# Flexible Support for Business Processes: Extending Cooperative Hypermedia with Process Support

**Jörg M. Haake** and **Weigang Wang**
GMD - German National Research Center for Information Technology
IPSI - Integrated Publication and Information Systems Institute
Dolivostraße 15, D-64293 Darmstadt, Germany
{haake, wwang}@darmstadt.gmd.de

**ABSTRACT**
In this paper, we present a cooperative hypermedia based process support system focusing on flexible business processes. An analysis of the communication, coordination and cooperation requirements of business processes reveals a gap in current computer support. We propose to address these requirements by extending a cooperative hypermedia system with process support. The resulting system, called CHIPS, uses hypermedia based activity spaces to model the structural, relational, and computational semantics of both individual tasks and processes. Application examples demonstrate that the CHIPS system retains the intuitive usability of hypertext and can support a wide range of business processes.

**Keywords**
Hypermedia, cooperative work, activity space, flexible workflow

**INTRODUCTION**
Business processes have been a focus for the development of computer support for a long time. Over the past few years, more emphasis has been placed on the inherent collaborative nature of business processes carried out by organizations composed of people [2, 20]. Due to the globalization of markets and increasing competition, there is a need for more efficient (i.e., fast and cost effective) business processes. The development and adoption of BPR (Business Process Reengineering) [9] supports this claim. In addition, new organizational forms are being developed (such as virtual organizations or virtual companies) that are aiming at more flexible and faster response to dynamically changing markets by temporarily combining specialized teams from different partner companies.

In general, three aspects of collaboration can be distinguished which also apply to collaborative business processes: communication, coordination, and cooperation. Communication refers to the basic ability to exchange information of any required form in the collaboration process between the parties involved. Coordination focuses on scheduling and ordering tasks performed by these parties whereas cooperation focuses on actually performing the business process collaboratively. One could consider coordination to deal with the meta-level of the collaboration process (i.e., the planning and scheduling of the business process) and cooperation to refer to the cooperative execution of the business process (e.g., asynchronously or synchronously). While the latter requires coordination of collaborators' activities, too, we subsume it under the cooperation aspect since it deals with coordinating rather basic (micro-level) activities, e.g., by using social protocols among people.

An analysis of business processes with respect to their coordination requirements shows that they vary with respect to their fluidity. Here, fluidity refers to the amount of unstructured or emergent process structure present in the process. On the one extreme, there are strictly defined processes (e.g., travel application) that can be captured once and for all in a fixed process definition, i.e., fluidity is low. On the other extreme, there are complex, intellectually demanding processes that involve ill-defined parts (e.g., for situation-dependent or highly complex sub tasks). The latter cannot be completely captured in a fixed process definition beforehand since the ill-defined parts usually involve changes or refinement of the process structure at execution time. Such processes have a higher fluidity than strict processes and a great deal of flexibility is required to fill the gaps and address the situation at hand.

In addition, processes are rarely executed by single workers but usually are tackled by a team of co-workers responsible for different tasks. Thus, the team needs to address both the coordination of their actual collaboration (i.e., the planning and scheduling of the business process at hand) and the carrying out of the actual work (i.e., the cooperative execution of the activities of which the process consists).

Current support for business processes usually focuses on either supporting the coordination aspect of generally asynchronously executed business processes by individuals (e.g., by a Workflow Management System or WFMS) or on providing communication and cooperation support for groups dealing with more fluid, ill-defined processes (e.g., by e-mail, shared workspaces). Although there are some approaches in the WFMS area that support flexible and adaptive Workflow (WF) and other approaches in the groupware area that combine some basic WF functionality with asynchronous cooperation on shared documents, there is still no integrated approach that provides extensive support for all three aspects of collaborative business processes. The integration of coordination support for gradually evolving process structure (from initially ill-defined to finally more strictly defined) and the corresponding cooperation support for executing processes by cooperating groups of users are still open issues.

In this paper, a collaborative hypermedia system is presented that addresses these issues. It provides flexible support for fluid business processes that are defined and executed by groups. Its novelty is based on the integration of several approaches within the cooperative hypermedia system and the provision of functionality that addresses the communication, coordination and cooperation requirements of collaborating groups when executing business processes. The approach can be characterized by (1) the dual use of hypermedia structures to model process structure as well as the information that people manipulate when cooperatively executing the process, (2) the capability to enact or support processes by using computational semantics associated with hypermedia objects, (3) the use of constraint checking to support consistent process definition by end users, (4) the ease of switching between process definition and process execution made possible by the seamless integration of both by the dual use of hypermedia structure, and (5) the support for asynchronous as well as synchronous process definition and execution provided by the collaborative hypermedia system.

The remainder of this paper is organized as follows: Section 2 presents a problem analysis and derives requirements for computer support systems for business processes. Section 3 discusses related approaches for providing computer support for business processes, and identifies gaps with respect to current approaches and the requirements. Section 4 then introduces our approach to provide flexible support for business processes, namely by using an extended collaborative hypermedia system, and some examples of usage. Section 5 compares our approach and our prototype system to related work. Finally, section 6 presents our conclusions and discusses future work.

## PROBLEM ANALYSIS and REQUIREMENTS

In recent years, the trends towards global markets and global competition have been a major force for companies to reengineer their organizations. The main goals of such reorganizations are to foster competitiveness by increasing speed to market and by enhancing fast and flexible response to changing markets. To achieve these goals, companies have invested in new computer and communication technology as well as in more flexible and effective organization of their business processes. Thus, designing more flexible and effective business processes as well as more efficient execution of business processes by distributed teams are gaining more attention.

Since these processes are defined and executed by groups, the aspects of communication, coordination and cooperation need to be addressed by any computer support system for such processes. Specifically, from the group's perspective such systems need to provide:

- Communication support for exchanging information and accessing shared data concerning process definitions as well as documents and data that are the content of a task;
- Coordination support for the group that is applicable to cooperative process definition and scheduling;
- Cooperative execution support for working on shared tasks and enabling groups to work in different cooperative modes (from asynchronous to synchronous) and in different settings.

Since business processes vary with respect to their fluidity, a computer support system needs to provide:

- Support for different types of processes (from strictly structured to unstructured). Appropriate support for execution needs to be provided;
- Support for the integration of these different process types. As Nutt pointed out, flexible workflow needs to integrate structured and unstructured workflow [16];
- Support for explicit formulation of goals and allow users to access this information for facilitating a better understanding [4];
- Support for emerging process definition and the easy switching between definition and execution phases for accommodating emerging processes (see also the requirement of interwoven definition and execution modules in [4]).

Finally, computer support systems for flexible business processes can only succeed if they support [16]:

- existing social models in the organization through additional informal communication channels, and
- variable coordination support.

Only if all three aspects of collaborative business processes and the requirements referring to flexibility are addressed in a computer support system we can expect successful support for collaborative business processes.

## RELATED WORK

In this section, related work is discussed in terms of the approach to support business processes. For this purpose, the three aspects of collaborative business processes (communication, coordination and cooperation) are discussed in that order.

*Communication* support has been a prime objective of many early software developments. Bilateral text-based communication by e-mail and chat programs have been complemented recently by multimedia e-mail, Internet telephony and video conferences. Multilateral communication is supported, e.g., by bulletin boards and list servers. One can argue that there is a lot of support for simple communication. However, due to their focus on communication, these systems do not provide much more support for groups trying to define and execute business processes.

*Coordination* support is the primary area of Workflow Management Systems. Workflow management systems (e.g., production WFMS and administrative WFMS [20]) support the user-controlled definition of processes by individual users. No explicit support for the cooperative definition of processes (i.e., schema design) is provided. The strength of current WFMS is the system-controlled enactment (or execution) of processes which enables features like consistency, status and history tracking, and automatic scheduling. However, most WFMS in the past only supported the definition of strict or formal processes, thus having problems in supporting ill-defined or incomplete processes. Early on, researchers phrased criticism at these workflow management systems. Lucy Suchman argued that in some cases, a prescriptive workflow system would have simply complicated a group's work [24]. Robinson and Bannon questioned the applicability of work models in the real world, since work models can never truly represent an organization's work because of ontological drift and other factors [18].

Recently, many efforts have been made to make workflow systems more flexible to changes. Various exception handling methods allow users to change the process definition of a running process in InConcert, GroupFlow, TeamWARE Flow, Visual WorkFlo, and CSE/WorkFlow [14]. The CSE/WorkFlow system integrates predefined workflow with ad-hoc workflow by enforcing those parts in a process definition which are predefined, and by turning to the end user or administrator at run-time, whenever a next step or series of steps is not predefined. The adaptive business process approach of WebFlow uses incremental process specification and changes the process definition at run-time, while Zippin uses declarative process modeling with rule-based scripts [16]. Automatic process definition inference is provided in TeamWARE Flow [14]. Some workflow systems, such as Process WEAVER (Cap Gemini Innovation), have also incorporated hypertext features for supporting information access. Usually they allow embedded links in process definition to point to background information. However, besides all this progress with respect to more flexible, incremental process definition and execution, WFMS usually still focus on the individual user instead of providing cooperation support for groups of users defining and executing collaborative business processes. Also, there is still the traditional separation between coordination (using the process definition) and the actual cooperation on the shared information or documents (that are usually maintained outside of the WFMS). Current WFMS do not address the cooperation aspect very well nor do they explicitly deal with the communication aspect.

Another example of coordination support is the COORDINATOR [13]. This system supports establishment and enforcement of commitments between users based on speech acts. By focusing on negotiation and commitments the COORDINATOR can support coordinating flexible processes. However, it does not provide support for the actual cooperation or other kinds of communication (other than e-mail).

The *cooperation* aspect is the prime issue of cooperative applications or groupware. Here, one can find many tools and approaches for supporting groups to share artifacts and to work on them either asynchronously or synchronously. Some recent systems have begun to integrate informal communication support (via desktop video conferences or Internet telephony) with shared editing environments (see, e.g., the ProShare and PictureTel products, or the Mbone tool suite). Group calendars offer scheduling support, shared document repositories allow access to shared documents, shared editors allow concurrent editing of documents, cooperative electronic meeting support systems permit distributed and co-located meetings, etc. There are also systems like Lotus Notes or Microsoft's Exchange that support replication and asynchronous editing of shared documents and that provide some basic (script-based) workflow functionality.

These groupware tools can (more or less) be used for the planning and definition of informal processes (e.g., in the form of a shared workplan document, or a shared calendar tool) as well as for cooperative execution of a task or process (e.g., cooperative writing of a document, or having a distributed decision meeting). However, they usually do not control the user's activities, but provide mechanisms for synchronizing cooperative behavior. Instead, users decide on what and how they work (individually or together). Thus, groupware tools support process execution only in a very limited way. Also, there has been a lack of facilities for process definition. These groupware tools therefore address the coordination aspect differently and especially neglect the support for automatic execution and monitoring of processes. In addition, groupware tools usually do not support analysis of and constraints on process definitions.

In summary, one can state that the different approaches discussed above provide a wide range of support but usually emphasize one aspect or the other. To accommodate the flexibility required by collaborative business processes an integrated approach addressing all three aspects would be beneficial. Our approach to this problem is presented in the next section.

## OUR APPROACH: the CHIPS SYSTEM

Our approach to tackle the above mentioned deficits is to use collaborative hypermedia as the backbone of a flexible business process support system. The hypertext concept distinguishes information components (nodes) that are connected by relationships (links) [15]. Using links, linear as well as nonlinear network structures can be formed. In addition to the basic notion of nodes and links, one can introduce types of nodes and types of links. These types can be used to capture application or domain semantics, e.g., by determining allowed types of nodes as link end points of specific types of links. In addition to simple nodes, many hypertext systems introduced composite nodes (composites) that contain other nodes and links. Thus, they can be used to form aggregated subnets within the hyperdocument which lead to the possibility of layered graphs or network. A hyperdocument denotes the collection of all nodes and links that constitutes the document (i.e., it can be modeled by a top level composite). Hypermedia extends the hypertext concept by allowing any kind of multimedia information to be the content of nodes. Collaborative hypermedia now adds to the hypermedia concept the possibility of sharing a hypermedia workspace among many people. In a collaborative hypermedia system the hypermedia document can play two roles: 1) to provide a representation of the content and subject matter, and 2) to also provide a medium for cooperation and coordination in cooperative work [21].

Based on the above identified requirements, we developed CHIPS (Cooperative Hypermedia Integrated with Process Support) as an extension of the COWFISH system [25]. CHIPS introduces flexible business process modeling and execution capabilities into a collaborative hypermedia system. Its novelty is based on (1) the integration of several approaches within the cooperative hypermedia system and (2) the provision of functionality that addresses the communication, coordination and cooperation requirements of dealing with business processes. Activity space and flexible hypertext are two major concepts underlying the approach. The resulting system can be characterized as an activity space based process support system. In the next three subsections, first the concepts of activity space and flexible hypertext are briefly introduced, and a meta-model of activity spaces as well as a set of tools for creating and using activity spaces are described. Then, the CHIPS system is introduced and its application of activity spaces and flexible hypertext to process support are described. Finally four application examples are given to demonstrate a flavor of such a flexible business process support system from users' points of view.

### Flexible Hypertext-Based Activity Spaces

A flexible hypertext system is a hypertext system which supports the co-existence and transformation of information structures in different degrees of formality, i.e., from very informal and unrestricted representations to very formal representations adhering to explicit rules prescribed by the system [7]. An activity space provides task-specific typed hypertext objects and operations [22, 25]. Its definition can be compared to a schema of a database system or a document type definition (DTD) in SGML [17] in that it determines the structures that can be instantiated in an activity space instance. The activity space instance provides support for creating information spaces consisting of instances of allowed object types, limiting its organization to allowed structures, and potentially offering task-specific operations. Our activity spaces use a labeled graph representation, which can be seen as a semantic network. This network has structural, relational, and computational semantics. Different activity spaces can be created by typing the hypertext objects and adjusting constraints in these three semantic dimensions.

The COWFISH system [25] is built on our experience with the SEPIA [23] and the DOLPHIN [22] systems. It extends the SEPIA activity space concept into a meta-model for defining cooperative hypermedia-based activity spaces for a wide range of tasks. The elements of the meta-model are nodes, links, node content pages, and other media objects. In this meta-model, an activity space is represented as a nested node structure (i.e., a composite node). Each node has a content page. The type of an activity space is determined by its root page type. The substructures (subactivity spaces) of an activity space are defined by the page types of the nodes at each level of the nested node structure (see Figure 1). In a structure consisting of nested nodes, an existing page type can be used recursively when the node contents at different levels are of the same type. A 'page' is also an interface metaphor for presenting an activity space. It is not only a 2-dimensional drawing area for planar semantic nets that can be presented directly as a network of typed nodes and links, but also a space for embedded semantic nets whose node contents can not be seen without opening the nodes. In addition to nodes and links many other media types, such as text and hand-drawings, can be included on a page.

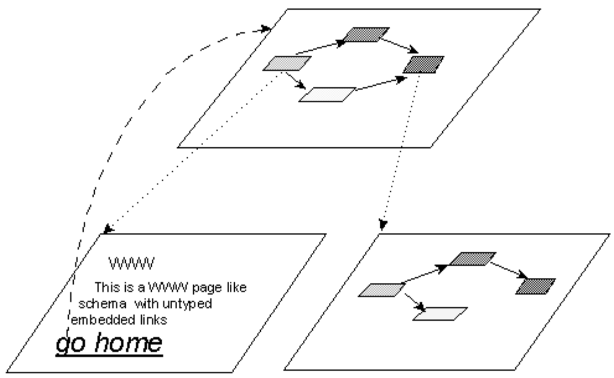


**Figure 1: Logical Structure of Activity Spaces**

This meta-model can describe an activity space from three semantic dimensions:

- *structural semantics* describe the graph structure and constraints of a hypermedia structure. Under this dimension, links and nodes are classified into two categories: organizational and referential. Organizational links and nested nodes are constrained to form a directed acyclic graph, while referential links and nested nodes can form a graph of any type;
- *relational semantics* describe the relationships and constraints between typed hypermedia objects. Under this dimension, links and nodes are further classified into domain-specific types. For instance, the relational constraint <Position, Answer, Issue> specifies that an Answer type link can be used to

connect from a Position type node to an Issue type node;

- *computational semantics* describe the operations, constraints, and triggering conditions attached to hypermedia objects. Examples of such computational semantics are an invocation function attached to a node for activating external applications, and forwarding time dependency along precedence links of a PERT Chart [26].

Based on the meta-model, a set of tools has been developed to help users define, manage, use, and create activity spaces. In order to support ordinary users to define new activity spaces, we developed an *example-based definition tool*, with which users can create activity space schemata by creating an example document. Thus, users do not need to learn any formal syntax for defining a new activity space. This approach also provides means to help users employ a predefined activity space: in the second tool, an *activity space browser*, the example used for its definition can be used as online guidance for new users, and the type information can be used by the system to maintain the consistency of the model and provide intelligent context-sensitive aid for users of the activity space instances. To support emergent structures, a tool called the *flexible space* is provided, which allows users to develop a pattern gradually using unconstrained hypermedia. To support the reuse of existing information, the tools allow users to copy, modify and integrate existing activity spaces. Also an *activity space launcher* is provided to help users manage existing activity spaces. More detailed description on these tools can be found in [25].

### Activity-Space-Based Process Support

CHIPS uses the COWFISH concepts introduced in the previous section to provide a semantic net based hypertext network as a graphical representation language for task and process definition. In CHIPS, both tasks and processes are represented as hypermedia-based activity spaces. Different task and process patterns are defined as different activity space schemata. The process enactment is implemented by attaching task and process computational semantics to hypermedia objects. In the following, we describe the process support extension provided by the CHIPS system using the terminology defined in the Workflow Management Coalition (WfMC) reference model [8]. In addition, we also use some concepts defined in our activity space meta-model in the previous sections.

The process meta-model provides the basic components for process definition which can then be executed. These three aspects are described in the sequel.

#### *The Process Meta-model*

The activity space meta-model includes general structural and computational semantic presentation constructs for modeling a process. The basic elements of the process meta-model are described in the following:

- *Task nodes* describe logical steps within a process. Since they are represented with hypertext nodes, each task node has an associated page as its content, and this page can also serve as *containers* of the task's input and output data;
- *Process links* represent precedence relationship between task nodes. A process link is often associated with a data flow from its source task node to its target task node;
- *Transition conditions* are logical expressions associated with process links. They are represented as conditions attached to process links;
- *Pre- and post- conditions* are logical expressions which may be evaluated to see whether a task can be started or completed. They are represented as conditions attached to task nodes;
- *Actors* are *users*, *roles*, *organization units*, or *agents* that perform the work represented in task nodes. Actors are identified by an attribute of a task node;

*Ordinary hypermedia links, nodes, pages, and media objects* are *application data*, which are application specific and are usually not directly accessible by a workflow system. However, in this model they are represented in a unified hypertext model, and therefore accessible by our system.

The distinction of task nodes and process links from ordinary hypermedia nodes and links is reflected in the computational semantics attached to them. Application data can also be stored outside of the hypermedia system and accessed through *workflow relevant data* kept in task node attributes. The *workflow control data* - the data that relate to state, time, and state transition are represented as attributes of task nodes and process links. Since this paper focuses on hypermedia extensions for process modeling and execution, the modeling details on user, role, organization and agents are not included.

#### *Support for Process Definition*

In this process meta-model, a *process* is represented as a set of task nodes connected by process links. More specifically, a process is a rooted directed acyclic graph consisting of (potentially nested) task nodes and process links among task nodes. The acyclic constraint is a hypertext invariant for the containment relationship in composite nodes, and the constraint makes many analysis and monitoring methods, such as PERT/CPM (Program Evaluation and Review Technology/Critical Path Method) [26], easier to implement. This definition is in conformance with the definition of an activity space, so we call it a process space. In the process meta-model, ordinary nodes, links, and media objects can also be included in a process space.

In CHIPS, a process definition consists of two parts:

- a process space schema, which also includes schemata of its subspaces, which may be subprocesses, tasks, or other spaces without workflow computational semantics. A schema can be defined by using either the *example-based definition tool* or the *flexible space* tool; and
- a process space template, which can be the example created for defining the process space schema, or an

instance of the process space schema instantiated in an *activity space browser*.

A process space schema is constrained by the process meta-model. Many process space schemata can be created from the meta-model. Process space templates are constrained by the process space schemata, and many process templates can be instantiated from a process space schema. Both a new instance created from a schema and an initialized copy of a template are instances of the same schema. Their difference lies in that an initialized copy of a template contains a properly initialized hypertext structure, so as to relieve people of the burden to create the structure again by creating nodes and crafting links. Actually, in CHIPS any activity space instance can serve as a template to be duplicated with a copy-as-template function.

In one incarnation of the process meta-model which is used as a default in our system, five task node types are identified. They are 'simple task nodes', 'process task nodes', 'automated task nodes', 'iteration simple task nodes', and 'iteration process task nodes'. A 'simple task node' represents an atomic task. A 'process task node' models a subprocess. An 'automated task node' represents a task to be performed by a computer program. An 'iteration task node' represents an iteration task or process. The and-, or- joints and the and-, or-splits of a process can be represented with a combination of the pre- and post- condition of task nodes and the transition conditions of process links. For instance, an and-joint can be represented by a pre-condition which requires all transition conditions of the incoming process links to be true. The acyclic constraint on process links and the organizational structure of nested nodes does not exclude the possibility to model loops in a process. Loops in a process can be represented with iteration task nodes.

As each task node has its own content page where application data can be stored and accessed, the process links serve as both control flow connectors and data flow connectors. For a simple task node, an automated task node, or an iteration simple task node, the node serves as its own input and output container. For a process task node or an iteration process task node, the input container is the start node of the process contained in the process task node and the output container is the end node of the process contained in the process task node. Based on their associated data flow patterns, currently process links are classified into five types: 'precede', 'share', 'copy', 'transfer', and 'integrate'. A 'precede' process link is a pure control flow connector, there is no data flow associated with it. A 'share' process link specifies that the output container of its source task node shares the same content (page) with the input container of its target task node. A 'copy' process link specifies that the input container of its target task node will contain a content copy from the output container of its source task node. A 'transfer' process link specifies that the content of the output container of its source task node will be transferred to the input container of its target task node according to predefined mapping rules. An 'integrate' process link specifies that the content of the output container of its source task node will be merged into the input container of its target task node. These labels for task node types and process link types are not necessarily the labels for the semantic types of the nodes and links. Nodes and links in a specific process space can have their own semantic type labels that are better for representing the domain information model of the process space. The process computational semantics, such as those reflected in task node types, process link types, and conditions on task nodes and process links, are assigned to those typed nodes and links through menu operations in the *example-based definition tool* or the *flexible space*. After the assignment, they become part of the properties of the typed nodes and links to be readily used for creating process space instances in an *activity space browser.*

The system uses a collaborative authoring metaphor for process definition. Users create process structure as hypermedia structures within the activity space browsers of the hypermedia authoring environment. Because the process spaces are constrained by their schemata, the structural and the relational consistency of the process is maintained by the system. For project management purposes, in addition to the workflow semantics, the PERT semantics have also been incorporated into the process network and their time dependency is triggered and updated dynamically. Also, CPM analysis can be performed upon the process network for business process refinement. A process execution animation has been implemented as a 'guided tour' of a process space in a test mode. A 'guided tour' is a hypertext concept that derived from the 'path' concept. A 'path' is an ordered traversal of some links in a hypertext [27]. A 'guided tour' is one of the forms of paths [11]. 'Paths' provide structural clues for user navigation in hypertext information space. 'Guided tours' provide system-controlled navigation. 'Paths' and 'guided tours' have been used to organize hypertext networks for intelligible presentation; while in this work they are extended to present the results of process simulation. Such process execution animation can help users to check correctness and completeness of a process definition.

#### *Support for Process Execution*

The process execution mechanism of the system is also implemented in its hypermedia environment. Similar to InConcert [12], in this system a process is a data object (a task node) not a program or script. The information on process state and conditions are distributed over task node instances, and state transitions are triggered and forwarded step by step along process links. This object-oriented feature allows a process instance to be modified by users (subject to their access rights) during the course of execution for exception handling or for flexible adaptation to new situations. Thus, the system supports both process definition and process execution within one activity space.

To enact a process, first a process instance is copied from a predefined process template. Then the process can be started by a user. When a process is started, the start task node is enabled and if it is an automated task node, its state will change to 'active' and the specified program is

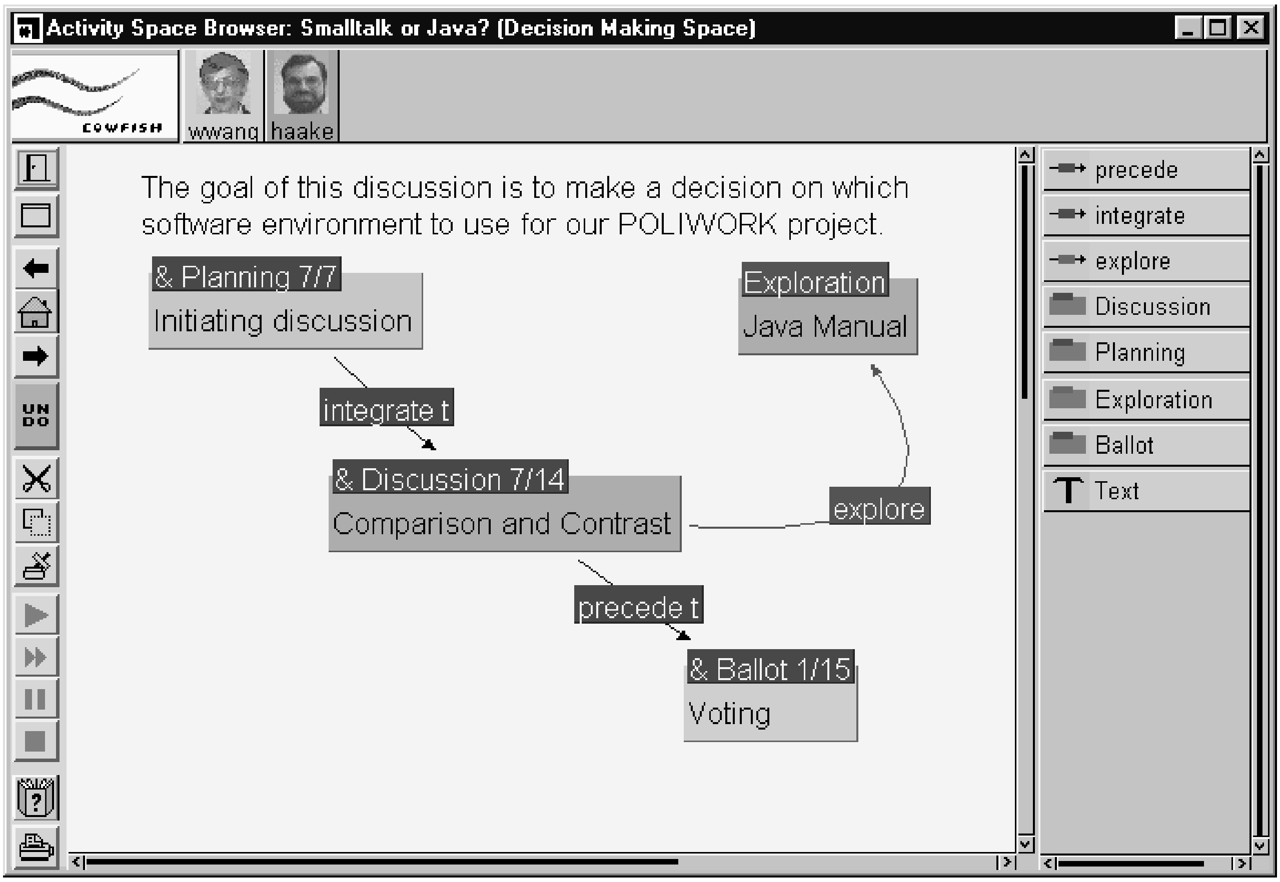


**Figure 2: A process space**

called. Otherwise, its human actors will be notified to activate the task node (the system will send an email or open the node in an activity space browser on the actors' desktop if they are working in the system). A human actor can activate a task by opening the task node, setting its state to 'active', and performing the task in the activity space. Other actors of the task can enter the activity space at any time in its 'active' duration. When the task is finished, the actor(s) can change the state to 'finished', which in turn triggers the evaluation of its post-condition. If the post-condition is met, all its outgoing process links and their transition conditions are evaluated. Otherwise the task has to be rescheduled for execution. In this case, if a deadline is reached, the task node will change its color to red and a message will be sent to notify the actors of the task and the coordinator of the process. A process link is selected if its transition condition evaluates to be true. The target tasks of the selected process links are then evaluated. If their pre-conditions are met, they send notifications to their actors. A process is considered terminated if all of its end tasks are 'completed'.

The state, time attribute values, pre-conditions of task nodes and the transition conditions of process links are visible on the user interface (See Figure 2). The state information is represented as colors of task nodes (for instance, gray for inactive, yellow for enabled, green for active, red for signaling exceptions, pink for suspended, and brown for completed). The planned task 'duration/end data' can be seen at the right-hand side of a node type label. These numbers indicate that the node is a task node. The pre-condition logic of a task node can be seen at the left-hand side of the node type label of a task node ('&' for an AND-joint and '|' for an OR-joint). The result of the transition condition of a process link can be seen at the right-hand side of the link type label ('t' for true and 'f' for false). More detailed task information can be seen in a property sheet activated through a menu operation. In addition to the special guided tour for process animation, a normal guided tour can be activated with a menu operation to inspect the progress of a process. The process space (including all of its nested nodes) can be linearized and printed on paper with a 'composite node print' menu operation.

**Examples of Use**

CHIPS is built on our COAST toolkit [19] and offers many groupware features for communication and cooperation. For instance, a task or process space can be launched in one of the private, loosely-coupled or tightly-coupled working modes, and users or applications may switch among these modes within a collaborative working process. In these differently coupled working modes, common object aspects, such as current editing page and the position of scrollbar, are coupled (shared and combined) in different ways. For instance, in a loosely-coupled cooperation mode, the navigation action of a user does not affect the browsers of other cooperating users. However, if users work in a tightly-coupled cooperation mode, all users would follow the navigational actions of other group members in the same

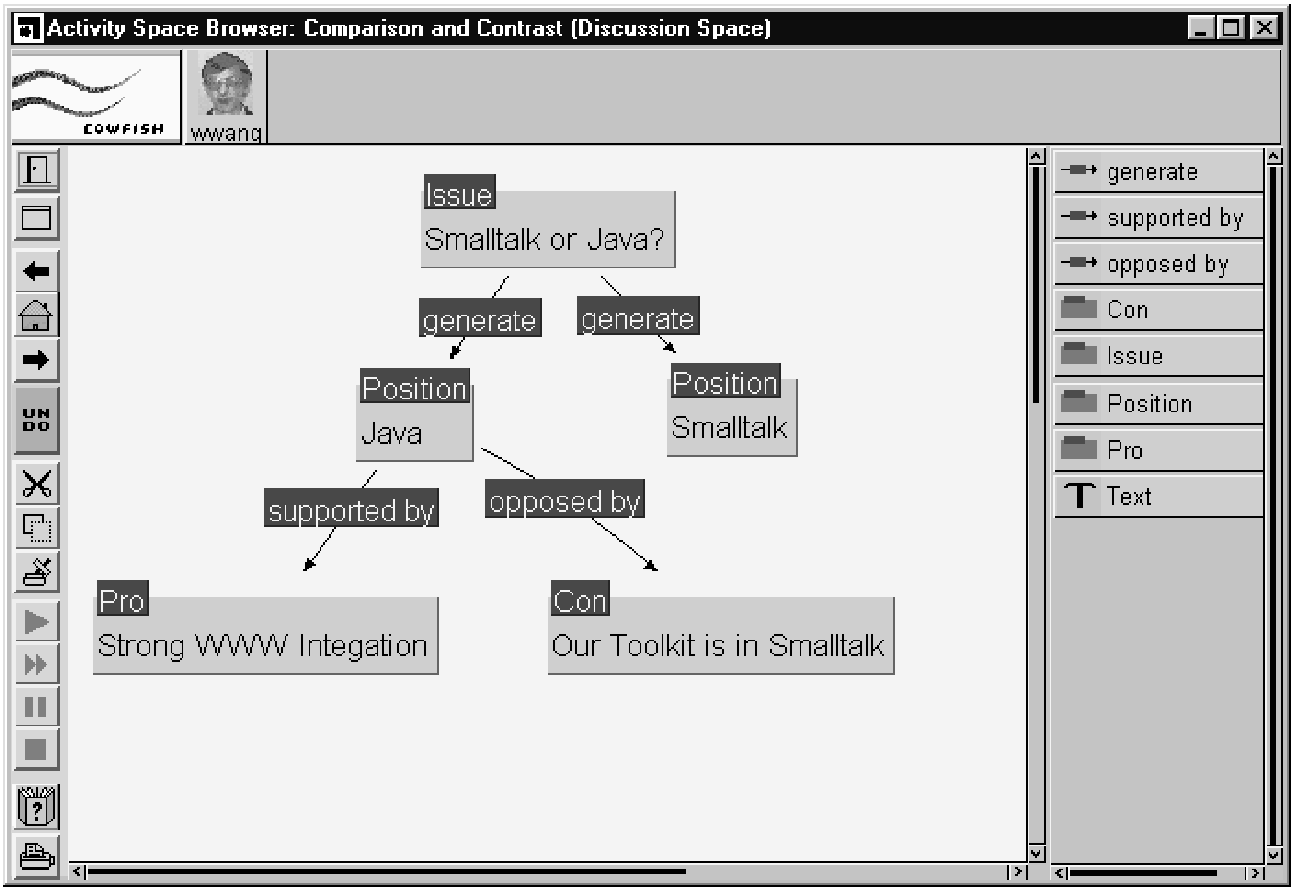


**Figure 3: A task subspace of the process space**

session. This section intends to reveal a portion of its flexibility and collaboration support from users' points of view. The screen layout of our activity space tools as illustrated in Figure 2 has five major areas. At the top is a title bar which gives the name of the tool, the name of the current node, and the type of the current page. Under the title bar is a system logo and a list of users (represented by pictures) currently working together on the same page. The largest area in the middle is the current page. To the left of the page is a palette of tools for navigation, editing, and task related triggers (i.e., buttons for 'start a task', 'fast animation', 'pause and continue', and 'finish a task'), and to the right of the page is a palette of hypertext objects that are allowed in the page. We now describe four examples of using CHIPS in scenarios beginning with high fluidity and progressing towards decreasing fluidity.

For a situated task that has no prescribed process structure, such as brainstorming in a face-to-face meeting, we can define an unstructured public space with common tools at its users disposal. For instance, for the brainstorming task, we can define a ''White board'' space to be launched in a tightly coupled mode, which makes it a public space for all meeting participants in real time. This activity space provides ordinary text, scribbles, untyped nodes and links as its default types of hypermedia objects. With our example-based definition tool, such a space can be defined by simply activating a node creation menu operation, providing a new page type (named as ''white board'') for the node in a page type assignment dialog box, then, opening the new node, and creating a line of text, a scribble, two untyped nodes and an untyped link between the two nodes. After such a ''white board'' space is defined by example, it will appear in the page type assignment dialog box popped up in subsequent node creation operations for users to choose from. When the participants work in such a space, they can type and sketch directly on the ''White board'', or pick-and-drop the provided hypermedia objects from a type palette in the browser onto the ''White board''. They can also make use of the informal communication channels in a face-to-face meeting to coordinate their collaboration.

For a task that is suitable for manual coordination, such as editing a book from a set of book chapter contributions, we can define a shared activity space with an information structure suitable for the task. In this way users can benefit from the groupware features of the system, but carry out the task in normal ways they feel comfortable with. For the book editing task, a ''book space'' can be defined whose structure consists of hierarchically nested nodes, with a ''book'' type page at the top, which contains nodes with ''chapter'' type pages, and ''chapter'' pages in turn contain nodes with ''section'' type pages. Again, with the example-based definition tool, the ''book space'' definition is very easy. All the users have to do is to create a small book sample with only one chapter and one section. After the book space is defined, it can be instantiated in the space launcher, or in a new node associated with a page of ‘‘book’’ type. The book editors can coordinate their work manually through a shared book space and through ordinary communication means. For instance, they can send emails or make phone calls to negotiate a working schedule and to assign chapters to responsible editors. They can also document their agreed working plan in textual and graphical descriptions on the cover page of the shared book space and use the plan as both a communication medium and an informal coordination means for their cooperation. After the chapters are loaded into the book space and are properly edited, the editors can invite others to comment or to check if any cross-references are needed among the chapters of the book.

For some tasks, such as asynchronous distributed co-decision making, a suggested working procedure might be helpful for initiating the task in a planned way, while in the meantime allowing its participants to deviate from the procedure or change it on the fly. For instance, for the co-decision task, we can define three subtasks: ‘initiating discussion’, ‘comparison and contrast’, and ‘voting’. For the ‘initiating discussion’ and ‘comparison and contrast’ tasks, we use an IBIS-like argumentation schema [1] as their information structure. The schema for the ‘initiating discussion’ space includes Issue and Position type nodes and ‘generate’ type links. The relational constraint of the link allows it be used only from an Issue type node to a Position type node. The ‘comparison and contrast’ space extends the schema of the ‘initiating discussion’ to include node type ‘Pro’ and ‘Con’, and link type ‘supported by’ and ‘opposed by’. A ‘Position’ can be ‘supported by’ a ‘Pro’ and ‘opposed by’ a ‘Con’ (See Figure 3). The ‘voting’ task node is an automatic task node, which, when activated, invokes an external voting program. The process definition uses ‘integrate’ type process link from the ‘initiating discussion’ task node to the ‘comparison and contrast’ task node, and uses a ‘precede’ process link from the ‘comparison and contrast’ task node to the ‘voting’ task node (See Figure 2). The first two tasks take one week each, and the last takes one day. This process definition together with its task definitions not only defines an argumentation structure to be created by a group, but also specifies how and when to create what parts of the structure by whom. For instance, the first week focuses on raising issues and expressing positions, and the second week is mainly for debating before voting is performed. During the execution of the process, the participants of the discussion may change the duration of some tasks or the task and process structures. Also, the coordinator of the process may take over the control of the process, and invite people to discuss some issues in a tightly-coupled session.

For a task with a mandatory schedule, such as the collaborative preparation of a project deliverable for a large project that involves many partners, a strict working procedure is necessary. Given such a deliverable preparation task, an instance of the book space defined in the above example can be used as a shared workspace. However, in this case, in addition to the task space for the document, a process space is created collaboratively to coordinate the contributions from the project partners so as to meet the mandatory schedule and the deadline of the project. Because the process definition is created collaboratively by all the project partners, it may reflect the consensus and commitments of the partners, and as a consequence, it may be followed better by all the partners. In this process definition, the process starts with an outline of the deliverable which involves all partners and takes six months. Then each of the partners of the project starts to work on the part that relates to their work package for one year. When the year ends, every partner has to deliver their part to other partners for comments and make revisions according to the comments. The deadline for this is after half a year. Finally the manager of the project has to go over the whole document and prepare an executive summary for the deliverable in one month, and send it to the funding agency. In this case, the detailed plan for each step is managed informally by each partner, so all steps can be represented as simple task nodes. Since the document is stored in a shared activity space, there is no need to have data flow between the task nodes. Therefore, only the ‘precede’ links are used. The content page of the start task node (the deliverable outline) and the end node (the executive summary) include some text description of their task and an embedded node pointing to the root page of the deliverable. The content page of each of the writing task nodes and the comment-and-revision task nodes includes some text description of its corresponding task and an embedded node pointing to its corresponding part (chapter). One week before each deadline, a reminder will be sent to its actors automatically. And if the deadline of a task is past before the task is completed, a notification will be sent to its actors and the manager of the project.

## COMPARISON to RELATED WORK

When describing the latest research and future work of workflow systems, Sheth points out that one direction for workflow systems is to have integral support for collaboration, not just coordination [20]. Nutt argues that workflow systems should be made more flexible, and the goal of leading edge systems is to support high logical immersion into the environment (i.e., to support existing social models in an organization through informal communication), with high computation support for specific tasks, and variable coordination support. Such systems should be able to support both situated work and workflow (or some customized combination of the two) on a case-by-case basis [16]. Ellis argues that successful workflow systems must extend the process models to include declarational specifications about the goal of a task (or a process) and descriptive specifications on how to perform a task (or a process) alongside operational ones [4]. These remarks are highly in agreement with our identified requirements for flexible business support systems and our identified need for integrated support for all the aspects of communication, cooperation, and coordination that are partly supported one way or another in today’s groupware systems and workflow systems.

Recently, there have been many research and development efforts on merging workflow and groupware [3]. The basic approach taken by the workflow community is to integrate some communication and cooperation models or features into workflow models or systems, such as the speech act theory underlying ActionFlow [13] and the recent extension to the ICN model in Chautaugua [3], or to integrate existing workflow systems and groupware systems for capturing informal process representations or handling exceptions, such as the recent work in the Exotica project to integrate FlowMark and Notes [14]. Our approach to the integration is in just the opposite direction. We start with a cooperative hypermedia system with many communication and cooperation features and integrate process support capabilities (i.e., more formal coordination support) into the system. Despite the difference, in our effort we have paid close attention to the lessons learned, experiences gained, and suggestions for future systems from those leading workflow researchers and practitioners.

The CHIPS system can be characterized as an activity space based process support system with integrated communication, cooperation, and coordination features. It provides support that bridges the gap between formal and informal mechanisms for coordination and cooperation. The possibility of process definition in a collaborative way enhances group members' motivation and group cohesion. In the following, we compare our system with related work by looking at each of its communication, coordination and cooperation aspects.

**Communication Aspect**

The system presented in this paper can make use of informal *communication* channels alongside automatic coordination. This is achieved by including informal specifications in ordinary hypertext alongside operational ones represented in hypertext with process semantics, and by incorporating cooperation support features implemented in our COAST toolkit and our COWFISH and DOLPHIN systems, such as shared activity space browsers with multiple tele-pointers, pen-based direct manipulation to hypertext structures, and Audio/Video connections between participating workstations or sites. A frequent means of informal communication in our system is the use of the shared hyperdocument. Within the shared hyperdocument (containing the process definition and the relevant information resources) users can also express goals and add explanations to processes and tasks, e.g., as annotations.

**Coordination Aspect**

The CHIPS system can provide a moderate and end-user-controlled amount of automatic *coordination* for a wide range of business processes, from fluid situated work to strictly coordinated processes. Especially a mixture of business processes whose degree of automatic coordination is decided on a case by case basis can be supported. This is enabled by our activity space structure that consists of nested nodes with various page types (schemata of sub activity spaces). Because of its object-oriented execution control, the system allows manual tasks to include subtasks that are automatic tasks or processes. This is a combination not available in current workflow systems. Another distinct flexibility of the system is that ordinary hypertext structure can be turned into process structure, and process structure can be changed into ordinary hypertext structure.

The dual interpretation of hypermedia structure as process structure and as the information that people manipulate when cooperatively executing the process makes it easier to switch between process definition and process execution. As a consequence, exceptions and adaptation to changing processes become easier to handle. Although there are process support systems that integrate some simple hypertext features, such as embedded links for information accessing in Process WEAVER and many Web-based workflow systems, these systems have not provided more advanced hypertext features such as composite nodes and system aided navigation support (e.g., guided tours) as we have. The Trellis project used Petri-Net based hypertext to model CSCW coordination protocols [5]. With respect to integrating computational semantics into hypertext, CHIPS is similar to Trellis. However, CHIPS uses semantic nets for representation and presentation of its hypertext structure, which is more intuitive for ordinary users. Also none of the other systems support the transformation between ordinary hypertext structure and hypertext structure with process semantics.

The process space schema and its example-based definition method offers another distinctive feature of the CHIPS system. Most other workflow systems have only one graphical editor (or one hard-coded schema) for process definition. In CHIPS, there could be many process schemata for different process patterns or for processes with different task structures. These schemata can be incrementally changed to meet the changing needs of their process space instances. A small change in a process instance (or a process template) only has a small local effect on that particular process instance. And a change in a process schema (i.e., a change made in the defining example of the process space) will affect all its process space instances to allow newly added constructs or to remove some old constructs, but the existing data in the process space instances will remain unchanged. The example-based definition method allows ordinary users to define new task or process spaces. The flexible space tool can support users to capture informal process descriptions and then gradually transform them into a more formal process structure. Moreover the system can make use of schemata to maintain the structural and relational consistency of a process structure. It can provide intelligent aid in creating task and process structures by providing allowable choices to prevent violations, and issuing warnings (and aborting the violating operation) for violations that can not be foreseen before an action is taken.

**Cooperation Aspect**

CHIPS *cooperatio*n features, such as shared information space, group awareness features, and differently coupled cooperation modes provide a good basis for collaborative process definition and execution, which is not available in most workflow systems. This allows groups to cooperate asynchronously as well as synchronously when defining and executing business processes.

Activity spaces can represent task-specific information structures (i.e., domain information models) and provide task-specific operations on the structures. This provides the system with a relatively high level of computational support for various tasks. The CSRS system [10] has a data modeling language for software review tasks which is similar in some ways to our activity space modeling. But CSRS is a system specially developed for software reviews: it does not offer a general meta-model and intuitive tools that allow ordinary users to define activity spaces for a wide range of tasks.

In addition, CHIPS supports smooth transitions between process definition and process execution by switching process semantics on and off as requested by the users. It also supports emergent process structure by allowing flexible combinations of automated and manual tasks and the incremental change or addition of more structure during process execution by adding task nodes and process links to the hypertext process representation.

## CONCLUSIONS and FUTURE WORK

As a conclusion of this work we can say that business processes need first, integrated support for communication, coordination and cooperation, and second, means for flexible process modeling and execution by groups. The extension of cooperative hypermedia systems with flexible process support is a promising approach for such an integration. The use of hypertext as both a graphical representation of processes and the content people manipulate in a cooperative work process bridges the gap between informal and formal mechanisms for coordination. Support for cooperative process definition, which is not available in current workflow systems, can enhance group members' motivation and group cohesion. Using hypermedia types to support variable degrees of automated coordination is one option for current workflow development directions. Our approach can be considered as a complementary approach to many of the current efforts that focus on extending WFMS with collaboration technology. Perhaps, cooperative hypermedia can serve as a bridge here!

Compared with workflow systems, many important issues such as coordination of large-scale processes and advanced transaction models, are not addressed by our system. Our short-term goal is to demonstrate the possibility and usefulness of integrating process support into collaborative hypermedia systems, rather than to create a new full-fledged workflow system.

Among other topics we are especially interested in are the problems of orientation within large process spaces and dealing with the consequences of changing process schemata. For the first problem, we investigate the usefulness of group-aware views and filters on hypermedia spaces to display the current state of the process as well as potentials for overview and conflict visualization. In addition to passive filtering, we also investigate the use of notification mechanisms for providing active support. For the second problem, we plan to develop means for detecting inconsistencies and for retaining as much of the process history as possible when a process schema change invalidates a part of a process already running. Here, we can make use of transaction logs and undo-redo mechanisms provided by our COAST toolkit.

CHIPS is a prototype system of our ongoing research project. We are continually modifying and improving it. The system has been tested in our research group and has been demonstrated frequently to our guests. It is currently used in experiments in the group. One of our next tasks is to gain more usage experience in larger user communities.

## ACKNOWLEDGMENTS

The authors want to especially thank Anja Haake for her constructive comments on this paper. We also thank Ajit Bapat, Jennifer Beck-Wilson, Christian Schuckmann, Ralf Steinmetz and Daniel Tietze for reading the paper and providing feedback.

## REFERENCES

1. Conklin, J., and Begeman, M. gIBIS: a tool for all reasons. *Journal of American Society of Information Science*, 40, 3 (1989), 200-213.
2. Denning, P. J. The fifteenth level. In *Proceedings of ACM SIGMETRICS Conference* on Measurement & Modeling of Computer systems (May 1994).
3. Ellis, C., and Maltzahn, C. Chautaugua: a flexible workflow system. In *Proceedings of the 30th HICSS* (Jan. 1997).
4. Ellis, C., and Nutt, G. Multi-dimensional workflow. In *Proceedings of the IDPT'96* (1996).
5. Furuta, R., and Stotts, D. Dynamic hyperdocuments: Authoring replaces programming. *Communication of the ACM* (Aug. 1995).
6. Grasso, A., Meunier, J., Pagani, D., and Pareschi, R. Distributed coordination and workflow on the world wide web. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 6, 1 (1997), 1-26.
7. Haake, J., Neuwirth, C., and Streitz, N. Coexistence and transformation of informal and formal structures: Requirements for more flexible hypermedia systems. In *Proceedings of ACM ECHT'94* (1994), pp. 1-12.
8. Hollingsworth, D. *The Workflow Reference Model*. Workflow Management Coalition, TC00-1003, Dec. 1994.
9. Johansson, H. J., McHugh, P., Pendelbury, A. J., and III, W. A. W. *Business Process Reengineering* (1993), John Wiley.
10. Johnson, P. M., and Tjahjono, D. Improving software quality through computer supported collaborative review. In *Proceedings of ECSCW'93* (Sept. 1993).
11. Marshall, C. C., and Irish, P. M. Guided tours and on-line presentations: How authors make existing hypertext intelligible for readers. In *ACM Hypertext'89 Proceedings* (1989), pp. 15-26.

12. McCarthy, D., and Sarin, S. Workflow and transactions in InConcert. *IEEE Data Engineering*, 16,2 (June 1993), 53-56.

13. Medina-Mora, R., Winograd, T., Flores, R., and Flores, F. The action workflow approach to workflow management technology. In *Proceedings of ACM CSCW'92* (1992), pp. 281-288.

14. Mohan, C. Tutorial: State of the art in workflow management research and products. In a tutorial presented at ACM SIGMOD International Conference on Management of Data (June 1996).

15. Nielsen, J. *Hypertext and Hypermedia*. Academic Press, Inc., San Diego, 1990.

16. Nutt, G. The evolution towards flexible workflow systems. *Distributed Systems Engineering Journal*, 3, 4 (Dec. 1996), 276-294.

17. Organization, I. S. Information Processing -- Text and Office Systems -- Standard. *Generalized Markup Language SGML*. ISO 8879, Geneva, October 15 1989.

18. Robinson, M., and Bannon, L. Questioning representations. In *Proceedings of ECSCW'91* (1991), Kluwer Academic Publications, pp. 219-233.

19. Schuckmann, C., Schümmer, J., Kirchner, L., and Haake, J. Designing object-oriented synchronours groupware with COAST. In *Proceedings of ACM CSCW'96* (Nov. 1996), pp. 30-38.

20. Sheth, A. State-of-the-art and future directions. In *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems* (May 1996).

21. Streitz, N. Putting objects to work: Hypermedia as the subject matter and the medium for computer-supported cooperative work.\newblock *In Lecture Notes in Computer Science: Object-Oriented Programming* (1994), M. Tokoro and R. Pareschi, Eds., Springer, pp. 183-193.

22. Streitz, N., Geissler, J., Haake, J., and Hol, J. DOLPHIN: Integrated meeting support across local and remote desktop environments and liveboards. In *Proceedings of ACM CSCW'94* (October 1994), pp. 345-358.

23. Streitz, N., Haake, J., Hannemann, J., Lemke, A., Schuler, W., Schutt, H., and Thüring, M.. SEPIA: a cooperative hypermedia authoring environment. In *Proceedings of ACM Hypertext'92 (*1992), pp. 11-22.

24. Suchman, L. Office procedure as practical action: Models of work and system design. *ACM TOIS*, 1, 4 (Oct. 1983), 320-328.

25. Wang, W., and Haake, J. Supporting user-defined activity spaces. In *Proceedings of ACM Hypertext'97* (Apr. 1997), pp. 112-123.

26. Wiest, J. D., and Levy, F. K. *A Management Guide to PERT/CPM*. Prentice-hall, New Jersey, 1969.

27. Zellweger, P. T. Scripted documents: a hypermedia path mechanism. In *ACM Hypertext'89 Proceedings* (1989), pp. 1-14.