University of Paderborn

Faculty of Business Management,
Business Computing & Economics

Prof. Dr. Ludwig Nastansky

Information Management, Workgroup Computing, Office Systems
Department of Business Computing

University of Paderborn
Warburger Straße 100

33098 Paderborn, Germany

Tel.:	(05251) 60-3368
Fax:	(05251) 60-3399
Internet:	NASTANSL.NOTES @ MHS.UNI-PADERBORN.DE WHILPERT.NOTES @ MHS.UNI-PADERBORN.DE
Notes:	Ludwig Nastansky @ WIUNIPB @ LOTUSINT Wolfgang Hilpert @ WIUNIPB @ LOTUSINT
Compuserve:	76657,420

The *GroupFlow* Framework:

Enterprise Model and Architecture of the Workflow System

Ludwig Nastansky - Wolfgang Hilpert

University of Paderborn

C-S-D-S

Client Server Distributed Systems

- Project Group -

A CSDS White Paper

Paderborn, April 1994

The *GroupFlow* Framework:

Enterprise Model and Architecture of the Workflow System

Keywords

Workflow management, workgroup computing, business process design, groupware, distributed organizations, wide area information exchange, Lotus Notes

Abstract

The *GroupFlow* environment integrates concepts that are typically referred to as *workgoup computing* or *Groupware* on the one hand, and *workflow management* or *business process design* on the other hand. *GroupFlow* offers business process and technology frameworks to set up versatile and flexible workflow systems for distributed information management within organizations and their outside communication partners.

Table of Contents

The *GroupFlow* Framework: Enterprise Model and Architecture of the Workflow System

Ludwig Nastansky - Wolfgang Hilpert

University of Paderborn

1. The *GroupFlow* Project Environment

In this paper, we will discuss (1) constitutive concepts of the enterprise information management model, (2) distributed data and object architecture layout, and (3) the procedures for practical implementation and application of the *GroupFlow* framework. *GroupFlow* is to be considered as a comprehensive system approach supporting business process engineering, deployment, monitoring, and the operational infrastructure of distributed workflow management frameworks within an organization. The *GroupFlow* system comprises the tools necessary for business process design and monitoring, as well as a complete set of operative modules for the various tasks and processes within a flexible LAN- and/or WAN-based workflow supporting runtime environment.

GroupFlow offers business process and technology frameworks to set up versatile and flexible workflow systems for distributed information management within organizations and their outside communication partners. We regard the synergetic approaches being used in the *GroupFlow* architecture, the design concepts, and practical implementation as innovative. The *GroupFlow* environment perhaps best can be profiled around integrating concepts that are typically referred to as workgroup computing or groupware on the one hand, and workflow management or business process design on the other hand.

The conceptual approach of the *GroupFlow* system with respect to business process modeling, distributed data and object architecture design, user and process/agent interaction is in concordance with research and discussions that are drawn along the lines of studies such as published in Harrington [1991], Keen [1991], Davenport [1993], Hammer/Champy [1993], Ishii/Ohkubo [1991], Marshak [1992], or Medina-Mora/ Winograd/ Flores/ Flores [1992]. Basic results of these works are reflected in this paper only to the extent of their impact on designing a pragmatic workflow environment which has its main focus on directly being applicable for deployment within organizations.

Also, business relevance discussions outlining the importance of business process and workflow oriented system approaches to corporate information management in increasing competitive and customer driven markets are not part of this paper. Without any question, though, they establish the essential motivating drivers for undertaking the efforts of developing a system with the complexity the *GroupFlow* framework constitutes. In another paper [Nastansky/Hilpert 1994], we have been focusing on complementary aspects of the entire *GroupFlow* framework, like e.g.:

- The appropriate business process paradigm underlying the actual design and deployment of workflow systems for business and public organization: *GroupFlow* is modeled around a continuous scale between cooperation and automation.
- The team processing architecture of *GroupFlow* making various classes of workflows on a continuous scale between flexibility and rigid predefined structures accessible for efficient rapid modeling of real life workflows in organizations.
- Some genuine aspects of the *GroupFlow* modeler *WOMED*, like seamless integration of a graphical design frontend in a distributed operative environment, or the clustering approach allowing layered modeling from the business process level down to the workflow runtime parameter settings.

GroupFlow has been implemented using Lotus Notes as the basic development platform and underlying distributed architecture. The user interfaces on the client sides are either based on Notes-native FORM and VIEW concepts, or developed using several other graphical frontend tools when

appropriate for the respective user tasks to be performed. On the backend server side of *GroupFlow*, solely Notes technology has been used for data repositories of the actual business information content, for the workflow structuring parts, and the set of workflow runtime modules supporting processes like messaging, replication, event management, gateway connections, or cross-platform data exchange and process coordination.

2. The *GroupFlow* Enterprise Information Management Model

According to Bracchi/Pernici [1984] the major target of conceptual office modeling is to make office information system design easier and more reliably. Thus, goals for a conceptual design of enterprise or office models are:

(1) obtain a description of the enterprise. The large number of exceptions and special cases makes it impossible to reach a complete formal description of the office. However, the model should describe as many aspects of the enterprise as possible in a definite way. It is important that this description is comprehensive not only to an analyst or systems designer but also to the staff members, managers and planners in the office. They should be able to validate the system, suggest modifications, identify inconsistencies, and maintain and further develop the system themselves.

(2) Identify processes that are no longer useful in achieving the current enterprise goals but are still performed only because of habit. These processes are to be modified or completely replaced. In literature we find many different approaches for office or enterprise models. Some of them are discussed more deeply in Desai [1991], Dutton [1993], Lochovsky/Hogg/Weiser/Mendelzon [1988], Kubota/Ishii [1989], or Tueni/Jianzhong/Fares [1988]. An enterprise model that is applicable to workflow design and processing obviously has to subsume at least the following layers of information management within the organization:

The information flow of an organization, the sequence of activities, the agents and resources such as abstract office workers or organizational units playing specific roles when performing activities. The enterprise model is to encompass the decomposition of the whole application environment into a workflow-based sequence of tasks, the resources required for these tasks, and the objects accessed and manipulated during task processing. Also, task and process decomposition require to refine lower-level task activities and sub-level processes. Any information and routing specification relevant to the authentic business applications being supported within a workflow framework is to be encompassed by the enterprise model as well.

Fig. 1.1 represents the fundamental structure of the enterprise information management model underlying *GroupFlow*. It consists of three distinguished but interdependent sub-models: (1) the *infrastructure* model, (2) the *process* model, and (3) the *information processing* model (*Fig. 1.1*).

Infrastructure model	Process model	Information processing model
unit agent role workgroup resource relationship	business process task activity rule time	object type information links folder

***Fig. 1.1*: Outline of the Enterprise Model underlying the *GroupFlow* Framework**

(1) The infrastructure model defines the structural components of the organization which can be represented within *GroupFlow*. This infrastructure model comprises the people who work at different locations/positions in different organizational units of an organization, the resources (e.g. tools) they use to perform their tasks, the formal groups/teams they belong to and the relations and formal communication paths between these groups. The infrastructure model uses terms to model the

organizational structure such as unit, resource, role, workgroup and agent. It determines the interrelationship between the units such as superior units. Also, the assignment of roles to agents as well as the specific position of agents within units are specified.

An agent can be both, a human/person actor or an automated software agent. A human agent holds a position related to an organizational unit within the infrastructure model. Access rights for individual actors are defined implicitly by organizational structures and may be inferred from the placement of an agent within the organization. Human agents potentially play different roles to perform office tasks which in turn imply certain access level/security privileges to be used in order to read and/or manipulate document objects. Human agents have different qualifications due to their personal education background, which must be considered when agents are assigned to tasks within a process. The assignment of system resources to organizational entities such as agents, roles and organizational units have to be considered.

Automated software agents perform a variety of information processing and communication procedures automatically using specific software modules. The processing is being performed on various data types stored in document objects to be considered as generic routable information containers within the *GroupFlow* framework.

Software agents that are invoked by the *GroupFlow* framework automatically for an asynchronous task performance retrieve documents or images from archives and store the back into archives. They could also issue database queries in order to prepare comprehensive data collections to be used within a task. Also, any specific and highly customized software module is to be launched by and gives process control back to the *GroupFlow* system via a generic interface component.

A formal unit combines/matches the human agent with the organizational structure. A unit describes the elementary components of the organizational structure. The enterprise model underlying *GroupFlow* allows a unit to either be elementary, defining the position and job of a staff member, or to be aggregated, like divisions, departments or workgroups which can consist of one or more subordinate units.

Organizational units reflect more or less stable components within a disjunctive organizational hierarchy. Workgroups, on the other hand, offer flexible concepts for perpetual organizational (re-) modeling. Particularly informal, short-lived task forces can easily be built/grouped together.

In general a role describes what the actor of a role is expected to do - and likewise not to do - within his social environment or within a special process. In the *GroupFlow* enterprise model the role combines a set of activities that is generally carried out by an individual or group within the organization.

A role such as the process manager would include numerous activities such as planning, reporting, monitoring, managing staff, working with the client, etc. The role of the process manager may be acted by a one person within a specific workflow and by another person within the next workflow. The role is distinct from the agent who acts it. A whole group of people can be acting a role and the group itself may change in time as people arrive and leave. A role can be understood as a type or class which has to be specialized into user defined instances. Many instances of a role can exist at a given time within an organization.

A role includes the right or competence to use different resources or to change objects. Competence serves as a generic term for responsibilities within a process and the rights necessary to complete a process, for instance to make use of resources. Competencies are primarily not directly bound to an agent but are assigned to a specific role and the related unit. This results in the rule that a competence assigned to a role is passed over to the related unit and that competencies of a unit are in turn passed over to an agent.

(2) The workflow routing specifications are based on the process model of the enterprise. It holds information about the business process layout and definition, the involved agents, resources, etc. This module focuses on the very process dynamics/specifications, their interdependencies, the several steps to be taken within a process, their serial or parallel sequences, etc.

As opposed to workflow systems supporting automation of bulk data processing tasks within rigid sequencing-patterns the *GroupFlow* approach primarily is user and team focused in supporting the actual processing of business data. In this user-interaction focus *GroupFlow* is taking a similar starting point as that used in Medina-Mora/ Winograd/ Flores/ Flores [1992] and being the underlying architecture of Action Technology's™ workflow system. On the other hand, *GroupFlow* extends as well to workflow systems which aim at complete automation as their principal goal. Thus, the core of the *GroupFlow* application architecture is modeled around the very groupware paradigm underlying its host environment Lotus Notes, and extending its functionality to a scale ranging from single user interaction to automatic software agent processed operations.

Using this approach, a *GroupFlow* task representing a workflow step of the supported business process comprises the following options for processing agents:

- a user is processing the task in an interactive fashion (user agent), or
- a software process is performing the task automatically, processing data in a workflow event driven schedule (software agent).

Typical interactive user agent processing includes: form and document oriented processing (i.e. 'fill-in-the-form' type work, rich document processing around many 'natural' data types [images, speech annotation, video], both based on Notes-native compound document editing), synchronous work with document embedded objects based on their supporting tool server environment (i.e. OLE, on MS Windows or OS/2 workstations), or supply of appropriate business parameters and initiation of asynchronous software-processes to be performed in finishing a task.

Typical software agent processing includes: macro agents performing scripted higher level command sequences, initiation of foreign processes (i.e. starting a transaction for a transaction driven environment, starting a batch process in a remote host or server environment), delivering a data set to a remote environment (table based formatted data, parameter set, semaphor or status type data structures, unformatted content [message text, binary object content]), or receiving a data set from a remote environment respectively. Using the software agent framework, various application functions can be included as automated intermediate steps into a *GroupFlow* based system. Examples include: content filtering and structuring of information streams provided by newswire services, integration of E-Mail based information in phone communication environments (pager, voice messaging, speech synthesizing), data exchange with SQL-based DB-repositories or transaction systems.

The defined office procedure classes and their respective workflow routing specifications are distributed via the built-in replication mechanism of the underlying development and application platform. Active as well as completed and archived workflow instances are disseminated in the same fashion. The replication engine supports the central coordination of the workflow specification and routing.

The actual routing and status management is handled in a local manner: the so to speak "intelligent" document knows its way through the office when it compares its actual status with the also decentralized available routing information in order to find who is going to be the next agent for further task performance.

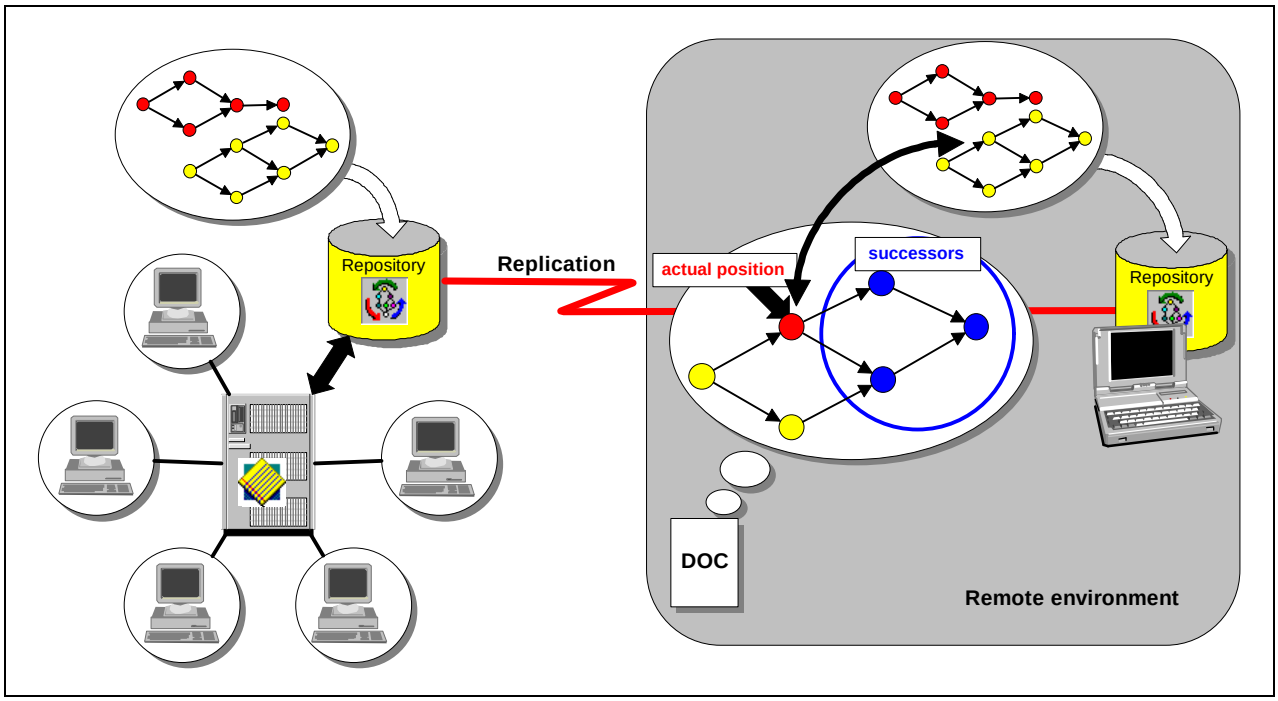

Fig 2.1: **Distributed Process Control**

(3) The *information processing model* traces the outline of the object types or information items such as messages, documents or forms that are manipulated and exchanged within the organizational infrastructure. It also describes the information links that can exist between the several objects.

Information items or document objects can be included into folders. Some of the included documents may reside in a folder for information purposes only without being changed while others are to be worked on or created. We distinguish between job folders, jobtask folder and task folders. A job folder contains all document objects related with one job. A jobtask folder is part of a job folder being processed at a specific task or workflow step. The entirety of all jobtaks folders being processed at one particular task over various workflow instances of a specific workflow type is called the task folder. Documents may be assigned to folders at design time as well as dynamically during runtime.

The functionality that actually drives the workflow routing is captured in *program objectssets* that are linked dynamically into the information object an hidden during user interaction. Thus, the user is presented only one common front-end that contains both the application data as well as the routing functionality. The system architecture is designed in a modular and layered manner that subsequently will be further explained and detailed if necessary for the purpose of this paper.

3. The *GroupFlow* Distributed Data and Object Architecture Layout

The architecture of the *GroupFlow* system framework (*Fig. 3.1*) consists of three major components that are outlined in a layered architecture model:

(1) the back end components managing the structural information of the workflow as well as the messaging and synchronization activities,

(2) the target application(s) being embedded in various distributed and networked workflow-driven tasks including their associated data repositories, and

(3) a tool environment, based on a set of independent interactive graphical tools, enabling a variety of workflow related functions, like workflow modeling, redesigning, analyzing, monitoring, or content (re-) structuring functions.

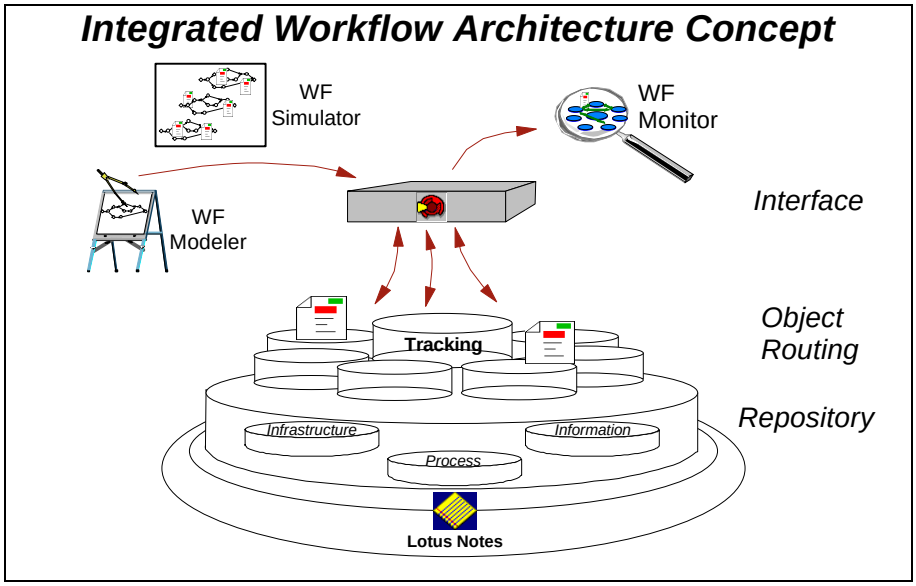

***Fig. 3.1*: *GroupFlow* Framework**

(1) The *GroupFlow* back end components encompass the distributed workflow structure repository, the workflow-protocol or routing-status tracking functionality, and the replication and workflow routing engine.

Compared to workflow systems being offered on the marketplace *GroupFlow* is innovative in its back end architecture: The workflow routing engine has been completely modeled around the rich functionalities for LAN-/ WAN-wide communication support inherent to Lotus Notes message dissemination properties complemented by its genuine replication-driven information sharing and accumulation paradigm. The workflow repository reflects the entities that are relevant to business process design and management. Based on a composite enterprise model the repository comprises structural information about the dynamics of the various business processes and the general organizational structure as well as internal application design specifications. The scaleable business process structuring model defining a workflow continuum that underlies *GroupFlow* is being discussed in Nastansky/Hilpert [1994].

The protocol information stored in the *GroupFlow* repositories is further used to perform analysis comparing actual with planned, allowing anomalies to be spotted and corrected.

(2) The *GroupFlow* target application side encompasses the entities defining the authentic application functions of the business processes being enabled by a workflow system.

Much of the processing within the *GroupFlow* system is performed based on predefined Lotus Notes-native form, view and macro/script templates, that are ready to run, but also customizable to organization specific requirements. On the other hand, *GroupFlow* is to be considered completely open for interconnecting incoming as well as outgoing processes within a LAN- / WAN-based cross-platform environment by providing a variety of bootstrapping, hooking, or event handling concepts. The target applications have access to and are controlled by the workflow back end engine.

Continuous processes with certain routine characteristics are best modeled in shared document databases: The involved persons know about the existence of information in the database and the respective tasks to be performed, or responses expected. Thus, staff members concerned with these

workflows consult the database checking for new documents. They apply a sort of pull approach towards information. They will react by working on existing documents and modify, complete, add data to documents, or set status fields. Also, within this share model users may be forwarding, copying, archiving documents, creating response documents, or initiating new documents.

The complementary send model is basically applied for rather simple ad hoc routing applications. It is easy to set up a new E-Mail based workflow. Usually the addressee does not expect requests beforehand. The focus of the next agent's attention is pushed on incoming documents to be worked on or to be forwarded. In such a mere point-to-point routing scenario it may be difficult to track the status of a specific workflow at a given moment when documents are sent from one actor to another. Typical sample ad hoc workflows can be found in project management of individual tasks or customer requests that cannot be matched with any known service of the organization.

The integration of both the send model and the share model as technically supported by the Lotus Notes platform can efficiently result in a variety of combinations over a wide spectrum. This spectrum stretches between E-Mail based applications including mechanisms like automatically sending a protocol entry into a shared database for easy tracking of the routing status on the one hand. And, it extends to usual shared databases which employ a pull approach, focusing the actor's attention on a newly inserted or modified document in the shared database on the other hand. In our opinion this almost continuous transition and combination of the two models provides a far better flexibility and reflects real business process requirements better than both, mere transaction based process management systems or pure E-Mail based routing applications.

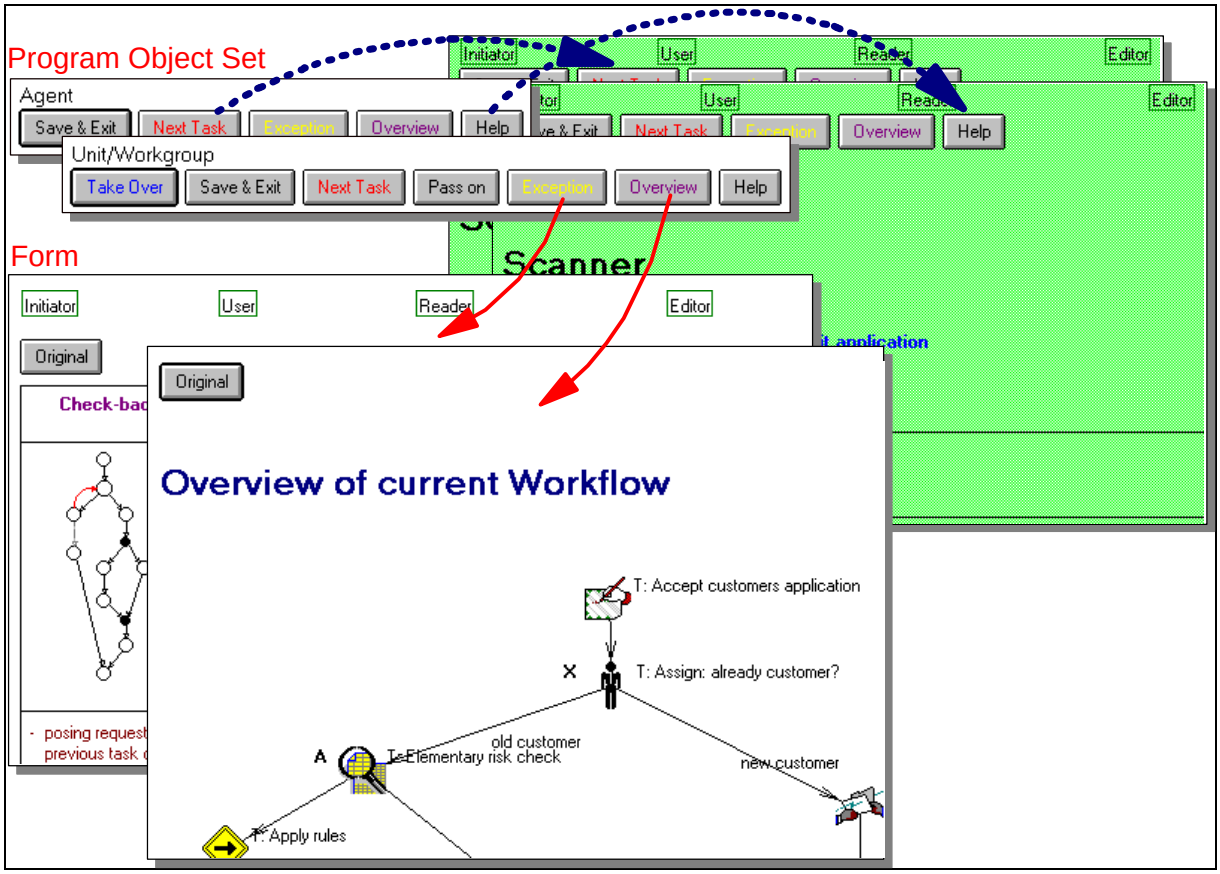

***Fig. 3.2*: Network of Workflow related Program Object Sets**

Regardless of the application either of the send or the share model a modular design of the workflow related functionality has been applied: the so called workflow program object sets are actually linked into target database applications (Fig 3.2 dotted lines in the upper part). The functionality is

linked into customizable forms within the target applications from pre-defined program object sets residing in the information processing module of the repository. The functionality driving the workflow is thus made available to distributed workflow applications being programmed only once, stored in the repository, and transmitted utilizing regular/built-in replication engine of the underlying application development platform of Lotus Notes. Any modification within the functionality of the core routing engine can thus be easily applied within various distributed target database applications that are tied into the routing engine of the back end system.

In summary this linking functionality enables a lean system design because it prevents life cycles of code generation, re-compilation, etc after modifications. The user accesses the functionality of the program object sets within any application specific form without noticing the underlying mechanism. The program object sets invoke functionality such as routing the workflow consisting of document objects to next agent, displaying a graphical map of the workflow for the user's orientation, launching the exception handling, etc (Fig 3.2 continuous lines in ther center of the figure).

(3) The *GroupFlow* tool environment provides an open set of independent interactive graphical tools enabling a variety of workflow related functions. Currently, the *GroupFlow* system offers as its key tool environments: *WOMED*, a graphical workflow modeling editor, and *WOMON*, the graphical *GroupFlow* Monitor. *WOMED* supports both workflow top-down as well as bottom-up design, dynamic clustering, update, redesign, and simulation of workflows. The organizational layout can also be graphically modeled and related to workflows using the higher clustering levels of *WOMED*. Any data defining the graphically modeled specifications are stored in Lotus Notes database objects and hence can be exchanged across distributed locations using the replication mechanism.

The *GroupFlow* Monitor *WOMON* supports various message tracing functions between agents. Particularly planned and actual workflow routing can thus be visualized and compared. A subset of the monitoring functionalities - the *GroupFlow Analyzer* - visualizes document routing and message exchange within any Lotus Notes database application. The results of this kind of analysis allow to derive information for possible re-design options for the workflow with *WOMED*.

All these tools are being designed as graphical frontends which are linked directly into the very runtime data structures supporting the operative tasks of the *GroupFlow* backend components.

4. Procedures for *GroupFlow* Specification

In order to actually utilize the *GroupFlow* system in a comprehensive workflow application a few major deployment phases have to be run through. It is important to recognize that these phases are highly interdependent. In an actual installation, typically it will be necessary to iterate back and forth between the outlined phases.

The phases are as follows:

(1) analysis and workflow concept design,
(2) implementation of the workflow system,
(3) application of the workflow system, and
(4) monitoring and analysis of actual workflow

4.1 Workflow Concept Design

In a first phase the business process side of an application has to be analyzed and (re-) constructed before the actual implementation of the workflow system can go under way. This involves the investigatation of existing or planned processes to eliminate potential bottlenecks and / or redundan-

cies. The flow of the information, the decision points and the various forms of data and information presentation have to be investigated.

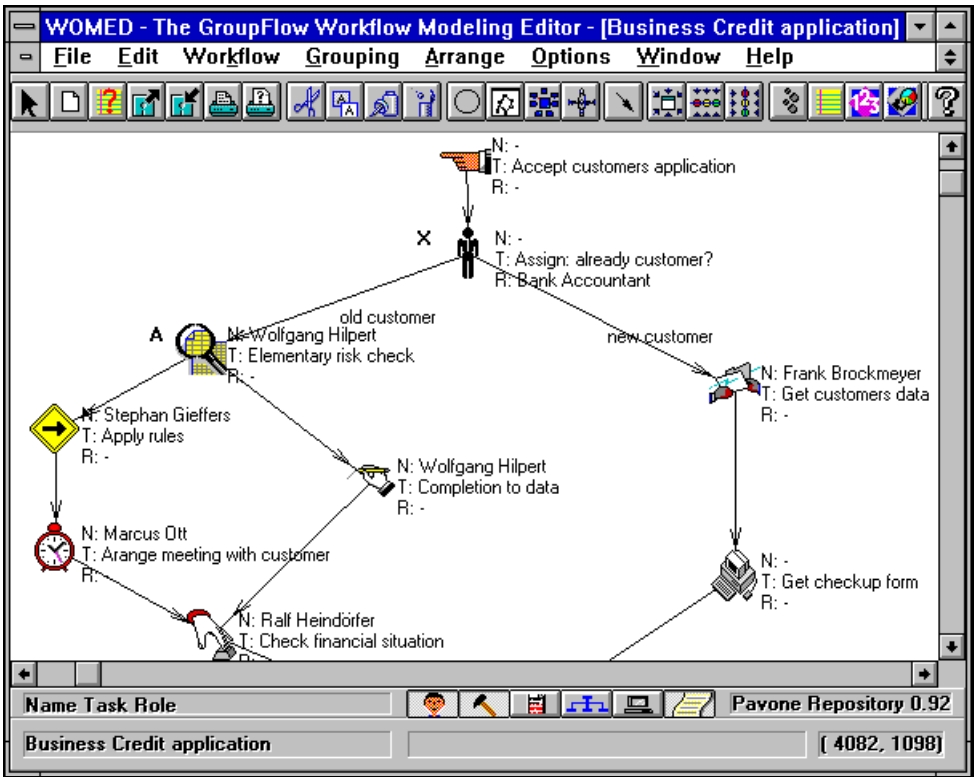

Figure 4.1: The workflow modeling editor

In this first phase the concept of the process is developed. This includes the design of a graphical workflow model as well as of the related organizational structure layout. The *GroupFlow* modeling tools support the efficient presentation, discussion, and refinement of the new system design. In order to support the ex ante analysis of the business process model the derived workflow can be simulated. The simulation may focus on static structural aspects based on the workflow visualization, or, it may include dynamic sequencing patterns based on basic object flow simulation in the workflow graph. Thus, consistency checks and possible bottleneck detection are possible beforehand.

4.2 Implementation of the Workflow System

In order to implement/set up a workflow application the graphical representation of the workflow as designed with the modeling editor is automatically transferred into operable workflow definitions. The graphical workflow model consisting of nodes and edges with its embedded specific properties and attributes is logically transformed and stored as operable routing specifications in the workflow repository. The transformation process involves both conceptual and technical interface capabilities.

The templates for the operative applications - or more specifically: the target workflow-enabled database applications - interoperate with the workflow repository. These applications are used to activate the document object routing. They are designed to be driven by the specifications entered in the workflow repository. These templates are used as prototype applications and thus form the basis for the design of the actual operative application databases. The target applications access the workflow repository to use its specifications and to manage the actual workflow routing. To do so, the templates contain certain/fixed workflow related elements such as workflow specific views, forms, part of forms (field sections), fields, and a set of powerful and full featured action buttons

giving direct access to a variety of rich workflow-related functionalities. To give an example, by clicking action buttons end users can route a task from one agent to the next agent(s) according to the repository entries entered by the graphical workflow modeler.

These pre-defined elements of the target database applications comprise the core workflow functionality. They can be integrated into any organization specific application design when customizing the templates for actual operation. This concept allows for both, workflow-enabling existing Lotus Notes database applications as well as building completely new cross-platform workflow-based applications using Lotus Notes functionalities by optionally including external processes or connnectivity tasks. The mechanisms are always the same: The *GroupFlow* templates are adjusted for the requirements of the target enterprise, this customization typically including tasks like the integration of organization specific forms, logos, tabular structures, fields, views, or reports.

4.3 Workflow Routing

For the application of the workflow system the users initiate actual workflows instances based on the specification and design performed at the former steps. The activation of predefined processes by creating document objects that are routed within the target workflow applications hides all of the background components such as the workflow repository from the end user in their everyday office work. Nevertheless the document objects will be routed automatically in accordance to the specified routing paths as defined in the workflow repository.

Agents create document objects that are routed through the workgroup at initiation nodes. Various alternatives exist of how a (human or process) agent can route a workflow object to the next appropriate agent in the chain of accurate workflow execution. Here, from all possible constellations we will cover just some basic aspects of the end users view on workflow management. First, we will describe the interaction of an end user dealing with standard and predefined workflows. Second, the combination of these standardized workflows with everyday ad hoc changes and refinements of such routes will be focused on.

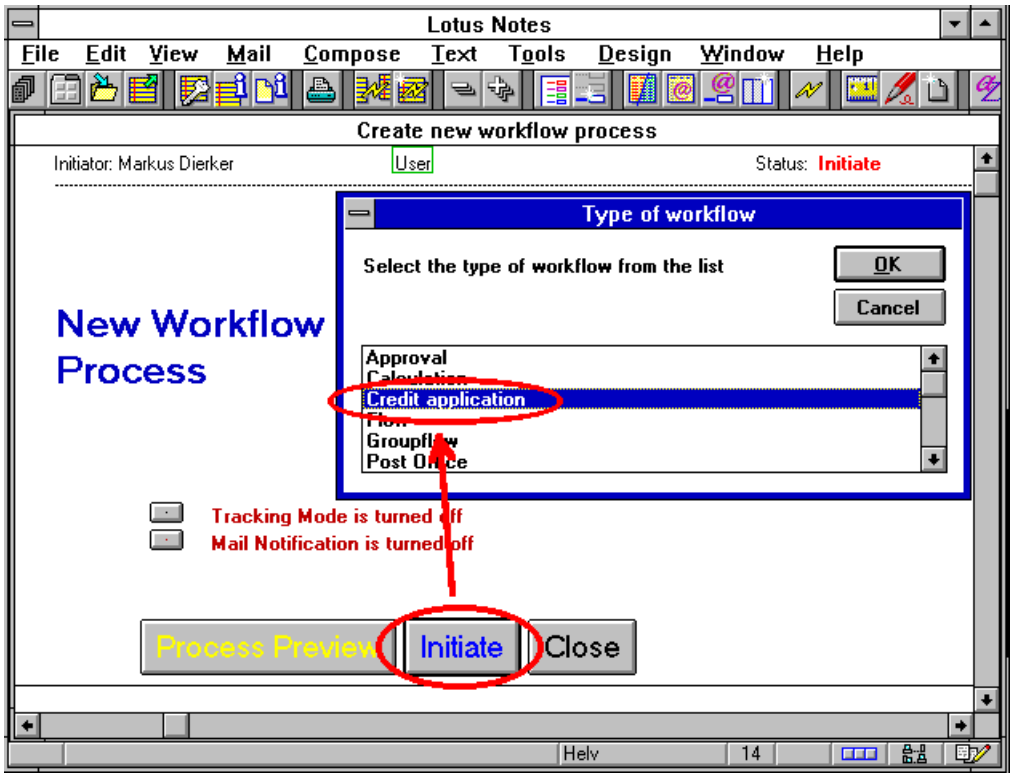

***Figure 4.2*: Compose new document routing object and choose type of workflow**

Finally, some ideas for the integration of standardized workflows with open processes will be presented.

(a) Standard Routing of Predefined Workflow

The end user initiates the object routing by composing new documents in the target workflow database application.

Either the workflow application has been setup to let the user choose and compose a specific document type. This implies already the specified workflow type to be applied. Alternatively, one common workflow startup form is used for more specific and workflow instance related setup. In this case, the underlying type of workflow can be choosen according to the access privileges of the user.

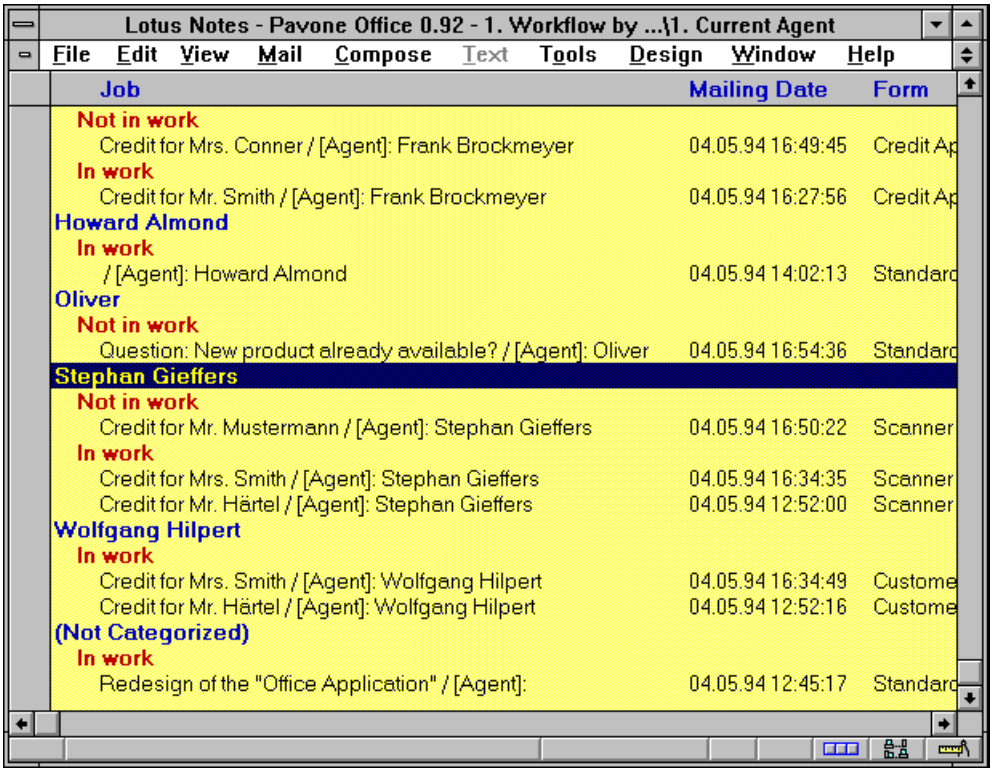

Figure 4.3 **: View "Next agent"**

All agents with access to the shared databases can view their tasks in several different contexts or views. The current agent of a workflow finds her work to be done in her work assignment area. The scenario described here is based on the shared database approach as described above. There are many advantages to this solution, the most important being the team responsiblity for keeping track of processes. Still, there may be situations where a quick response is necessary and users may have to be notified about new tasks they are being required for completion. This implies the need for the application of a send approach for the workflow routing.

Within the *GroupFlow* system we apply a combination of both models: basically, the document objects are hosted in common, shared databases accessable by all team members.

In order to comply with the required notifaction of the next agent for an on-time task completion the above mentioned action button to be pushed by the current agent not just (logically) routes the workflow to the next agent by assigning her as the new current agent. It also initiates automatically an E-Mail notification supplied with a hypertext-/document-link that leads the addressee directly to

her new task. This means, that the end user is notified of the arrival of new tasks both in his public task list as well as in the private E-Mail environment.

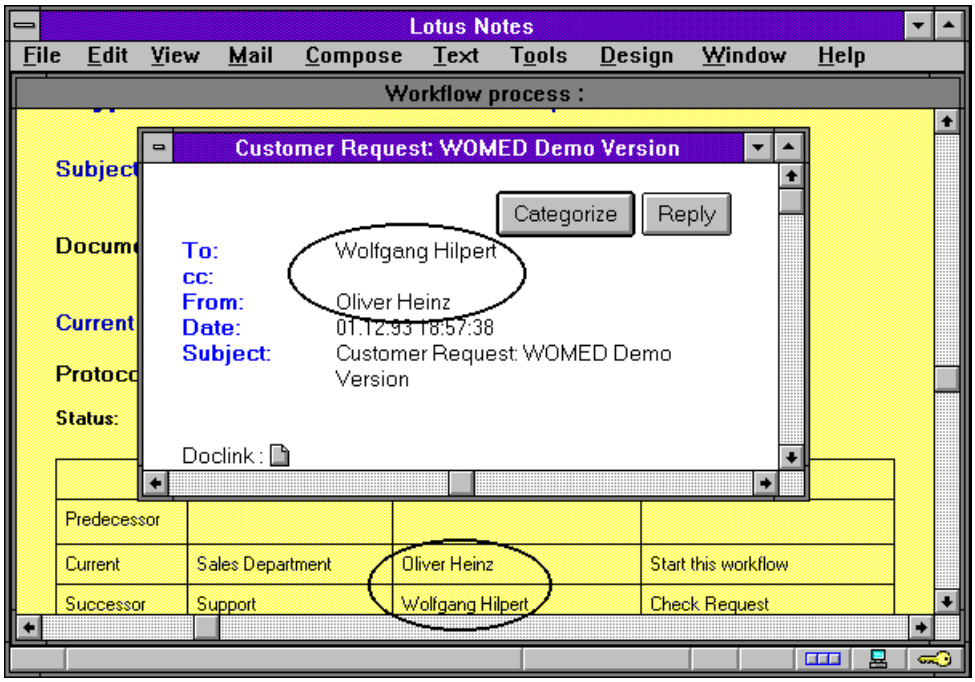

***Figure 4.4*: Automatic mail notification of the next agent in workflow**

When opening a task or document object, the agent will automatically get the desired view of the task according to the specifications in the workflow repository. This includes mechanisms such as a context driven assignment of the appropriate form which has to be used to present the current content of a data set to be worked upon to the acting person. These concepts support the different involved agents in completion of their partial responsibilities within a complete workflow in a task- or role-specific manner.

In the example of a credit application this may imply that one agent can only read and edit the address parts of a complete customer document while the succeeding agent will be capable of filling in confidential data about the financial assets of this customer. Thus, in general agents have access only to those sections of circulating document objects which they are asigned for.

At any state of the workflow the current agent is informed about the context of his tasks related to both, his predecessor's and his successor's reponsibilities, as supplied by the *GroupFlow* form design modules. The context information is derived from the repository entries.

After a task has been completed the user can easily route the document object to the next specified agent of this workflow. Or, he can put the document back on his work assignment area. He can do so by pushing the appropriate action button supplied for this purpose.

When pushing the button for the 'next task routing' the user incorporates a key feature of the *GroupFlow* architecture. A background process determines who will be the next person in the routing chain based upon the process specifications in the workflow repository for the chosen type of workflow and the individual circumstances of the current case. For example, the *GroupFlow* system will lookup all alternative routing paths from the current agent's task and evaluate the one to choose based on the data that has been entered by the current agent.

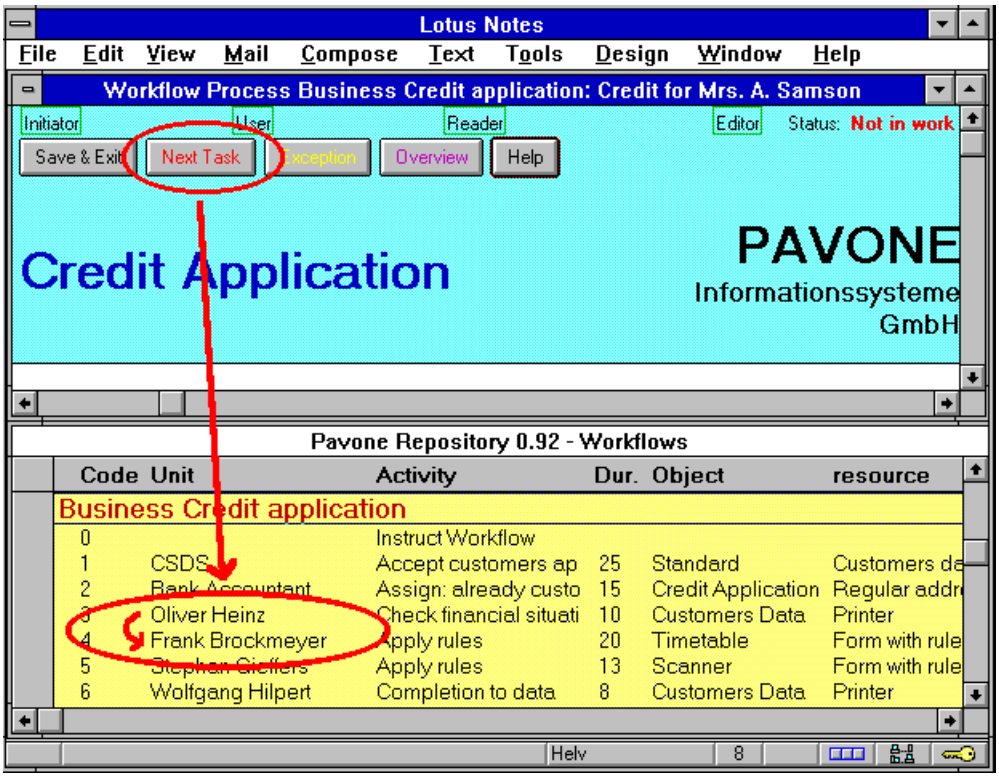

***Figure 4.5*: Workflow routing to next agent based on repository entries**

Another example is a next task that has been assigned to a whole group of prospective agents. Either this group has been defined as a workgroup, is conformed by the people assigned to a formal organizational unit, or it includes all the actors that are to carry out the role that the task has been assigned to.

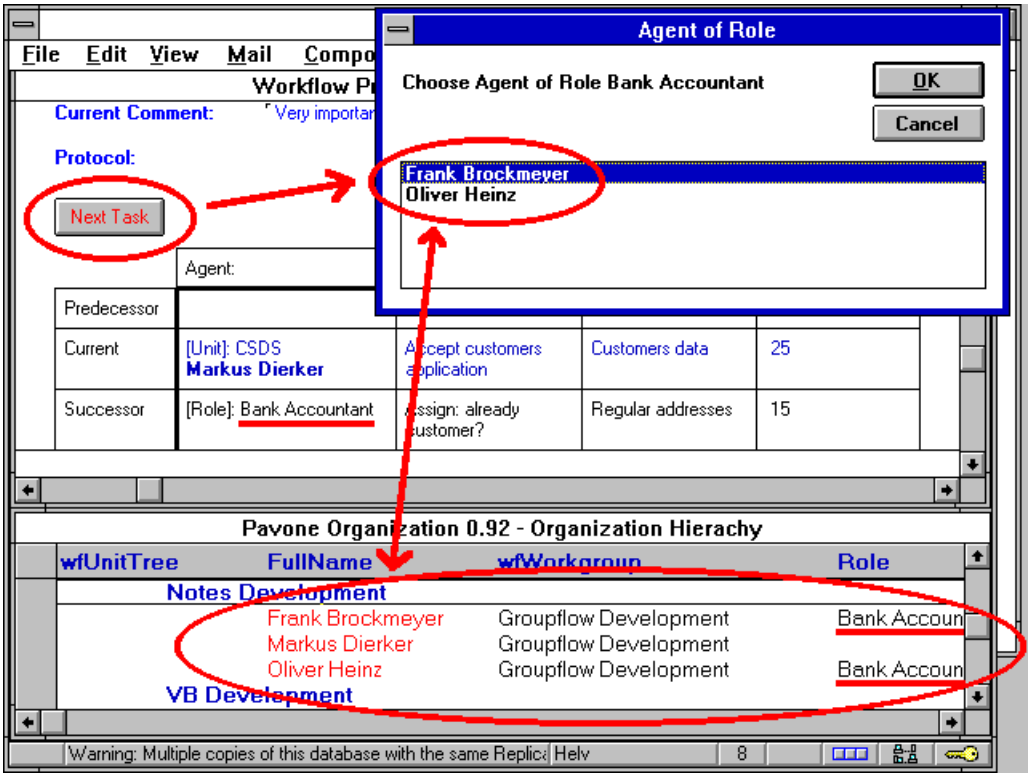

***Figure 4.6*: Specify next agent more precisely at runtime**

In all of these cases the work could be left for the whole group to do. Thus, he task appears on the to-do list of all group members and will only disappear after one actor completes the task. Alternatively, the agent carrying out the task before, specifies *one* person out of the group when routing the document object after his task has been completed.

Within the *GroupFlow* framework it is also possible, to incorporate automation features as appropriate, an example being workload balancing agents. A background agent (= process) could check the to-do lists of the various alternative agents that are members in the group of current task assigments. The background process would then determine the next agent that the task is to be routed to according to certain criteria such as number of tasks, estimated duration, etc.

The workflow routing can easily be structurally changed/modified by adapting the graphical specification of the workflow model using the *WOMED* tool. This change can be instantaneously relayed back to the workflow run-time/process execution environment if necessary. Thus, the functionality of the *GroupFlow* system enables a seamless integration of easy to apply changes to predefined workflow specifications, both, for the phases of principal design as well as for the runtime environment.

(b) Flexible ad hoc Modification and Exception Handling of Predetermined Workflows

The *GroupFlow* architecture supports modification of the standard routing sequence *within a predefined* workflow at run-time by ad hoc modifications and dynamic re-routing of tasks for special cases and exceptions (i.e., without any structural change or modification of the workflow as discussed in the previous praragraph). This allows for tight integration of standardized and predefined processes with flexible ad hoc changes to the workflow definitions that may be required by specific circumstances during everyday work. A workflow system that does not provide the flexibility for the user to respond to this highly probable type of real life necessity forces him or her to leave or interupt a well defined context of task processing - thus possibly causing fatal disruption. The *GroupFlow* environment integrates a set of structured mechanisms to handle exception or disruption problems of various kinds in predefined workflows [Nastansky/Hilpert 1994].

The possible modification rules applying to given workflows have been modeled after common exception situations in business processes, such as:

- *check-back* with previous agent,
- *inquiry* or question to anyone else,
- *task delegation* invoking adetour in routing pathor
- complete *change* of the underlying workflow type such as a "cross selling".

In order to apply an ad hoc change the user clicks on a specific action button and fills in some information describing the exceptional situation. As with the two leftmost ad hoc changes in *Fig. 4.7*, the accustomed E-Mail interface will be evoked. All of the context relevant data will automatically be inserted including a hyperlink (*DocLink*) into a newly created message object that will handle individual the exception.

If the current agent has to pose a question to another person that is not yet involved in the workflow, or wants to confirm a particular fact or item, he or she can choose the addressee from a list. An E-Mail will be generated that offers automatic context notification such as the workflow reference as well as a hyperlink/DocLink into the referred document.

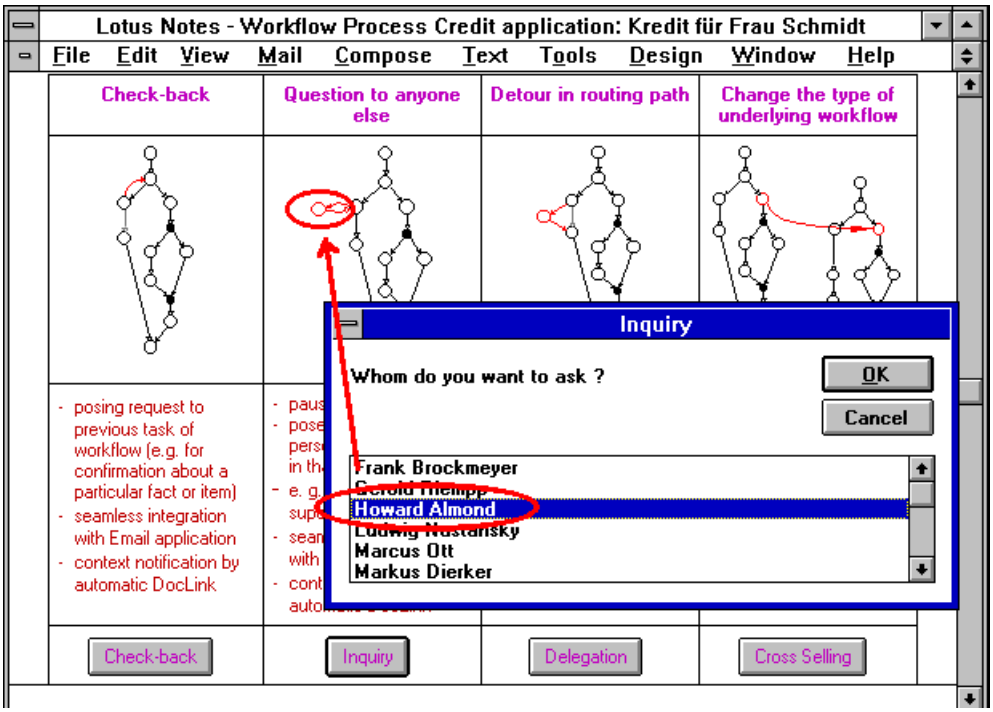

***Figure 4.7*: Choose Agent to involve ad hoc into predetermined workflow**

Exception management necessitates rather complex considerations in the information management model underlying the *GroupFlow* approach. For example re-routing a task implies the delegation of tasks within a predefined workflow at run-time. Managing a delegation requires authorization granting, collecting of the task description and pending job context by the *current* agent and routing to the new delegated handling agent including automatic E-Mail notification, setting up the resume entry points, and more.

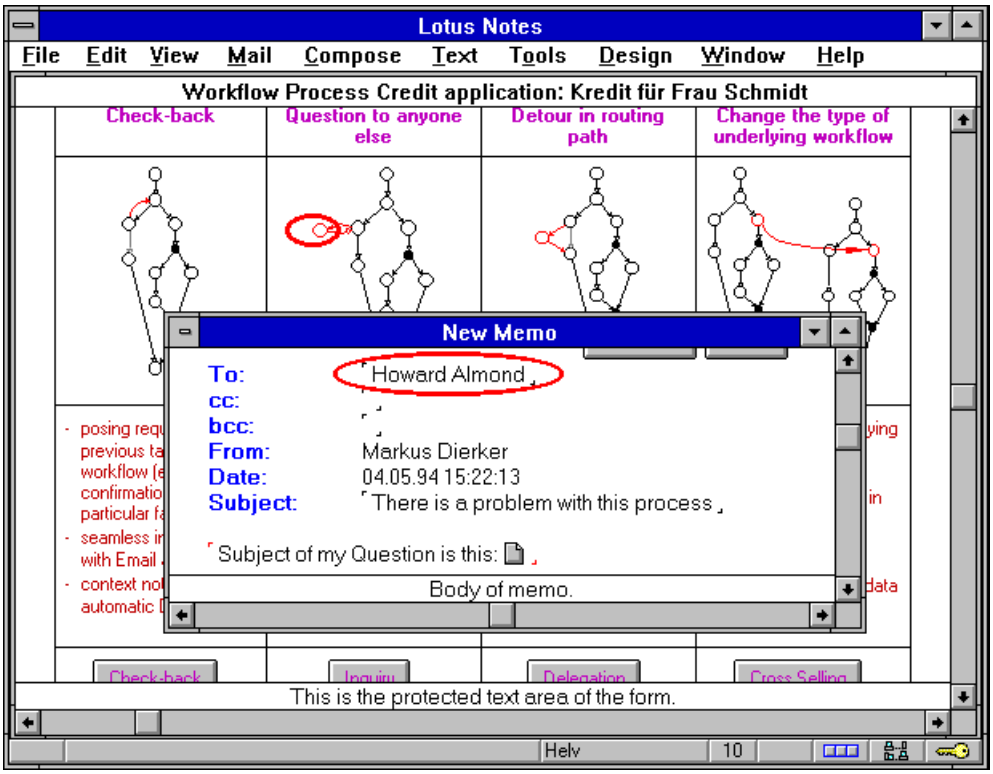

***Figure 4.8*: Automatic E-Mail integration by ad hoc workflow modification**

In contrary to the interactive exceptional cases of the *check-back* and the *inquiry*, a *re-route* implies that the control over the document is handed over to the new agent. The new agent will then route the document to the next agent as the resume entry point who was originally modeled as the successor. In order to apply are-routing the appropriate access level is required.

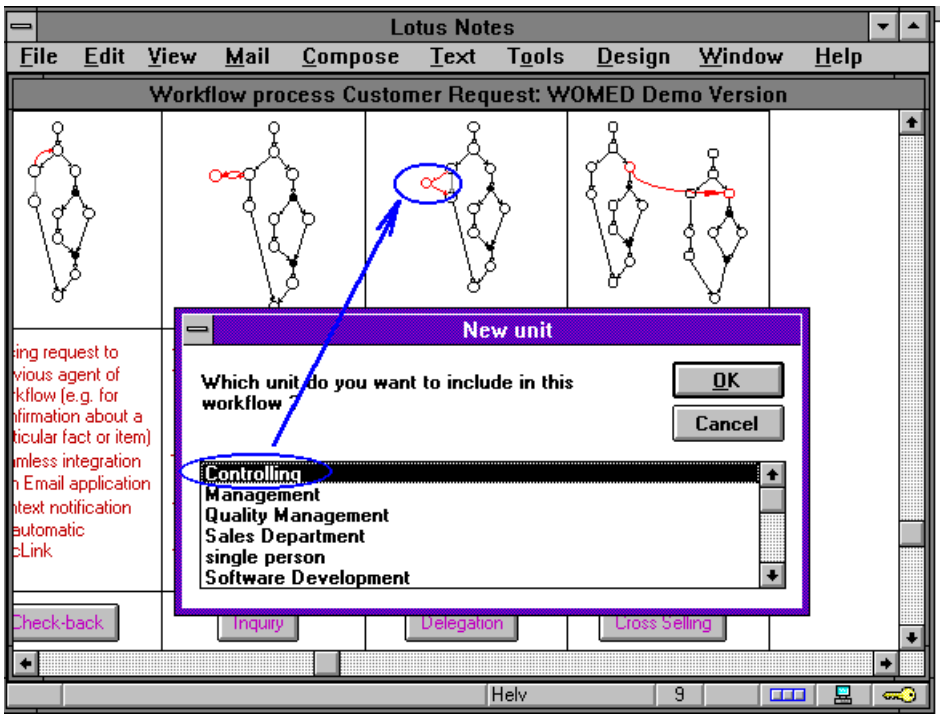

***Figure 4.9*: Re-routing the workflow: Delegation of new task to be done**

Another example is workflow change. Causes for complete *change* of an underlying workflow type are manifold in real life business processes. *GroupFlow* allows for a job, having been routed part way through one process, to seamlessly swap to a different process, transferring the relevant information, and initiating the second process at the appropriated workflow step.

The following cross selling scenario further explains this concept: A customer arrives at a bank to obtain a loan. After taking the basic details and progressing the application it becomes clear that the customer does not have enough security to support the loan, and the application cannot be progressed any further. An alternative would be for the bank to offer starting a savings scheme where it would be possible to take out a loan later. This naturally requires different application forms, and in a conventional system the application would need to be reprocessed from new. Often much of the data entered previously would need to be reentered, and possibly by a different person - as it may be a different department that is responsible.

Another *ad hoc* change to a well-structured workflow would be a *rollback*. The *previous* agent discovers that his task in the workflow requires an additional activity *after* he has routed the task to the *next* agent. Now, a secure way of getting back the control over the workflow has to be executed. The protocol for this would be a pre-defined negotiation via E-Mail exchange.

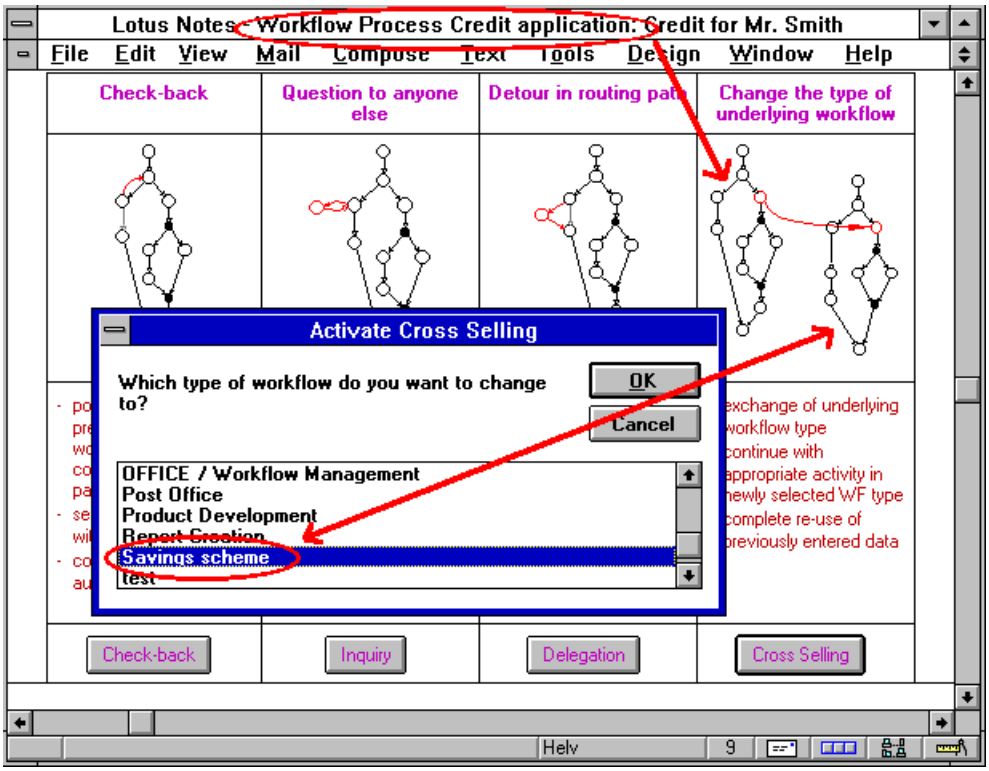

***Figure 4.10*: Cross selling changes underlying workflow type**

As with all other activities of this workflow system, the exception handling to the standard workflow specifications must be thoroughly recorded. The audit trails can be found as entries in the workflow protocol. If for instance a certain inquiry pattern recurrs very often at a certain task within a type of workflow the reason for this modification has to be investigated. This may result in general changes to the workflow definition.

We will cover this subject in the next following section that deals about the workflow monitoring. A graphical tool is used for the analysis of such modifications to the predefined workflow: the *GroupFlow* Monitor.

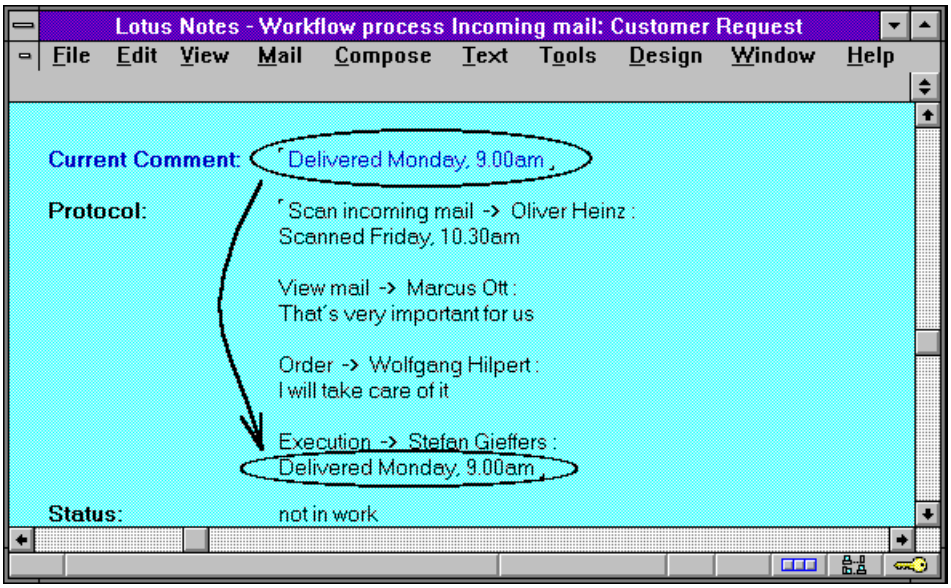

***Figure 4.11*: Audit trails as the workflow protocol**

4.4 Workflow Monitoring (*ex post* analysis)

A comprehensive workflow management environment such as the *GroupFlow* architecture requires monitoring and control of the process for a sound business operation. An advanced visualization of the actual process development is especially important in an open and flexible environment where various opportunities for ad hoc modification or refinement of pre-defined processes within the above mentioned *workflow continuum* [Nastansky/Hilpert 1994] are given and encouraged. If the workflow system provides a high level of adaptability and responsiveness to *real life* conditions extensive care must be taken not to cause chaotic circumstances. Even though this is rather an organizational than a technical issue, there may be certain features of a workflow management environment that support the task of an organizational engineer by large when structuring the processes. A workflow monitor is to provide the required means for an *ex post* analysis of the actual business processes and thus supports organizational development.

The *GroupFlow Monitor* visualizes the *actual workflow* of the workgroup. Based upon the entries both in the target workflow application databases and the protocol of the workflow routing status as recorded in the tracking database, an *ex post* as well as a concurrent analysis of the business process can be performed. The most interesting feature of a *Monitor* is the possibility to compare the entries of the actual workflow instances with the underlying *common* or *planned* workflow type specifications in the workflow repository. This analysis tool is to point out the differences and visualize quantitative relations.

Most of the considerations about the workflow analysis perspectives apply to both the ex ante and the *ex post* analysis. The verification or simulation of modeled workflow types prior to their activation consists basically of the same views or dimensions of the business process as the analysis of active or completed actual workflows. The *ex ante* analysis will be provided by the *GroupFlow Simulator* to enable interactive testing of the workflow design.

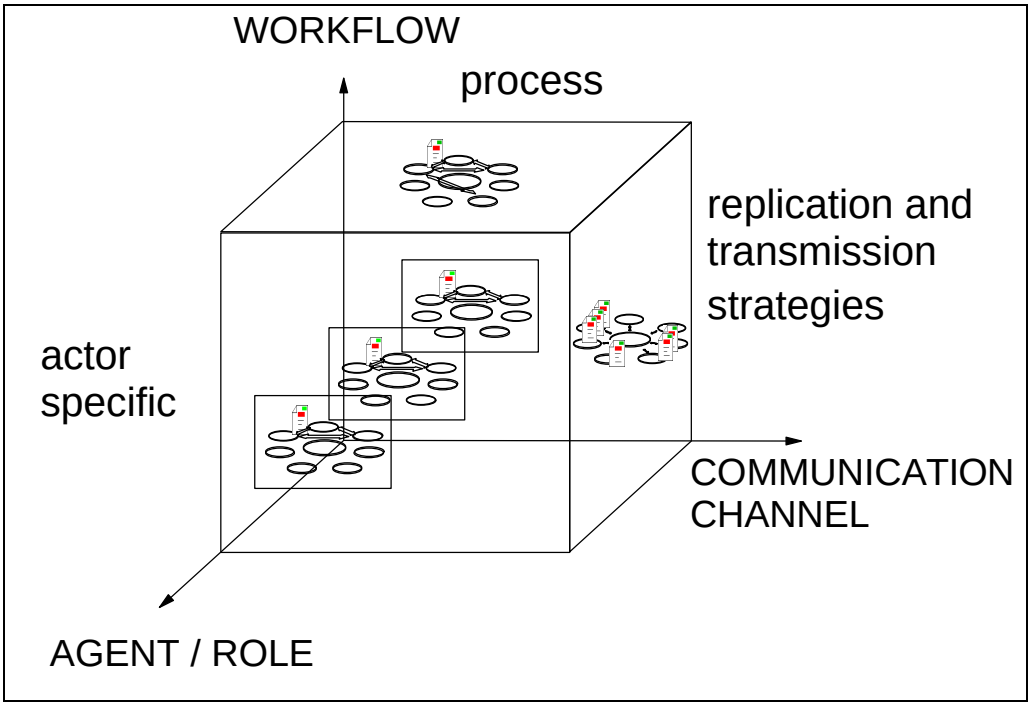

Figure 4.12 **: Dimensions of the workflow analysis**

The *GroupFlow* Monitor features the concurrent and the *ex post* analysis of the workflow. In order to monitor the progress of the processes we propose three major dimensions to focus on:

- workflows

- actors
- time.

When keeping each of these dimensions fixed one after another producing three different surfaces of cut, we obtain various views of interest: The first is to provide a "snap shot" of all activities of the team at a certain point of time, e.g. currently. The second is to show a process specific continuous routing flow of objects between team members - or actors - over a longer period of time. The third shows how particular actors are involved within the overall business process.

A. *Snap shot*: static workflow analysis

For this view all document objects of all business processes that are currently be worked on - which means that they have been created but not archived yet - are included. They are only shown in their actual status at present. This view helps estimating the current workload and the workload balance between team members. This *static* workflow analysis serves to distribute the tasks appropriately between the several workgroup members according to their individual capabilities.

B. Process specific: dynamic workflow analysis

By determining a particular workflow or a specific cluster of workflows all document objects that have been dealt with during a process can be followed in their development over the time. This view shows the currently active workflows, preferred routing paths and also relationships between activities such as time restrictions. It should give hints about recurrent ad hoc changes to predefined workflow specifications. If such an exception occurs very often during a specific task completion the redesign of the underlying common workflow type may be indicated.

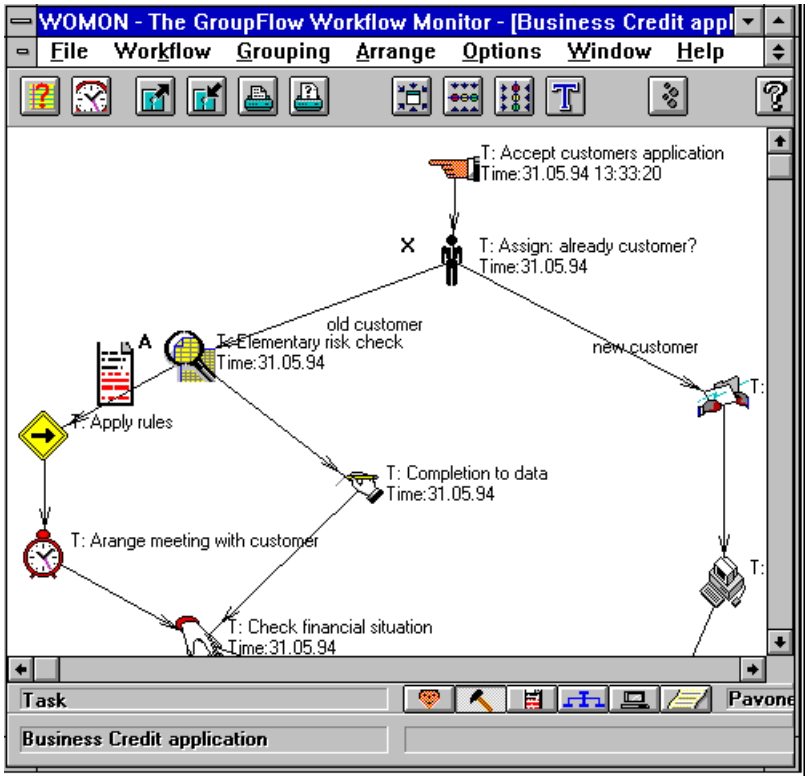

***Figure 4.13*: Monitoring of workflows**

Also, it may be of interest not only to detect possible inconsistencies and potential bottlenecks but also to find a measure for the evaluation of specified workflow types. This could encompass the number of crossings over departmental borders or escalation and de-escalation within the organizational hierarchy. A preferable workflow type would minimize the number of those both *vertical*

and *horizontal* crossings in order to decrease the potential for delays caused by organizational interface problems and thus promote *lean* business processes.

C. Actor specific: involvement in various processes

The third dimension of analysis investigates the activities of single actors or workgroups on document objects of various workflows. It can be shown, how individual actors are involved in different workflows within the team over a longer period of time. Particularly the consequences of long lasting absentee or reassignment on new tasks of single actors can be analyzed. This is specifically interesting for short-term, non planned actor replacements.

In order to visualize the business processes a basic workflow monitor can be developed both utilizing Lotus Notes internal front ends such as forms and list views. For a more refined and powerful analysis a graphical representation of the team work is required as it is shown in the figures above. The visualization of quantities of document objects, simply by total volume numbers or proportioned by the task complexity, is included in all of the three analysis perspectives mentioned above.

The Monitor provides an aggregated overview of the hierarchy of process elements. This is important to assist management or administration personnel keeping track of the status of active processes.

There may be many more views of interest potentially to be supported by the *GroupFlow* Monitor. Still, one should bear on mind that the overall aim of the *GroupFlow* environment is to offer a platform with a well balanced sophistication between the productive platform for the actual workflow routing and end user interaction during business processes performance on one hand and the graphical tool set for workflow design and analysis on the other hand.

5. References

[Bracchi/Pernici 1984]
Bracchi, Giampo;Pernici, Barbara: The Design Requirements of Office Systems. In: ACM Transaction on Office Inform. Systems, Vol.2, No. 2, 1984, S. 151ff.

[Clark/O'Donnell 1991]
Clark, Bruce; O'Donnell, Sharon: Computer supported co-operative work. In: Br Telecom Technol J, Vol 9 No 1 January 1991, 1991, pp. 47-55.

[Davenport 1993]
Davenport, Thomas H.: Process Innovation, Reengineering Work through Information Technology. Harvard Business School Press, Boston, 1993.

[Desai 1991]
Desai, S.M.: Unification of Underlying Concepts in Different Office Models. In: DataBase Winter/Spring, 1991, pp. 38-45.

[Dutton 1993]
Dutton, James E.: Commonsense Approach to Process Modeling. In: IEEE Software, 4., 10, 1993, pp. 56-64.

[Hammer/Champy 1993]
Hammer, Michael; Champy, James: Reengineering the Corporation, A Manifesto For Business Revolution. Harper Business, New York, NY,USA, 1993.

[Harrington 1991]
Harrington, H. James: Business Process Improvement: The Breaktrough Strategy for Total Quality, Productivity, and Competitiveness, McGraw-Hill, New York, 1991.

[Ishii/Ohkubo 1991]
Ishii, H.; 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, S. 184-191.

[Jari/Nils/Risto/Markku/Reijo 1993]
Jari, Alasuvanto; Nils, Enlund; Risto, Mäntylä; Markku, Riipinen; Reijo, Sulonen: Workflow in Newspaper Prepress Departments. In: Groupware 93 - Europe, 1993, pp. 279-295.

[Keen 1991]
Keen, Peter G.W.: Shaping the Future: Business Design through Information Technology, Harvard Business School Press, Boston, 1991.

[Kubota/Ishii 1989]
Kubota, Kazunari; Ishii, Hiroshi: Office Procedure Knowledge Base for Organizational Office Work Support. In: Office Information Systems: The Design Process, Pernici, Verrijn-Stuart, A.A. (Ed.), Elsevier Science Publishers B.V., North-Holland, 1989, pp. 55-72.

[Lochovsky/Hogg/Weiser/Mendelzon 1988]
Lochovsky, F.H.; Hogg, J.S.; Weiser, S.P.; Mendelzon, A.O.: OTM: Specifyung Office Tasks. In: ACM Computing Survey, 1988, S. 46-54.

[Marshak 1992]
Marshak, Ronni T.: Requirements for Workflow. In: Office Computing Report, Seybold, Boston, USA, Vol. 15, No. 3, 1992, 1992, S. 3-16.

[Medina-Mora/Winograd/Flores/Flores 1992]
Medina-Mora,Raúl; Winograd,Terry; Flores,Rodrigo; Flores,Fernando: The Action Workflow Approach to Workflow Management Technology. In: CSCW 92 Proceedings, 1992.

[Nastansky/Hilpert 1993]
Nastansky, Ludwig; Hilpert, Wolfgang: Critical Success Factors for Workflow Management as a Key Component in Banking Services. In: Proceedings, WKWI Conference Nürnberg, Germany, October 7./8., 1993.

[Nastansky/Hilpert 1994]
Nastansky, Ludwig; Hilpert, Wolfgang: The *GroupFlow* System: A Scalable Approach to Workflow Management between Cooperation and Automation, Working Paper, University of Paderborn, February, 1994. accepted for the IFIP Congess '94 in Hamburg, Germany.

[Newman 1979]
Newman, William: Office Modells and Office Systems Design. In: Integrated Office sytems, N. Naffah, Ed., North-Holland, Amsterdam, October, 1979, S. 3-10.

[Ott 1994]
Ott, M.: Conceptional design and implementation of graphical workflow modeling editor in the context of distributed groupware databases, master thesis, University of Paderborn, May 1994.

[Scherr 1993]
Scherr, A.L.: A new Approach to Business Processes. In: IBM Systems Journal, 32, 1, 1993, pp. 80-98.

[Tueni/Jianzhong/Fares 1988]
Tueni, Michael; Jianzhong, Li; Fares, Pascal: AMS: A knowledge-based approach to tasks representation, organization and coordination. In: ACM, 1988, S. 78-87.