick, Mark; Sheth, Amit: An Overview of Workflow Management, From Process Modeling to Workflow Automation Infrastructure, in: Journal on Distributed and Parallel Database Systems, Kluwer Academic Publishers, Boston, 3 (2), April, 1995, pp. 119-153.

[Gerstein/Shaw 1994]
Gerstein, Marc S.; Shaw, Robert B.: Organisations-Architekturen für das einundzwanzigste Jahrhundert, in: Nadler, David A.; Gerstein, M. S.; Shaw, R. B. (eds): Organisations-Architektur, Optimale Strukturen für Unternehmen im Wandel, Campus Verlag, Frankfurt, New York, 1994., Campus Verlag, Frankfurt, New York, 1994, pp. 262-272.

[Gessford 1991]
Gessford, John E.: How to Build Business-Wide DATABASES, Wiley & Sons, Inc, 1991.

[GfürO 1996]
Gesellschaft für Organisation e.V.: Organisationales Lernen, Zeitschrift für Organisaton, Ergänzungsheft, Schäffer-Poeschel, Stuttgart, 1, 1996.

[Gieffers 1995]
Gieffers, Stephan: Vorstudie zur Analyse von Organisationsmodellierungskomponenten von Workflow Management Systemen, Studienarbeit, Paderborn, Dezember, 1995.

[Gordon 1997]
Gordon, Mark: Accessing the Domino directory with LDAP, 1997.

[Götzer 1995]
Götzer, Klaus: Innovative Organisationsstrukturen mit Office Reengineering, FBO Office Media, Wiesbaden, 1995.

[Grabowski et al. 1996]
Grabowski, Hans; Grein, Gunther; Milde, Peter; Weber, Ulrich: Conceptual Design of Information Systems based on Enterprise Modelling, in: Bernus, P.; Nemes, L. (eds.): Modelling and Methodologies for Enterprise Integration, Chapman & Hall, London, 1996, pp. 113-125 .

[Gray/Reuter 1991]
Gray, Jim; Reuter, Andreas: Transaction Processing: Concepts and Techniques, Morgan Kaufman, 1991.

[Grochla 1975]
Grochla, E.: Entwicklung und gegenwärtiger Stand der Organisationstheorie, in: Grochla, E. (ed.): Organisationstheorie, 1. Teilband, 1975, pp. 2-32.

[Grochla 1982]
Grochla, E.: Grundlagen der organisatorischen Gestaltung, Stuttgart, 1982.

[Groiss/Eder 1997]
Groiss, Herbert; Eder, Johann: Bringing Workflow Systems to the Web, URL: http://www.ifi.uni-klu.ac.at/~herb/Overview.html, Klagenfurt, 1997.

[Grudin 1990]
Grudin, Jonathan: The computer reaches out, The historical continuity of interface design, in: Carrasco-Chew, J., Whiteside, J. (eds.): CHI '90, Proceedings Conference on Computer Human Interaction, ACM, New York, NY, 1990, pp. 261-268.

[Grudin 1994]
Grudin, Jonathan: Groupware and Social Dynamics: Eight Challenges for Developers, in: Communications of the ACM, January, Vol. 37, No. 1, 1994, pp. 92-105.

[Güldenberg/Eschenbach 1996]
Güldenberg, Stefan; Eschenbach, Rolf: Organisatorisches Wissen und Lernen, erste Ergebnisse einer qualitativ-empirischen Erhebung, in: zfo, 1, 1996, pp. 4-9.

[Hackman/Oldham 1976]
Hackman, J. Richard; Oldham, Greg R.: Motivation through the design of work, Test of a theory, in: Organizational behavior and human performance, 1976, pp. 250-279.

[Hadamitzky 1995]
Hadamitzky, Michael C.: Restrukturierung, organisatorisches Lernen und Unternehmenserfolg, in: Zeitschrift für Betriebswirtschaftslehre, Ergänzungsheft, Gabler, Wiesbaden, 3, 1995, pp. 173-189.

[Hagemeyer et al. 1997]
Hagemeyer, Jens; Herrmann, Thomas; Just-Hahn, Katharina; Striemer, Rüdiger: Flexibilität bei Workflow-Management Systemen, in: Tagungsband "Software Ergonomie 97", Dresden, 3.3.-6.3., 1997, pp. 179-190.

[Hammer 1990]
Hammer, Michael: Reengineering Work: Don't Automate, Obliterate, Managers can release the real power of computers by challenging centuries-old notions about work, Harvard Business Review, July/August, 1990, pp. 104-112.

[Hammer/Champy 1993]
Hammer, Michael; Champy, James: Reengineering the Corporation, A Manifesto For Business Revolution, Harper Business, New York, 1993.

[Hansen 1991]
Hansen, W.C.: The Integrated Enterprise, in: Foundations of World-Class Manufacturing Systems: Symposium Papers, National Academy of Engineering, 2101 Constitution Ave, N.W., Washington DC, 1991.

[Harrington 1991]
Harrington, H. James: Business Process Improvement: The Breaktrough Strategy for Total Quality, Productivity, and Competitiveness, McGraw-Hill, New York, 1991.

[Hars 1994]
Hars, Alexander: Referenzdatenmodelle, Grundlagen effizienter Datenmodellierung, Wiebaden, 1994.

[Hartmann/Kahler/Wulf 1993]
Hartmann, A.; Kahler, H.; Wulf, V.: Groupware - Probleme und Gestaltungsoptionen (2), in: Office Management, 12, 1993, pp. 64-79.

[Hasenkamp 1987]
Hasenkamp, Ulrich: Konzipierung eines Bürovorgangssystems - Informations- und Kommunikationstechnik zur aktiven Steuerung von Bürovorgängen, Habilitationsschrift, Wirtschafts- und Sozialwissenschaftliche Fakultät, Universität Köln, 1987.

[Hasenkamp/Syring 1993]
Hasenkamp, Ulrich; Syring, Michael: Workflow-Management, Beispiele, in: Office Management, Juni, 1993, pp. 32-36.

[Hasenkamp/Syring 1994]
Hasenkamp, Ulrich; Syring, Michael: CSCW in Organisationen - Grundlage und Probleme, in: Hasenkamp, U. et al. (Hrsg.): CSCW - Computer Supported Cooperative Work, Informationssysteme für dezentralisierte Unternehmensstrukturen, Addison-Wesley, Bonn etc., 1994, pp. 15 - 37.

[Hasenkamp/Syring/Kirn 1994]
Hasenkamp, Ulrich; Syring, Michael; Kirn, Stefan: CSCW - Computer Supported Cooperative Work, in: Hasenkamp, Ulrich; Kirn, Stefan; Syring, Michael: CSCW - Computer Supported Cooperative Work, Informationssysteme für dezentralisierte Unternehmensstrukturen., 1994, pp. 5-12.

[Hawker 1996]
Hawker, Marion: Secrets to Running Lotus Notes, The Decisions No One Tells You to Make, Lotus / IBM, Research Triangle Park, 1996.

[Heath/Luff 1993]
Heath, C.; Luff, P.: Collaborative activity and technological design, Task coordination in London underground control rooms, in: Proceedings of the third european confenrence on Computer Supported Cooperative Work - ECSCW'93 (Amsterdam), Kluwer, Dordrecht, September, 1993, pp. 65-80.

[Heilmann 1994]
Heilmann, Heidi: Workflow Management: Integration von Organisation und Informationsverarbeitung, in: HMD, Theorie und Praxis der Wirtschaftsinformatik, Schwerpunktthema "Workflow Management", Heft 176, März, 1994, pp. 8-21.

[Heilmann 1996]
Heilmann, Heidi: Die Integration der Aufbauorganisation in Workflow-Management-Systeme, in: Heilmann, Heidi; Heinrich, Lutz J.; Roithmayer, Friedrich (eds.): Information Engineering, Oldenbourg, München, Wien, 1996, pp. 147-165.

[Heilmann/Simon 1989]
Heilmann, Heidi; Simon, M.: Organisationsanalyse und -planung mit ODB/OIS: Integration mit bestehenden Anwendungssystemen im Unternehmen, in: Paul, M. (ed.): Proceedings der 19. GI - Jahrestagung, Bd. II, Computergestützter Arbeitsplatz, München, 18.-20. Oktober 1989, Springer, Berlin, Heidelberg, New York, 1989, pp. 190-203.

[Heinz 1995]
Heinz, Oliver: Enterprise Information Management Model as Basis for Groupware based Workflow Management, Seminar paper, University of Paderborn, October, 1995.

[Heinz 1997]
Heinz, Oliver: Standards for Groupware-based Organizational Subsystems within Information Systems - the Impact on Workflow-supporting Office Platforms, Masters Thesis, University of Paderborn, February, 1997.

[Heinz/Ott 1997]
Heinz, Oliver; Ott, Marcus: X.500 Directory and LDAP Standards in dynamic Groupware-based Organizational Subsystems, Working Paper, University of Paderborn, Department of Business Computing, February, 1997.

[Heinzl/König 1993]
Heinzl, Armin; König, Wolfgang: Erfahrungen bei der computergestützten Organisationsgestaltung am Beispiel des Systems ORSYS, in: Information Management, 3/93, 1993, pp. 16-22.

[Hennessy/Kreifelts/Ehrlich 1993]
Hennessy, Pippa; Kreifelts, Thomas; Ehrlich, Ute: Distributed work management, Activity coordination in the EuroCoOp project, in: Computer Communications, 15 (8), 1993, pp. 477-488.

[Herrmann 1995]
Herrmann, Thomas: Workflow Management Systems: Ensuring organizational Flexibility by Possibilities of Adaption and Negotiation, in: Proc. Conference on Organizational Computing Systems (COOCS) '95, ACM, New York, 1995, pp. 83-95.

[Hilpert 1994]
Hilpert, Wolfgang: Effiziente und wirtschaftliche Einsatzplanung von Workflow Management-Systemen auf Groupware-Basis - vom Pilotprojekt zur unternehmensweiten Lösung, Arbeitspapier, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2 und PAVONE Informationssysteme GmbH, Paderborn, 1994.

[Hilpert/Riempp/Nastansky 1994]
Hilpert, Wolfgang; Riempp, Gerold; Nastansky, Ludwig: The GroupFlow System: Workflow Management in Distributed Organizations, White Paper, University of Paderborn, Business Computing, 1994.

[Hinrichs 1996]
Hinrichs, Jochen: Information Object Modeler, CASE tool for Database Design under Lotus Notes, Masters Thesis, University of Paderborn, Department of Business Computing 2, 1996.

[Hirschheim 1985]
Hirschheim, Rudy A.: Office Automation: A Social and Organizational Perspective, John Wiley & Sons, Chichester, 1985.

[Hoffmann 1989]
Hoffmann, Friedrich: Erfassung, Bewertung und Gestaltung von Organisationstrukturen, Ein anwendungsorientierter Ansatz, in: Zeitschrift für Organisation (zfo), 5, 1989, pp. 333-338.

[Hoischen/Otto 1997]
Hoischen, Markus; Otto, Rüdiger: Business Process Reengineering-Werkzeuge und Modelle zur Organisationsgestaltung, Dokumentation, Bewertung und Vergleich von Business Process Reengineering-Werkzeugen und deren mögliche Verknüpfung mit Workflow Management-Systemen, Diplomarbeit, Universität Paderborn, April, 1997.

[Hoppen 1992]
Hoppen, Dirk: Organisation und Informationstechnologie, Grundlagen für ein Konzept zur Organisationssystemgestaltung, Dr. Kovac, Hamburg, 1992.

[Horak/Hoffmann 1986]
Horak, W.; Hoffmann, K.: Office Document Architecture and Office Document Interchange Formats, Elsevier Science of the European Communities, in: ESPRIT '85: Status Reports of Continuing Work, Elsevier, 1986.

[Houghton Mifflin Company 1994]
Houghton Mifflin Company: American Heritage Dictionary of the English Language, SoftKey International Inc., Third Edition, 1994.

[Hummel/Schoder/Strauss 1996]
Hummel, Thomas; Schoder, Detlef; Strauss, R.E.: Why CSCW Applications Fail, Problems in the Support of Lateral Cooperation and the Appropriatness of CSCW Applications, in: Proceedings of the Annual Meeting of the Northeast Decision Sciences Institute (NEDSI), St. Croix, USA, 17-19 April, 1996.

[Huse 1980]
Huse, E.F.: Organization development and change, St. Paul, 2nd ed., 1980.

[Huth 1998]
Huth, Carsten: Plattform- und werkzeugübergreifende verteilte Organisationsmodellierung für Workflow Management, Entwicklung einer Werkzeugunterstützung zur Modellierung von Organisationssubsystemen für die Arbeit verteilter Teams, Masters Thesis, University of Paderborn, Department of Business Computing 2, 1998.

[IBM 1996]
IBM - International Business Machines: Lotus Notes Release 4.5: A Developer's Handbook, edited by Marion Hawker, Joe Arteaga, Gerald Krause, Dave Morrison, Hiroki Nakamura, Reinhold Strobl, International Technical Support Organization, IBM, First Edition, NC, 1996.

[Ishii/Kubota 1989]
Ishii, Hiroshi; Kubota, Kazunari: Office Procedure Knowledge Base for Organizational Office Work Support, in: Pernici, B.; Verrijn-Stuart, A.A. (eds.): Office Information Systems: The Design Process, Working Conference, IFIP WG 8.4, August 15-17, 1988, Linz, Austria, Elsevier Science Publishers B.V., North-Holland, 1989, pp. 55-72.

[Ishii/Ohkubo 1991]
Ishii, Hiroshi; Ohkubo, M.: Message-driven groupware design based on an office procedure model, OM-1, in: Journal of Information Processing, Japan, Vol. 14, No. 2, 1991, pp. 184-191.

[ISO/IEC 1993a]
ISO/IEC: Information technology - Open Systems Interconnection, The Directory: Models, Recommendation X.501, ISO/IEC 9594-2, 1993.

[ISO/IEC 1993b]
ISO/IEC: Information technology - Open Systems Interconnection, The Directory: Selected Object Classes, Recommendation X.521, ISO/IEC 9594-7, 1993.

[ISO/IEC 1993c]
ISO/IEC: Information technology - Open Systems Interconnection, The Directory: Authentication Framework, Recommendation X.509, ISO/IEC 9594-8, 1993.

[ISO/IEC 1993d]
ISO/IEC: Information technology - Open Systems Interconnection, The Directory: Replication, Recommendation X.525, ISO/IEC 9594-9, 1993.

[ISO/IEC 1993e]
ISO/IEC: Information technology - Open Systems Interconnection, The Directory: Protocol Specifications, Recommendation X.519, ISO/IEC 9594-5, 1993.

[ISO/IEC 1993f]
ISO/IEC,: Information technology - Open Systems Interconnection, The Directory: Selected Attribute Types, Recommendation X.520, ISO/IEC 9594-6, 1993.

[Jablonski 1992]
Jablonski, Stefan: Data Flow Management in Distributed C.I.M. Systems, in: Proceedings 3rd International Conference on Data and Knowledge Systems for Manufacturing and Engineering, March, 1992, pp. 65ff.

[Jablonski 1995a]
Jablonski, Stefan: Workflow-Management-Systeme, Modellierung und Architektur, Internat. Thomson Publ., Bonn, 1995.

[Jablonski 1995b]
Jablonski, Stefan: Workflow-Management-Syteme: Motivation, Modellierung, Architektur, in: Informatik Spektrum, Springer-Verlag, 18, 1995, pp. 13-24.

[Jablonski 1997]
Jablonski, Stefan: Architektur von Workflow Management Systemen, in: Informatik, Forschung und Entwicklung, Band 12, Heft 2, 1997, pp. 72-81.

[Jablonski/Bußler 1996]
Jablonski, Stefan; Bußler, Christoph: Workflow Management, Modeling Concepts, Architecture and Implementation, International Thomson Computer Press, London etc., 1996.

[Jaschik/Lang 1997]
Jaschik, Gunnar; Lang, Florian: Konzept der Vertreterregelung in der heutigen Arbeitswelt - Übertragung auf WfMS, Project and Seminar Work, University of Paderborn, Department of Business Computing, October, 1997.

[Jirotka/Gilbert/Luff 1992]
Jirotka, M.; Gilbert, N.; Luff, P.: On the social organisation of organisations, in: CSCW 1-2, 1992, pp. 95-118.

[Johansen 1988]
Johansen, Robert: Groupware, Computer Support for Business Teams, New York, 1988.

[Johansen 1991]
Johansen, Robert: Leading Business Teams, How Teams Can Use Technology and Group Process Tools to Enhance Performance, Addison-Wesley, New York, 1991.

[Just 1996]
Just, Katharina: Step-by-Step, A concept for describing co-operation within workflow-management-systems, in: SIGOIS Bulletin, Vol. 17, No. 1, April, 1996.

[Kamath et al. 1997]
Kamath, Mohan; Ramamritham, Krithi; Gehani, Narain; Lieuwen, Daniel: WorldFlow : A System For Building Global Transactional Workflows, Proc. of 7th International Workshop on High Performance Transaction Systems, Asilomar, California, 1997.

[Kaminske/Fürmann 1995]
Kaminske, Gerd F.; Fürmann, Timo: Reengineering versus Prozeßmanagement, Der richtige Weg zur prozeßorientierten Organisationsgestaltung, in: zfo, 3, 1995, pp. 142-148.

[Karbe 1994]
Karbe, Bernhard H.: Flexible Vorgangsteuerung mit ProMInanD, in: U. Hasenkamp, K. Kirn, M. Syring (eds.): CSCW Computer Supported Cooperative Work: Informationssysteme für dezentralisierte Unternehmensstrukturen, Addison-Wesley, 2. Auflage, 1994, pp. 117-133.

[Karbe/Ramsperger 1991]
Karbe, Bernhard H.; Ramsperger, Norbert G.: Concepts and Implementation of Migrating Office Processes, in: Brauer, W., Hernandez, D. (eds.): Proceedings of Verteilte Künstliche Intelligenz und kooperatives Arbeiten, Gesellschaft für Informatik, Informatik Fachberichte 291, Springer Verlag, October, 1991.

[Karbe/Ramsperger/Weiss 1990]
Karbe, Bernhard H.; Ramsperger, Norbert G.; Weiss, R.: Support of Cooperative Work by Electronic Circulation Folders, Conference on Office Information Systems, ACM Special Interest Group on Office Information Systems, IEEE Computer Society, Technical Committee on Office Automation, Los Alamitos, CA, USA, Vol. 11, Issues 2/3, 1990, pp. 109-117.

[Katzenbach/Smith 1993]
Katzenbach, Jon R.; Smith, Douglas K.: Teams: der Schlüssel zur Hochleistungsorganisation, übers. aus dem Amerik. "The Wisdom of Teams", Wirtschaftsverlag Ueberreuter, Wien, 1993.

[Katz/Kahn 1978]
Katz, D.; Kahn, R.L.: The social psychology of organization, 2nd ed., New York, London, 1978.

[Kawell et al. 1992]
Kawell, Leonard; Beckhardt, Steven; Halvorsen, Timothy; Ozzie, Raymond; Greif, Irene: Replicated Document Management in a group communication system, in: Marca, Dave, Bock, Geoffrey (eds.): Groupware: Software for computer supported cooperative work, IEEE Computer Society Press, 1992, pp. 226-235.

[Kieser 1993]
Kieser, Alfred: Der Situative Ansatz, in: Kieser, Alfred (ed.): Organisationstheorien, Kohlhammer, Berlin Köln, 1993, pp. 161-191.

[Kieser 1996]
Kieser, Alfred: Business Prozess Reengineering - neue Kleider für den Kaiser, in: zfo, 3, 1996, pp. 179-185.

[Kille 1996]
Kille, Steve: LDAP and X.500, in: Messaging Magazine, September, 1996.

[Kilmann 1989]
Kilmann, Robert E.: A completely integrated program for organizational changen, in: Mohrman, A. M.; et al.: Large-Scale Organizational Change, Jossey-Bass, San Francisco,CA, 1989, pp. 200-228.

[Kirn 1995]
Kirn, Stefan: Organisatorische Flexibilität durch Workflow-Management-Systeme?, in: Becker, J. u.a. (eds.): Arbeitsberichte des Institutes für Wirtschaftsinformatik Nr. 38, 1995, p. 10.

[Kirn/Unland 1994]
Kirn, Stefan; Unland, Rainer: Workflow Management mit kooperativen Softwaresystemen: State of the Art und Problemabriß, Arbeitsbericht Nr. 29, Institut für Wirtschaftsinformatik der Westfälischen Wilhelms-Universität Münster, 1994.

[Kirsch/zu Knyphausen 1991]
Kirsch, W; zu Knyphausen, D.: Unternehmen als "autopoietische" Systeme?, in: Staehle, W.H.; Sydow, J. (eds.): Managementforschung 1, Berlin, New York, 1991, pp. 75-101.

[Klein 1994]
Klein, Stefan: Virtuelle Organisation, Informations- und kommunikationstechnische Infrastrukturen ermöglichen neue Formen der Zusammenarbeit, in: WiSt, 23, 6, 1994, pp. 309-311.

[Kling 1980]
Kling, R.: Social analysis of computing: Theoretical perspectives in recent empirical research, in: ACM Computing Survey, March, 12, 1, 1980, pp. 61-110.

[Klotz 1993]
Klotz, Michael: Integrierte Anwendungssoftware und Unternehmensorganisation, Ein neues aufbauorganisatorisches Konzept zum Abbau von Hierarchieebenen, in: Krallmann, Hermann (ed.): Betriebliche Informations- und Kommunikationssysteme, Band 17, Erich Schmidt Verlag, Berlin, 1993.

[Klotz 1995]
Klotz, U.: Business Reengineering, Networking und Groupware - Schlüssel zur Organiation der Zukunft (II), in: Office Management, 3, 1995, pp. 12-19.

[Knolmayer/Herbst 1993]
Knolmayer, Gerhard; Herbst, Holger: Business Rules, in: Wirtschaftsinformatik, 35, 4, 1993, pp. 386-390.

[Kozar/Zigurs 1992]
Kozar, Kenneth A.; Zigurs, Ilze: Human and machine roles in team product reviews, A prescription for change, in: Information & Management, North-Holland, 23, 1992, pp. 149ff.

[Kozlowski 1997]
Kozlowski, Mary Ann: Delphi Presents Wide Area Workflow Seminar at BPR Conference, Press Release, Delphi Consulting Group, http://www.delphigroup.com/ pressreleases/1997-PR/02131997DelphiGigaSeminar.html, 1997.

[Krackhardt 1994]
Krackhardt, David: Graph Theoretical Dimensions of Informal Organizations, in: Carley, K.M.; Prietula, M.J. (eds.): Computational Organization Theory, Lawrence Erlbaum, Hillsdale, New Jersey, 1994, pp. 89-103.

[Krackhardt/Stern 1988]
Krackhardt, David; Stern, R. N.: Informal networks and organizational crises, An experimental simulation, Social Psychology Quarterly, 51(2), 1988, pp. 123-140.

[Krallmann/Boeckhoff/v. Bogdandy 1996]
Krallmann, Herrmann; Boeckhoff, Hanno; v. Bogdandy, Christian: Multiagentensysteme für die technologische Unterstützung der lernenden Organisation, in: Bullinger, H.-J. (ed.): Lernende Organisationen - Konzepte, Methoden und Erfahrungsberichte, Schäffer-Poeschel, Stuttgart, 1996.

[Krallmann/Derszteler 1996]
Krallmann, Herrmann; Derszteler, Gérard: Workflow Management Cycle, An Integrated Approach to the Modelling, Execution, and Monitoring of Workflow-Based Processes, in: Scholz-Reiter, B.; Stickel, E. (eds.): Business Process Modelling, Springer Verlag, Berlin u.a., 1996, pp. 23-42.

[Krallmann/Klotz 1994]
Krallmann, Herrmann; Klotz, Michael: Grafisches Organisationswerkzeug zur Unternehmensmodellierung, in: Office Management, 34-36, 5, 1994.

[Krallmann et al. 1989]
Krallmann, Herrmann; Feiten, L.; Hoyer, R.; Kölzer, G.: Die Kommunikationsstrukturanalyse (KSA), Zur Konzeption einer betrieblichen Kommunikationsarchitektur, in: Kurbel, K.; Mertens, P.; Scheer, A.W. (Eds.): Interaktive betriebswirtschaftliche Informations- und Kommunikationssysteme, Walter de Gruyter, Berlin, 1989.

[Kreifelts et al. 1991]
Kreifelts, Thomas; Hinrichs, E.; Klein, K.-H.; Seuffert, P.; Woetzel, G.: Experiences with the DOMINO office procedure system, in: Bannon, L., Robinson, M., Schmidt, K. (eds.): ECSCW '91, Proceedings of the 2nd European Conference on Computer Supported Cooperative Work, Kluwer, Dordrecht, 1991, pp. 117-130.

[Kreifelts/Hinrichs/Woetzel 1993]
Kreifelts, Thomas; Hinrichs, E.; Woetzel, G.: Sharing To-Do Lists with a Distributed Task Manager, DeMichelis, G.; Simone, C.; Schmidt, K. (eds.): Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW'93, Kluwer Academic Publishers, Dordrecht u.a., 1993, pp. 31-46.

[Kreifelts/Woetzel 1987]
Kreifelts, Thomas; Woetzel, G.: Distribution and Error Handling in an Office Procedure System, in: Tsichritzis, D.; Bracchi, G. (eds.): Office Systems: Methods and Tools, IFIP WG.8.4 Conference on Methods and Tools for Office Systems, Elsevier, Amsterdam, Netherlands, 1987, pp. 197-208.

[Krickl 1995]
Krickl, Otto Ch.: Business Redesign, Neugestaltung von Organisationsstrukturen unter besonderer Berücksichtigung der Gestaltpotentiale von Workflowmanagementsystemen, FBO-Verlag, Wiesbaden, 1995.

[Kronert 1988]
Kronert, G.: Genormte Austauschformate für Dokumente, in: Informatik-Spektrum, Vol. 11, No. 2, 1988, pp. 71-84.

[Krulwich 1997]
Krulwich, Bruce T.: The Good Ol' Bot Network, http://www.ac.com/cstar/hsil/agents/framedef_art_cf.html, 1997.

[Kürn 1997]
Kürn, Hans-Christoph: Intranet als Stellenmarkt, in: Personalwirtschaft, 3, 1997, pp. 32f.

[Kuutti 1993]
Kuutti, Kari: Notes on Systems Supporting "Organizational Context", An Activity Theory Viewpoint, in: Bannon, L.; Schmidt, K. (eds.): Issues of Supporting Organizational Context in CSCW Systems, COMIC Deliverable Report 1.1, 1993, pp. 101-118.

[Lawrence 1997]
Lawrence, Peter: Workflow Handbook, John Wiley & Sons Ltd, Chichester, England, 1997.

[Lehner 1991]
Lehner, Franz: Expertensysteme für Organisationsaufgaben, in: ZfB, 61.Jg., 7, 1991, pp. 23-40.

[Lehner 1997]
Lehner, Franz: Organisationsgestaltung im Spannungsfed computergestützter Werkzeuge, in: Rosemann, M., Becker, J. (eds.): Organisatorische und technische Aspekte beim Einsatz von Workflowmanagementsystemen, Arbeitsbericht Nr. 54, Universität Münster, Institut für Wirtschaftsinformatik, 10. April, 1997, pp. 1-23.

[Lehner et al. 1991]
Lehner, Franz; Auer-Rizzi, W.; Bauer, R.; Breit, K.; Lehner, J.; Reber, G.: Organisationslehre für Wirtschaftsinformatiker, München, 1991.

[Lewe/Krcmar 1991]
Lewe, Henrik; Krcmar, Helmut: Groupware, in: Informatik Spektrum, 14, 6, 1991, pp. 345-348.

[Leymann 1997]
Leymann, Frank: Transaktionsunterstützung für Workflow, in: Informatik, Forschung und Entwicklung, Springer, 12, 1997, pp. 82-90.

[Liebrand 1995]
Liebrand, Mark: Information Object Modeler, CASE tool for Lotus Notes Database Design, Masters Thesis, University of Paderborn, Department of Business Computing 2, 1995.

[Lilienthal 1998]
Lilienthal, Antje: Aus eigener Kraft, Neustrukturierung, in: Forum - Magazin für Mitarbeiter der Deutschen Bank, Frankfurt/Eschborn, März, 1998, pp. 14f.

[Lilienthal/Bonk 1998]
Lilienthal, Antje; Bonk, Thomas: Chancen am Scheideweg, Neustrukturierung, in: Forum - Magazin für Mitarbeiter der Deutschen Bank, Frankfurt/Eschborn, April, 1998, pp. 14f.

[Lin 1994]
Lin, Zhiang: A Theoretical Evaluation of Measures of Organizational Design, Interrelationship and Performance Predictability, in: Carley, K.M.; Prietula, M.J. (eds.): Computational Organization Theory, Lawrence Erlbaum, Hillsdale, New Jersey, 1994, pp. 113-159.

[Lindo 1996]
Lindo, W.: Prozeßmodelle als Basis für IT-Projekte, in: PC Magazin, 46, 1996, pp. 46-47.

[Lippold/Hilgenfeldt/v. Kortzfleisch 1993]
Lippold, H.; Hilgenfeldt, J.; v. Kortzfleisch, Harald F. O.: Werkzeuge für die computergestützte Organisationsgestaltung, BIFOA-Marktstudie, BIFOA-Selbstverlag, Köln, 1993.

[Li/Lochovsky 1996]
Li, Qing; Lochovsky, Frederick H.: Advanced Database Support Facilities for CSCW Systems, in: Journal of Organizational Computing and Electronic Commerce, Vol.6, No.2, 1996, pp. 191-210.

[Lochovsky et al. 1988]
Lochovsky, Frederick H.; Hogg, J. S.; Weiser, S. P.; Mendelzon, A. O.: OTM: Specifying Office Tasks, in: Proceedings of the Conference on Office Information Systems (Palo Alto, CA, March 23-25) , ACM, New York, 1988, pp. 46-53.

[Lohse et al. 1994]
Lohse, Gerald L.; Biolsi, Kevin; Walker, Neff; Rueter, Henry H.: A Classification of Visual Representations, in: Communications of the ACM, unbekannt, December, Vol.37, No.12, 1994, pp. 36-49.

[Lotus Dev. 1994]
Lotus Dev.: Security Requirements for Groupware Applications, A Lotus Development Corporation Strategy Brief, Cambridge, MA, May, 1994.

[Lotus Dev. 1995]
Lotus Dev.: Groupware - Communication, Collaboration, Coordination, November, 1995.

[Lotus Dev. 1996]
Lotus Dev.: Lotus Notes® - A System for Managing Organizational Knowledge, A Technical White Paper, September, 1996.

[Lucas/Baroudi 1994]
Lucas, Henry C.; Baroudi, J.: The Role of Information Technology in Organization Design, in: Journal of Management Information Systems (JMIS), Armonk, N. Y., Vol. 10 Nr. 4, 1994, pp. 9-23.

[Lutze/Kohl 1991]
Lutze, R.; Kohl, A. (eds.): Wissensbasierte Systeme im Büro, Ergebnisse aus dem WISDOM-Verbundprojekt, Oldenbourg, München, Wien, 1991.

[Malik 1993]
Malik, Fredmund: Systematisches Management, Evolution, Selbstorganisation: Grundprobleme, Funktionsmechanismen und Lösungsansätze für komplexe Systeme, Haupt, Bern, 1993, pp. 1-124.

[Malone/Crowston 1990]
Malone, Thomas W.; Crowston, Kevin: What is coordination theory and how can it help design cooperative systems?, in: Proceedings of CSCW '90, ACM, New York, 1990, pp. 357-370.

[Malone/Crowston 1994]
Malone, Thomas W.; Crowston, Kevin: The interdisciplinary study of coordination, in: ACM Computing Surveys, 26, 1994, pp. 185-201.

[Malone/Fry/Lai 1992]
Malone, Thomas W.; Fry, Ch.; Lai, K.-Y.: Experiments with Oval, A Radically Tailorable Tool for Cooperative Work, in: Proceedings of CSCW '92, ACM, Los Angeles, USA, 1992, pp. 289-297.

[Malone/Rockart 1993]
Malone, Thomas W.; Rockart, John F.: How will Information Technology Reshape Organizations?, Computers as Coordination Technology, in: Bradley et al. (eds.): Globalization, Technology and Competition, The Fusion of Computer and Telecommunications in the 1990s, Harvard Business School Press, Boston, Massachusettes, 1993, pp. 36-56.

[Markus/Connolly 1990]
Markus, K.; Connolly, T.: Why CSCW Applications Fail: Problems in the Adoption of Interdependent Work Tools, Problems in the Adoption of interdependent work tools, in: Proceedings of CSCW '90, ACM, Los Angeles, USA, 1990, pp. 371-380.

[Marshak 1995]
Marshak, Ronni T.: Pavone GroupFlow, Providing a Workflow Suite for Lotus Notes Environments, in: Workgroup Computing Report, Vol. 18, Nr. 11, Patricia Seybold Group, Boston, November, 1995, pp. 3-23.

[Martin 1996]
Martin, James: Enterprise Networking, Prentice Hall, Englewood Cliffs, New Jersey, 1996.

[Matysczok in prep.]
Matysczok, Carsten: The Consolidated GroupOrga Enterprise Information Management Model (GEIMM), Masters Thesis, University of Paderborn, Department of Business Computing 2, in preparation.

[McCarthy/Sarin 1993]
McCarthy, D.; Sarin, Sunil K.: Workflow and Transaction in InConcert, in: IEEE Data Engineering, Vol. 16, No. 2, 1993.

[McCready 1992]
McCready, Scott C.: Workflow Software: Chariots of Fire, in: FIBA Conferences: Office Automation '92, 1992.

[Medina-Mora et al. 1992]
Medina-Mora, Raul; Winograd, Terry; Flores, Rodrigo; Flores, Fernando: The Action Workflow Approach to Workflow Management Technology, in: Proceedings of CSCW ´92: Sharing Perspectives, The Fourth International Conference on Computer Supported Cooperative Work, ACM, Alameda, CA, 1992, pp. 281-288.

[Mertens/Faisst 1995]
Mertens, Peter; Faisst, Wolfgang: Virtuelle Unternehmen - eine Organisationsstruktur für die Zukunft?, in: technologie & management, vol 44, no. 2, 1995, pp. 61-68.

[Meyer 1995]
Meyer, Stefan: Konzeption und Implementierung eines Werkzeugs zum Entwurf und zur Analyse von Bürovorgängen mit offenen objekt-orientierten Strukturen (Workflow Object Modeler), Diplomarbeit, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, 1995.

[Meyer 1996]
Meyer, Stefan: Neue Konzepte zur computergestützten Organisationsmodellierung, Untersuchung und Konzeption eines computergestützten Hilfsmittels zur Organisationsmodellierung zum Einsatz in innovativen Workflow Management Systemen (Organization Object Modeler), Diplomarbeit, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, August, 1996.

[Meyersiek 1995]
Meyersiek, Dietmar: Organisatorisches Monitoring als Instrument zur Produktivitätsverbesserung, in: Wirtschaftsinformatik, 37/1, 1995, pp. 73-74.

[Mintzberg 1979]
Mintzberg, Henry: The structuring of organizations, Prentice-Hall, Englewood Cliffs, NJ, 1979.

[Moad 1994]
Moad, Jeff: Welcome To The Virtual IS Organization, in: Datamation, February 1, 1994, pp. 32-35.

[Morabito 1995]
Morabito, Joseph: Organization Modeling: an object-oriented approach, Dissertation, Stevens Institute of Technology, 1995.

[Morgan 1986]
Morgan, G.: Images of organization, Beverly Hills, 1986.

[Morschheuser/Raufer/Wargitsch 1996]
Morschheuser, Stefan; Raufer, H.; Wargitsch, Christoph: Challenges and Solutions of Document and Workflow Management in a Manufacturing Enterprise: A Case Study, in: Nunamaker jr., J. F. et al. (eds.): Proceedings of the 29th Annual Hawaii International Conference on System Science, IEEE Computer Society Press, Los Alamitos, Vol. V, 1996, pp. 4-13.

[Müller 1997]
Müller, Carsten: Gestaltung eines lernenden Organisationsinformationssystems (OIS), Untersuchung über die Entwicklung der Organisationsforschung und Prototypische Entwicklung eines Organisationsinformationssystems (Dynamische Organisationsdatenbank), Masters Thesis, University of Paderborn, Department of Business Computing 2, 1997.

[Mummert 1996]
Mummert, Mummert + Partner Unternehmensberatungs GmbH: Marktübersicht Vorgangssteuerungssysteme, Frankfurt, 1996.

[Mummert 1997]
Mummert, Mummert + Partner Unternehmensberatungs GmbH: Marktübersicht Vorgangssteuerungssysteme, Hamburg, 1997.

[Nadler 1994]
Nadler, David A.: Organisations-Architektur, Eine Metapher für den Wandel, in: Nadler, David A.; Gerstein, M. S.; Shaw, R. B. (eds.): Organisations-Architektur, Optimale Strukturen für Unternehmen im Wandel, Campus Verlag, Frankfurt, New York, 1994., Campus Verlag, Frankfurt, New York, 1994, pp. 12-20.

[Nadler/Gerstein/Shaw 1992]
Nadler, David A.; Gerstein, Marc S.; Shaw, Robert B.: Organizational Architecture, Designs for Changing Organizations, San Francisco, Jossey-Bass Publishers, 1992.

[Nadler/Tushman 1994]
Nadler, David A.; Tushman, Michael L.: Das Design optimaler Organisationen, Ein Leitfaden für das Verständnis neuer Archtekturen, in: Nadler, David A.; Gerstein, M. S.; Shaw, R. B. (eds.): Organisations-Architektur, Optimale Strukturen für Unternehmen im Wandel, Campus Verlag, Frankfurt, New York, 1994., Campus Verlag, Frankfurt, New York, 1994, pp. 49-68.

[Nastansky 1992]
Nastansky, Ludwig: Multimedia und Imageprocessing im Spannungsfeld neuer Technologiemärkte und innerbetrieblicher Produktivitätszwänge, Tagungsband X/92, AIT, Halbergmoos, 1992, pp. 1-36.

[Nastansky 1993a]
Nastansky, Ludwig: Nach 20 Jahren CSCW-Forschung: Durchbruch in der Praxis bei Groupware-Anwendungen in Client-Server Architekturen, in: Nastansky, Ludwig (ed.): Workgroup Computing, S+W Steuer und Wirtschaftsverlag, Hamburg, 1993, pp. 1-20.

[Nastansky 1993b]
Nastansky, Ludwig: Workgroup Computing, Computergestützte Teamarbeit (CSCW) in der Praxis; neue Entwicklungen und Trends, Steuer- und Wirtschaftsverlag, Hamburg, Band 12, 1993.

[Nastansky/Hilpert 1994]
Nastansky, Ludwig; Hilpert, Wolfgang: The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation, in: Wolfinger, Bernd (ed.): Innovationen bei Rechen- und Kommunikationssystemen - Eine Herausforderung an die Informatik, Proceedings of 24 th Anual Conference of the German Computer Society during 13 th World Computer Congress, IFIP '94, Springer Verlag, Berlin, Heidelberg etc., 1994, pp. 473-479.

[Nastansky/Hilpert 1995]
Nastansky, Ludwig; Hilpert, Wolfgang: Das GroupFlow System für Workflow-Management, Balance zwischen Struktur und Flexibilität, in: Business Computing, Vogel-Verlag , Würzburg, Nr. 7, Juli, 1995, pp. 30-31.

[Nastansky/Hilpert 1996]
Nastansky, Ludwig; Hilpert, Wolfgang: The GroupFlow Framework: Enterprise Model and Architecture of the Workflow System, Shortened version published in Koenig, W., Kurbel, K., Mertens, P., Pressmar, D. (eds.): Distributed Information Systems in Business, Springer, Berlin etc., 1996, pp. 201-212.

[Nastansky et al. 1995]
Nastansky, Ludwig; Hilpert, Wolfgang; Ott, Marcus; Riempp, Gerold: Geschäftsprozeßorientierte Modellierung, Einführung von Groupware Anwendungen, in: Groupware Forum, Schöll Trainingszentrum Darmstadt, Darmstadt, Oktober, 1995.

[Nastansky/Ott 1996]
Nastansky, Ludwig; Ott, Marcus: Office Management im Team zwischen Struktur und Flexibilität (GroupOffice), GroupOffice-Infrastruktur für das Informationsmanagement, in: Uellner, Stefan (ed.): Computer Supported Coorperative Work (CSCW) in großen Unternehmungen, Tagungsband zum Workshop der Gesellschaft für Informatik (Fachgruppe 5.5.1) und dem Technologiezentrum der Deutschen Telekom AG, Darmstadt, Mai, 1996, pp. 39-53.

[Nastansky/Otten/Drira 1994]
Nastansky, Ludwig; Otten, Angelika; Drira, Mohamed: Bedeutung des Replikationskonzepts, Mobile Office, Replikation in verteilten heterogenen Datenbankumgebungen, Replikation und Informationsabgleich in Workgroups, Arbeitspapier, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, 1994.

[Nippa 1995]
Nippa, Michael: Bestandsaufnahme des Reengineering-Konzepts, Leitgedanken für das Management, in: Nippa, Michael; Picot, Arnold (eds.): Prozeßmanagement und Reengineering, Die Praxis im deutschsprachigen Raum, 1995, pp. 61-70.

[Nolte 1997]
Nolte, Dirk: Verteilte Organisationsmodellierung, Seminarbeit, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Juni, 1997.

[Nolte 1998]
Nolte, Petra: Einsetzbarkeit von Groupwaretechnologien im Organisationsdesign, Untersuchung zur Akzeptanz computergestützer Arbeitshilfen zur Organisationsmodellierung im Rahmen von Tele- und Teamarbeit, Working Paper, University of Paderborn, Department of Business Computing, Mai, 1998.

[Nunamker/George/Valacich 1989]
Nunamker, J.F.; George, J.F.; Valacich, J.S.: ODSS: Information Technology for Organizational Change, in: CMI WPS 89-12, university of Arizone, 1989.

[Oberschulte 1996]
Oberschulte, Hans: Organisatorische Intelligenz - ein Vorschlag zur Konzeptdifferenzierung, in: Schreyögg, G., Conrad, P. (eds.): Managementforschung, de Gruyter, Berlin, New York, 6, 1996, pp. 41-82.

[Orlikowski 1995]
Orlikowski, Wanda J.: Evolving with Notes: Organizational Change around Groupware Technology, White Paper, MIT, Center for Coordination Science, Boston, June, 1995.

[Orlikowski/Hofman 1996]
Orlikowski, Wanda J.; Hofman, Debra J.: An Improvisational Model of Change Management: The Case of Groupware Technologies, in: Center for Coordination Science, MIT, Working paper No. 191, Cambridge, MA, 1996.

[Ortner 1991a]
Ortner, Erich: Ein Referenzmodell für den Einsatz von Directory/Repository Systemen in Unternehmen, in: Wirtschaftsinformatik, 33, 5, 1991, pp. 420-430.

[Ortner 1991b]
Ortner, Erich: Unternehmensweite Datenmodellierung als Basis für integrierte Informationsverarbeitung in Wirtschaft und Verwaltung, in: Wirtschaftsinformatik, 33. Jg., 4, August, 1991, pp. 269-280.

[Österle 1993]
Österle, Hubert: Ein Modell für den Prozeßentwurf, 1993.

[Österle 1995]
Österle, Hubert: Business Engineering, Prozeß- und Systementwicklung, Band 1: Entwurfstechniken, Springer, Berlin, Heidelberg, New York, Tokyo, 1995.

[Österle/Saxer/Hüttenhain 1994]
Österle, Hubert; Saxer, R.; Hüttenhain, T.: Organisatorisches Monitoring in der Gestaltung von Geschäftsprozessen, in: Wirtschaftsinformatik, 36 (1994) 5, 1994, pp. 465-477.

[Ott 1994]
Ott, Marcus: Conceptual Design and Implementation of a Graphical Workflow-Modelling Editor in the Context of Distributed Groupware-Databases, Masters Thesis, University of Paderborn - Faculty of Business Studies (FB 5), Department for Information Management (Wirtschaftsinformatik 2), Paderborn, May, 1994.

[Ott 1995]
Ott, Marcus: The Enterprise Model Underlying GroupFlow, Working Report, Universität Paderborn, Lehr- und Forschungseinheit Wirtschaftsinformatik 2, Unpublished, 1995.

[Ott 1997a]
Ott, Marcus: Groupware, Charakterisierung und technologische Perspektive, in: WiSt - Wirtschaftswissenschaftliches Studium, C.H. Beck, Vahlen, München, Frankfurt a.M., Februar, 1997, pp. 90-94.

[Ott 1997b]
Ott, Marcus: Groupware als Quelle für Internet/WWW Präsenz, Möglichkeiten und Potentiale der Kombination innovativer Technologien zum Informationsmanagement, in: WiSt - Wirtschaftswissenschaftliches Studium, C.H. Beck, Vahlen, München, Frankfurt a.M., Mai, 1997, pp. 253-260.

[Ott/Huth 1998a]
Ott, Marcus; Huth, Carsten: Einsatz von Java Applikationen für das Organisationsdesign virtueller Unternehmen, in: Sommer, M., Remmele, W., Klöckner, K. (eds.): Interaktion im Web - Innovative Kommunikationsformen, Fachtagung und Kongreß, Berichte des German Chapter of the ACM und der Gesellschaft für Informatik, Fachgruppe 2.0.1 - Personal Computing, Fachgruppe 3.1.3 - Arbeitsplatzrechensysteme, Band 50, B.G. Teubner, Stuttgart, Mai, 1998.

[Ott/Huth 1998b]
Ott, Marcus; Huth, Carsten: Collaborative Organization Design: A Synergy of Groupware and Web-based Infrastructures and Technology, in: WET ICE '98, Web-based Infrastructures for Collaborative Enterprises: Ready for a Second Generation?, Stanford University, June, 1998.

[Ott/Huth/Nastansky 1998]
Ott, Marcus; Huth, Carsten; Nastansky, Ludwig: Reengineering Organizational Structures from Within, submitted to the 32nd Hawai'i International Conference On System Sciences, Workflow and Reengineering on the Internet, Maui, Hawaii, 1998.

[Ott/Nastansky 1997a]
Ott, Marcus; Nastansky, Ludwig: New perspectives for higher education processes as a team-based approach, Back-office information technology for higher education and training, in: Krallmann, Herrmann (ed.): Wirtschaftsinformatik '97, Internationale Geschäftstätigkeit auf Basis flexibler Organisationsstrukturen und leistungsfähiger Informationssysteme, Physica-Verlag, Heidelberg, February, 1997, pp. 241-261.

[Ott/Nastansky 1997b]
Ott, Marcus; Nastansky, Ludwig: Modelling Organizational Forms of Virtual Enterprises (VoNet), The Use of CSCW Environments for a Team Based, Distributed Design of Virtual Organizations, in: Griese, J.; Sieber, P. (eds.): VoNet, The Newsletter @ http://www.virtual-organization.net, Institute of Information Systems Department of Information Management University of Berne, Vol. 1, No. 4, September, 1997, pp. 20-39.

[Ott/Nastansky 1997c]
Ott, Marcus; Nastansky, Ludwig: Kommunikationsmanagement zwischen Teams, Paperless Office am Beispiel GroupOffice, in: Dangelmaier, W., Fischer, J., Herold, W., Nastansky, L., Suhl, L., Wolff, R. (eds.): Kommunikationsmanagement in verteilten Unternehmen - Fünf Jahre Wirtschaftsinformatik an der Universität-GH Paderborn, VDI Verlag, Düsseldorf, Reihe 10, Nr. 478, 1997, pp. 21-44.

[Ott/Nastansky 1997d]
Ott, Marcus; Nastansky, Ludwig: Verteilung und Kooperation bei der Modellierung von Workflow Management, in: Lehner, F., Dustdar, S. (Hrsg.): Telekooperation in Organisationen, Deutscher Universitäts Verlag, Gabler Edition Wissenschaft, Wiesbaden, 1997, pp. 79-98.

[Ott/Nastansky 1998a]
Ott, Marcus; Nastansky, Ludwig: Groupware Technology for a new Approach to Organization Design Systems, in: Watson, Hugh J.; Nunamaker jr., F.; Sprague, R.H (eds.): Proceedings of the 31st Hawai'i International Conference On System Sciences, Vol. VI, Organizational Systems and Technology, Kona, Hawaii, IEEE Computer Society Press, Los Alamitos CA, Washington, Brussels, Tokio, January, 1998, pp. 562-571.

[Ott/Nastansky 1998b]
Ott, Marcus; Nastansky, Ludwig: Distributed Global Enterprise Directories Implemented as Organizational Subsystems to WfMS, in: Abramowicz, W. (ed.): Proceedings Business Information Systems ’98 (BIS'98), Poznañ, Poland, April, 1998, pp. 75-91.

[Picot/Franck 1995]
Picot, Arnold; Franck, Egon: Prozeßorganisation, Eine Bewertung der neuen Ansätze aus Sicht der Organisationslehre, in: Nippa, Michael; Picot, Arnold (eds.): Prozeßmanagement und Reengineering, Die Praxis im deutschsprachigen Raum, 1995, pp. 13-38.

[Picot/Maier 1993]
Picot, Arnold; Maier, Matthias: Interdependenzen zwischen betriebswirtschaftlichen Organisationsmodellen und Informationsmodellen, in: Information Management (IM), Nr. 3, 1993, pp. 6-15.

[Picot/Rohrbach 1995]
Picot, Arnold; Rohrbach, Peter: Organisatorische Aspekte von Workflow-Management-Systemen, in: Information Management, No. 1, 1995, pp. 28-35.

[Prinz 1989a]
Prinz, Wolfgang: Application of the X.500 directory by office systems: support for representation, authentication, autorization, and MHS, Oldenbourg, München, Wien, 1989.

[Prinz 1989b]
Prinz, Wolfgang: Survey of group communication models and systems, in: Pankoke-Babatz, U. (ed.): Computer Based Group Communication - the AMIGO Activity Model, Ellis Horwood, Chicester, 1989, pp. 128-180.

[Prinz 1993]
Prinz, Wolfgang: TOSCA: Providing Organisational Information to CSCW applications, in: Michelis, G.d., Schmidt, K., Simone, C. (eds.): Proceedings of the Third European Conference on Computer Supported Cooperative Work - ECSCW'93, Kluwer, Dordrecht, 1993, pp. 139-154.

[Prinz 1996]
Prinz, Wolfgang: A Framework for Organizational Information Systems in Cooperative Environments, Dissertation, University of Nottingham, October, 1996.

[Prinz/Kolvenbach 1996]
Prinz, Wolfgang; Kolvenbach, Sabine: Support for Workflows in a Ministerial Environment, in: Proceedings of CSCW'96, ACM, Boston, MA, 1996, pp. 199-208.

[Rathgeb 1994]
Rathgeb, Michael: Einführung von Workflow-Management-Systemen, in: Hasenkamp, U.; Kirn, S.; Syring, M. (eds.): CSCW - Computer Supported Cooperative Work, Informationssysteme für dezentralisierte Unternehmensstrukturen, Addison-Wesley, Bonn etc., 1994, pp. 45 - 66.

[Rathgeb 1996]
Rathgeb, Michael: Verfahren zur Gestaltung rechnergestützter Büroprozesse, Springer Verlag, Berlin etc., 1996.

[REFA 1984]
REFA, Verband für Arbeitsorganisation und Betriebsorganisation e.V.: Methodenlehre des Arbeitsstudiums, Teil 1 Grundlagen, Carl Hanser, 7, München, 1984.

[Reichwald 1984]
Reichwald, Ralf: Coorperation in the office - Office communication systems as management tool, in: Prooceedings of the 2nd ACM SIGOA Conference on Office Information Systems, ACM, New York, June, 1984, pp. 212-218.

[Reichwald/Picot 1987]
Reichwald, Ralf; Picot, Arnold: Gestaltungsgrundsätze in der Büroorganisation, Karl Schmidt Verlag, 1987.

[Reim 1992]
Reim, Friedemann: Entwicklung eines Verfahrens zur rechnergestützten Gestaltung verteilter Informatioinssysteme, Springer, Berlin, Heidelberg, New York etc., 1992.

[Rein 1992]
Rein, Gail Louise: Organization Design Viewed as a Group Process Using Coordination Technology, MCC Technical Report Number CT-039-92, Dissertation, University of Texas, Austin, February, 1992.

[Richards 1997]
Richards, Cate: Using Lotus Notes 4.5, QUE, Indianapolis, 1997.

[Riempp/Nastansky 1996]
Riempp, Gerold; Nastansky, Ludwig: Workflow Management between distributed organizations - the Wide Area GroupFlow Approach, in: Krcmar, H. et al. (eds.): Herausforderung Telekooperation - Proceedings of the German Computer Supported Cooperative Work 1996, Springer Verlag, Berlin, Heidelberg etc., 1996, pp. 275-292.

[Riempp/Nastansky 1997]
Riempp, Gerold; Nastansky, Ludwig: Conceptions and solutions for the interaction of distributed workflow management systems, in: Abramowicz, W. (ed.): Proceedings Business Information Systems ’97 (BIS'97), Poznañ, Polen, April, 1997, pp. 583-599.

[Rogers 1994]
Rogers, Yvonne: Exploring obstacles: Integrating CSCW in evolving organizations, in: Proceedings of the CSCW'94 Conference, ACM-Press, New York, 1994, pp. 67-78.

[Rohloff 1996]
Rohloff, Michael: An object oriented approach to business process modelling, in: Scholz-Reiter, B.; Stickel, E. (Eds.): Business Process Modelling, Springer Verlag, Berlin u.a., 1996, pp. 251-264.

[Roithmayr 1996]
Roithmayr, Friedrich: Ansätze zu einer Methodik des Know-how-Engineering, in: Heilmann, Heidi; Heinrich, Lutz J.; Roithmayr, Friedrich (eds.): Information Engineering, Wirtschaftsinformatik im Schnittpunkt von Wirtschafts-, Sozial- und Ingenieurwissenschaften, Oldenbourg, München, 1996, pp. 103-122.

[Roos/Starke 1981]
Roos, Leslie L.; Starke, Frederick A.: Organizational roles, in: Nystrom, P. C.; Starbuck, W. H. (eds.): Handbook of Organizational Design, Vol. 1: Adapting organizations to their environments, Oxford University Press, Oxford, UK, 1981, pp. 290-308.

[Rosemann 1996]
Rosemann, Michael: Komplexitätmanagement in Prozeßmodellen, Methodenspezifische Gestaltungsempfehlungen für die Informationsmodellierung, Wiesbaden, 1996.

[RSA 1997]
RSA Data Security: RSA Company Brochure, July 18, 1997.

[Rupietta 1990]
Rupietta, Walter: An Organization & Resources Model for Adapting Office Systems to Organizational Structures, in: Tjoa, A.M., Wagner, R. (eds.) Database and Expert Systems Applications, Springer, Wien, New York, 1990, pp. 346-350.

[Rupietta 1992]
Rupietta, Walter: Organisationsmodellierung zur Unterstützung kooperativer Vorgangsbearbeitung, in: Wirtschaftsinformatik, Februar, 1992, pp. 26-37.

[Rupietta 1994]
Rupietta, Walter: Organization Models for Cooperative Office Applications, in: Karagiannis, Dimitris (ed.): Database and Expert Systems Applications, Proceedings of the 5th International Conference, DEXA'94, Athens, Greece, Sep. 7-9.1994, Springer Verlag, Berlin, Heidelberg, New York, 1994, pp. 114-124.

[Rupietta 1997]
Rupietta, Walter: Organization and Role Models for Workflow Processes, in: Lawrence, P. (ed.): Workflow Management Coalition: Workflow Handbook 1997, John Wiley & Sons, Chichester, 1997, pp. 165-172.

[Sarin/Abbott/McCarthy 1991]
Sarin, Sunil K.; Abbott, Kenneth R.; McCarthy, D.: A Process Model and System for Supporting Collaborative Work, in: de Jong, Peter (ed.): Proc. ACM-SIGOIS Conference on Organizational Computing Systems, ACM Press, Atlanta, Georgia, Nov. 5-8, 1991, pp. 213-224.

[Sauter/Morger 1996]
Sauter, C.; Morger, O.: Die Workflow Management Coalition, in: Wirtschaftsinformatik, Nr. 2, April, 1996, pp. 228-229.

[Sauter/Mühlherr/Teufel 1994]
Sauter, C.; Mühlherr, Th.; Teufel, Stefanie: Sozio-kulturelle Auswirkungen von Groupware - Ein Ansatz zur Adaption und Operationalisierung eines sozialpsychologischen Modells für die Gestaltung und den Einsatz von Groupware, in: Rauch W., Strohmeier F., Hiller H., Schlögel C. (eds.): Tagungsband des 4. Internationalen Symposiums für Informationswissenschaft, Universitätsverlag Konstanz, 1994, pp. 517-526.

[Scheer 1992]
Scheer, August-Wilhelm: Architektur integrierter Informationssysteme - Grundlagen der Unternehmensmodellierung, Springer Verlag, 2. Auflage, Berlin etc., 1992.

[Scheer 1995]
Scheer, August-Wilhelm: Wirtschaftsinformatik - Referenzmodelle für industrielle Geschäftsprozesse, Springer Verlag, 6., Berlin, Heidelberg, 1995.

[Scheer/Nüttgens/Zimmermann 1995]
Scheer, August-Wilhelm; Nüttgens, Markus; Zimmermann, Volker: Rahmenkonzept für ein integriertes Geschäftsprozeßmanagement, in: Wirtschaftsinformatik, 37/5, Oktober, 1995, pp. 426-434.

[Scheller 1974]
Scheller, P.: Systematische Untersuchungen bisheriger Anwendungen der Nutzwertanalyse zwecks Bestimmung der Möglichkeiten und Grenzen dieser Bewertungsmethode, Bericht NO 2/74, Technische Universität Berlin, 1974.

[Schmalenbach 1996]
Arbeitskreis "Organisation" der Schmalenbach-Gesellschaft e.V.: Organisation im Umbruch, (Was) Kann man aus den bisherigen Erfahrungen lernen, Deutsche Gesellschaft für Betriebswirtschaft e. V., zfbf, 6, 1996, pp. 621-665.

[Scholz-Reiter/Bastian 1995]
Scholz-Reiter, B.; Bastian, D.: Groupware Based Simultaneous Organizational and Manufacturing Modelling, in: Bubnicki, Z. (ed.): Proceedings of the 12th International Conference on Systems Science Vol. 2, Wroclaw, 1995, pp. 225 - 232.

[Schreyögg/Conrad 1996]
Schreyögg, G.; Conrad, P.: Managementforschung 6, Wissensmanagement, de Guyter, Berlin, New York, 1996.

[Schwarzer/Zerbe/Krcmar 1995]
Schwarzer, Bettina; Zerbe, Stefan; Krcmar, Helmut: Neue Organisationsformen als Untersuchungsgegenstand - Das NOF-Projekt, Arbeitspapier des Instituts für Wirtschaftsinformatik, Stuttgart-Hohenheim, 4, 1995.

[Seitz/Galster/Lang 1993]
Seitz, R.; Galster, C.; Lang, A.: Die Bedeutung organisationstheoretischer Grundlagen für die Entwicklung von Groupware-Komponenten, Bodendorf, Freimuth; Mertens, Peter (eds.): Arbeitspapier, Bereich Wirtschaftsinformatik, Universität Erlangen-Nürnberg, Nr. 4, 1993.

[Senge 1990]
Senge, Peter: The Fifth Discipline: The Art and Practice of the Learning Organization, Doubleday, New York, 1990.

[Sheng Liu/Wei Tai 1989]
Sheng Liu, Fang; Wei Tai, Ju: Some principles of icon construction, in: Kunii, Tosiyasu L. (ed.): Visual Database Systems, Proceedings of the IFIP TC 2/WG 2.6 Working Conference, Elsevier, North-Holland, Amsterdam, Tokyo, 3-7 April, 1989, pp. 89-104.

[Sheth/Rusinkiewicz 1993]
Sheth, Amit; Rusinkiewicz, M.: On transactional Workflow, in: IEEE Data Engineering, Vol. 16, No. 2, 1993.

[Sieber 1995]
Sieber, Pascal: Annotated Bibliography zum Thema Virtuelle Unternehmen, Arbeitsbericht Nr. 73, Arbeitspapier der Reihe "Informations und Kommunikationssysteme als Gestaltungselement Virtueller Unternehmen", Bern, Leipzig, Nürnberg, 4, 1995.

[Siemens Nixdorf 1997a]
Siemens Nixdorf, Informationssysteme AG: Facts and Figures 1996/97, Paderborn, 1997.

[Siemens Nixdorf 1997b]
Siemens Nixdorf, Informationssysteme AG: OIS, Organisations-Informationssystem, in: Informationsbroschüre, Draft, Paderborn, 1997.

[Sinz 1990]
Sinz, Elmar J.: Das Entity-Relationship-Modell (ERM) und seine Erweiterungen, in: HMD, 152, März, 1990, pp. 17-29.

[Sinz 1996]
Sinz, Elmar J.: Ansätze zur fachlichen Modellierung betrieblicher Informationssysteme, Entwicklung, aktueller Stand und Trends, in: Heilmann, Heidi; Heinrich, Lutz J.; Roithmayr, Friedrich (eds.): Information Engineering, Wirtschaftsinformatik im Schnittpunkt von Wirtschafts-, Sozial- und Ingenieurwissenschaften, Oldenbourg, München, 1996, pp. 123-143.

[Slater 1965]
Slater, P. E.: Role differentiation in small groups, in: Hare, A. P.; Borgotta, E. F.; Bales, R. F. (eds.): Small Groups, Knopf, New York, 1965, pp. 610-627.

[Slovin/Di Nunno 1994]
Slovin, Malcom; Di Nunno, Don: Engineering the Enterprise: Evolving Quality Systems, in: Software Engineering Notes, Vol. 19 Nr. 3, 1994, pp. 45-47.

[Snow/Miles/Coleman 1992]
Snow, C.; Miles, R.; Coleman, H.: Managing 21 st Century Network Organization, in: Organizational Dynamics, New York, Vol. 20, No. 3, 1992, pp. 5-20.

[Sodan 1994]
Sodan (ed.): Workflow Management Products, Uxbridge, 1994.

[Spiegel 1996a]
Klick in die Zukunft, Internet Serie (I), in: Der Spiegel, 11, 1996, pp. 66ff.

[Spiegel 1996b]
Goldgräber im Cyberspace, Internet (II): Das Geschäft der Zukunft - ganze Branchen werden sich ändern, in: Der Spiegel, 12, 1996, pp. 116ff.

[Stader 1996]
Stader, Jussi: Results of the Enterprise Project, in: Proceedings of Expert Systems '96, the 16 th Annual Conference of the British Computer Society, Specialist Group on Expert Systems, Cambridge, UK, December, 1996.

[Staehle 1991]
Staehle, Wolfgang H.: Management, Eine verhaltenswissenschaftliche Perspektive, Vahlen, München, 1991.

[Stonebraker 1992]
Stonebraker, M: The Integration of Rule Systems and Database Systems, in: IEEE Transactions on Knowledge and Database Engineering, 4, 5, 1992, pp. 415-423.

[Streitz 1992]
Streitz, N. et al.: SEPIA: A Cooperative Hypermedia Authoring Environment, Proceedings of the Second European Conference on Hypertexts (ECHT'92), ACM, New York, 1992, pp. 11-22.

[Striemer 1997]
Striemer, Rüdiger: Workflow Management und Groupware, Abgrenzung und Integration verschiedener Philosophien in der Vorgangsbearbeitung, in: Sorg, St. (ed.): Büroautomation/Bürokommunikation: Konzepte und Unternehmen im Umbruch, Tagungsband der 20. Europäischen Congressmesse für technische Kommunikation, ONLINE'97, Congress V, Hamburg, 4.-7.2., 1997, pp. C512.01-C512.12.

[Sydow 1993]
Sydow, J.: Strategische Netzwerke: Evolution und Organisation, Wiesbaden, 1993.

[Tapscott/Caston 1993]
Tapscott, Don; Caston, Art: Paradigm Shift, The New Promise of Information Technology, McGraw-Hill, New York, 1993.

[Teorey/Yang/Fry 1986]
Teorey, T.J.; Yang, D.; Fry, J.P.: A Logical Design Methodology for Relational Databases Using the Extended Entity Relationship Model, in: ACM Computing Services, 18, 2, 1986, pp. 197-222.

[Theuvsen 1996]
Theuvsen, Ludwig: Business Reengineering, Möglichkeiten und Grenzen einer prozeßorientierten Organisationsgestaltung, in: zfbf, 48,1, 1996, pp. 65-83.

[Thiétart/Forgues 1997]
Thiétart, Raymond-Alain; Forgues, Bernard: Action, Structure and Chaos, in: Organization Studies, 18/1, 1997, pp. 119-143.

[Tiemeyer/Chrobok 1996]
Tiemeyer, Ernst; Chrobok, Reiner: OrgTools: AfürO-Softwareführer für die Organisationsarbeit, Band 1: Geschäftsprozeßorganisation, Schäffer-Poeschel Verlag, Stuttgart, 1996.

[Tiemeyer/Chrobok 1997]
Tiemeyer, Ernst; Chrobok, Reiner: AfürO-Softwareführer OrgTools, Band 4: Organisationsinformationssystem, Schäffer-Poeschel Verlag, Stuttgart, 1997.

[Tsichritzis 1985]
Tsichritzis, D.C.: Objectworld, in: Tsichritzis, D.C. (ed.): Office Automation: Concepts and Tools, Springer, New York, 1985, pp. 379-398.

[Tueni/Jianzhong/Fares 1988]
Tueni, Michael; Jianzhong, Li; Fares, Pascal: AMS: A knowledge-based approach to tasks representation, organization and coordination, in: ACM, 1988, pp. 78-87.

[Turner/Kraut 1992]
Turner, J.; Kraut, R.: CSCW '92, Proceedings Conference on Computer-Supported Cooperative Work (Toronto, Oct.31 - Nov. 4), ACM, New York, NY, 1992.

[Unterstein 1994]
Unterstein, Michael: Datenbankunterstützte Organisationsanalyse, in: Kracke (ed.): Datenbank-Management, Teil 6/8, Weka-Verlag, Augsburg, März, 1994.

[Uschold et al. 1996]
Uschold, Mike; King, Martin; Moralee, Stuart; Zorgios, Yannis: The Enterprise Ontology, August, 1996.

[von Rosenstiel 1978]
von Rosenstiel, L.: Arbeitsgruppe, in: Mayer, A. (ed.): Organisationspsychologie, Poeschel, 1978, pp. 236-271.

[Vossbein 1990]
Vossbein, R.: Das Berufsbild des Organisators im technologiebedingten Wandel, in: io management, 59.Jg., 5, 1990, pp. 53-55.

[v. Kortzfleisch 1993]
v. Kortzfleisch, Harald F. O.: Rechnergestützte Organisationsmodellierung zur Unterstützung der Tätigkeiten von Organisatoren, in: Information Management (IM), 3, 1993, pp. 30-39.

[v. Kortzfleisch 1995]
v. Kortzfleisch, Harald F. O.: Werkzeuge für die computergestützte Organistionsgestaltung, Marktübersicht und betriebswirtschaftliche Beurteilung, in: Wirtschaftsinformatik, 37,4, 1995, pp. 384-392.

[Walsch 1997]
Walsch, Udo: Groupwarebasierte Organisationsmodellierung - Weitreichende Möglichkeiten oder schwerwiegende personelle und soziale Akzeptanzprobleme?, Working Paper, University of Paderborn, Department of Business Computing, November, 1997.

[Weiß/Krcmar 1996]
Weiß, Dietmar; Krcmar, Helmut: Workflow Management: Herkunft und Klassifikation, in: Wirtschaftsinformatik, Nr. 5, Oktober, 1996, pp. 503-513.

[Wendel 1996]
Wendel, Thomas: Computerunterstützte Teamarbeit - Konzeption und Realisierung eines Teamarbeitssystems, Deutscher Universitäts Verlag, Gabler Edition Wissenschaft, Wiesbaden, 1996.

[WfMC 1996a]
Workflow Management Coalition: Terminology & Glossary, Document Number WFMC-TC-1011 Document Status - Issue 2.0, June, 1996.

[WfMC 1996b]
Workflow Management Coalition: The Workflow Reference Model, Document Number TC00-1003 Document Status - Draft 1.1, by David Hollingworth, 12-Jun, 1996.

[WfMC 1996c]
Workflow Management Coalition: Workflow Standard - Interoperability Abstract Specification, Document Number WFMC-TC-1012, Version 1.0, 20-Oct, 1996.

[WfMC 1996d]
Workflow Management Coalition: Workflow Standard - Interoperability Internet e-mail MIME Binding, Document Number WFMC-TC-1018, Version 1.0, 20-Oct, 1996.

[WfMC 1997]
WfMC - Workflow Management Coalition: The Workflow Reference Model, in: Lawrence, P. (ed.): Workflow Handbook 1997, Wiley, Chichester etc., 1997, pp. 243-293.

[Whitaker 1995]
Whitaker, Randall: (Re-)Design of Self-Organizing Enterprises, Paper presented at the workshop on 'Groupware for Self-Organizing Units - The Autopoietic Turn in Organization Science and its Relevance for Groupware', in: ECSCW'95 (Fourth European Conference on Computer-Supported Cooperative Work), Stockholm, 10 September, 1995.

[Whitaker 1996]
Whitaker, Randall: Managing context in enterprise knowledge processes, in: European Management Journal, Special Issue on 'The Epistemological Challenge', 14, 4, August, 1996, pp. 399-406.

[Whitaker 1997]
Whitaker, Randall: Self-organization, Autopoisis, and enterprises, in: http://www.acm.org/sigois/auto/Main.html, April, 1997.

[Wildemann 1995]
Wildemann, Horst: Ein Ansatz zur Steigerung der Reaktionsgeschwindigkeit von Unternehmen: Die Lernende Organisation, in: Zeitschrift für Betriebswirtschaftslehre, Ergänzungsheft, Gabler, Wiesbaden, 3, 1995, pp. 1-23.

[Williamson 1985]
Williamson, O. E.: The Economic Instistution of Capitalism: Firms, Markets, Relational Contracting, Free Press, New York, 1985.

[Winograd 1995]
Winograd, Terry: A Language/Action Perspective on the Design of Cooperate Work, 1995, pp. 623-653.

[Winter/Ebert 1996]
Winter, Andreas; Ebert, Jürgen: Ein Referenzschema zur Organisationsbeschreibung, in: Vossen, Gottfried; Becker, Jörg (eds.): Geschäftsprozeßmodellierung und Workflow Management, Modelle, Methoden, Werkzeuge, International Thomson Publishing, Bonn, Albany, 1. Auflage, 1996, pp. 101-123.

[Wirtz 1996]
Wirtz, Bernd W.: Business Process Reengeniering - Erfolgsdeterminanten, Probleme und Auswirkungen eines neuen Reorganisationsansatzes, zfbf, 11, 1996, pp. 1023-1036.

[Wittlage 1995]
Wittlage, H.: Organisationsgestaltung unter dem Aspekt der Geschäftsprozeßorganisation, in: zfo, 4, 1995, pp. 210-214.

[Woo/Lochovsky 1986]
Woo, Carson C.; Lochovsky, Frederick H.: Supporting distributed office problem solving in organizations, in: ACM Transactions on Office Information Systems, Vol. 4, No. 3, July, 1986, pp. 185-204.

[Zisman 1977]
Zisman, M. D.: Representation, specification and automation of office procedures, Ph.D. dissertation, Wharton School, Univ. of Pennsylvania, Philadelphia, 1977.

[Zisman 1978]
Zisman, M. D.: Office automation: Revolution or evolution, in: Sloan Management Review (M.I.T.), 19, 3, Spring, 1978, pp. 1-16.

STATUTORY DECLARATION

I hereby declare that this Ph.D. dissertation contains no material which has been accepted for the award of any other degree or diploma in any University or equivalent institution, and that it, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference is made in the text of the dissertation. I have compiled this Ph.D. dissertation independently.

Paderborn, August 1998

Marcus Ott