Putting Words to Work: Integrating Conversation with Workflow Modeling

Peter J. Kammer and David W. McDonald
Information and Computer Science
University of California, Irvine
Irvine, CA 92697-3425, USA
{pkammer, dmcdonal}@ics.uci.edu

Technical Report 99-30

ABSTRACT
The ad-hoc nature of work practice is a challenge for developers of systems that attempt to model and support the enaction of work processes. Development in workflow modeling systems has taken a number of approaches to addressing the issue of exceptions, or deviations from the "ideal" process. We suggest that one significant element, that of informal communication between process participants, has been largely overlooked by these efforts. Not only is social interaction a key component of work practice, it serves a particularly important role in addressing unusual situations, solving problems, and coordinating the implementation of those solutions. This paper explores the design space of integrating tools for conversation with workflow systems.[1]

Keywords
Dynamic/adaptive workflow, work processes, informal communication, chat, exceptions, workflow management systems, tool integration, Endeavors

INTRODUCTION

The process of work is never quite what it seems. People commonly rationalize their own work to make it seem cleaner than it really is. In reality work is often a set of social exchanges with explicit and implicit contingencies that is overloaded with assumptions about each individual's role in the work process. Communication, formal and informal, and associated negotiation that clarifies expectations is the social lubricant that keeps work processes flowing.

Systems that support and augment work and work flow in organizations must be capable of representing a rational, clear, work process while still supporting the messy nature of real work. Part of this task is supporting the conversations that coordinate people in shared activity, giving them awareness of the activity of others, and allowing them to react collectively to problems as they arise. Further necessitating this kind of support are the increasingly numerous instances of widely distributed work groups and virtual enterprises composed of workers who never share an office space [16].

The integration of informal conversational tools with workflow systems is an open problem. This work examines the social and informational aspects of conversational tools and the exceptional nature of work processes. This is the design space that must be covered by the integration of workflow systems with conversational tools. Further, we explore the ramifications of combining these technologies. Finally, we demonstrate some aspects through a simple prototype integration.

THE EXCEPTIONAL NATURE OF WORK PROCESSES

The term, "exception" in a work process model refers to events that deviate from the normal flow of the work procedure. This reflects the perspective that work, at least if correctly organized and specified, normally proceeds in fairly routine consistent ways. The concept of an exception suggests an unusual event, a random occurrence that diverges from this norm. This view, the "random-event perspective" as it is described by Strong and Miller [33], is commonly assumed by managers and researchers. This point of view is not born out, however, by studies of actual work processes.

On the contrary, exceptions are the rule in practical work. Standard processes serve as "post-hoc rationalizations" of the ad-hoc problem solving activities necessary to complete work tasks [21]. In her study, Suchman [34] found that standard procedures serve as a directive for "what things should come to, not necessarily how they should arrive there," reflecting the goal of the work, but not the process of achieving it.

Case studies by others, Bowers [8] and Gasser [19] for example, also reflect the routine nature of exceptions. Gasser

1. Effort sponsored by the Defense Advanced Research Projects Agency, and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-97-2-0021. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory or the U.S. Government.

describes this as the "ubiquity of anomaly" and indicates that some exceptions occur with such frequency that approaches to handling the exceptions are themselves routinized.

This lack of adherence to fixed work procedures suggests the need not only to model work processes in the ideal, but also to facilitate the coordination of work as it actually takes place. Systems should support interactions necessary to complete practical tasks, to construct ad-hoc solutions to problems as they arise, and to obtain results directed toward the work goals embodied in procedures.

The Need to Support Work with Communication

Communication is one element that clearly affects the ability of workers to complete practical tasks, identify and resolve problems, and produce results. Most work is an inherently social process. The range of approaches and applications of communication tools demonstrates the important role communication plays in the work process (we will discuss some of these past efforts briefly in a later section.) Kraut et al. [26] document the role and significance of formal and informal communication in both social interaction and production. Other authors note the need for supporting or modeling communication to support specific work applications, for example, Posner and Baecker [29] in collaborative writing, and Curtis et al., [11] in the software development process. Suchman's [34] case study shows workers making extensive use of informal communication to coordinate their ad-hoc work solutions. Not only does communication support the smooth progress of work activities, it is a fundamental component of problem solving and the composition of these ad-hoc work procedures that Suchman describes.

An Open Design Space

While extensive work has been done both in modeling work processes and supporting work with communication, not much attention has been paid to the representation and coordination of informal communication as part of the work process model. This omission limits the capability of workflow systems to model and support the activities that actually occur in practical work.

To avoid confusion, it is important to distinguish the intent of this work from that of the language/action perspective of workflow [38]. The language/action approach characterizes work process using a formal conversational model and has been the subject of some discussion and debate [35], [37]. A number of workflow systems have leveraged off of this model or variations of it (e. g. [3], [28]) as well as integrated communication tools to support it [4]. Instead, our emphasis is on supporting the informal "real" conversation that takes place between process participants and how it may be used to enrich the work model and its execution.

The remainder of this paper is structured in the following way. First we will describe a number of key aspects of both work process modeling and the use of conversational tools in work contexts. We conceptually layout the design space for integrating communication tools with workflow systems. Some of the design issues we cover include process support infrastructure of workflow systems and the modeling of communication artifacts as components of the work process. We describe the implications of combining these technologies, suggesting a substantial benefit in this kind of integration. Lastly, we will illustrate some of these concepts through a simple prototype integration of ESPRESSO, a tool supporting informal communication, with the ENDEAVORS workflow support system.

MODELING WORK PROCESSES

Workflow technologies adopt a variety of approaches to supporting process specification and enaction, differing significantly in the degree of specification they support as well as the methodology employed [6].

For the purposes of our discussion, our interest lies in systems that support the modeling of common work process constructs and their relationships. Fundamental workflow entities include:

Activities represent tasks within a work process, steps in completing the process being modeled. These may be automated, carried out by a single individual, or the responsibility of a group of people who must be coordinated to complete the task. Examples of activities include creating a document, holding a meeting, and compiling program source code into executable form,

Artifacts represent items created and/or manipulated within the work process. These could include such things as information forms, minutes of meetings, transcripts of conversations, and program source code.

Resources are facilities used within the process. Examples are meeting rooms, hardware, and software.

Agents participate in executing activities and manipulating artifacts. Agents could be automated (e.g. components of the workflow system and integrated tools) or represent people, groups, or roles within the process.

Relationships are connections between work entities. There are a broad span of relationships that may be modeled. A relationship might tie an activity to agents participating in it, indicate artifacts needed and where they come from, or establish the order in which activities take place.

These components are used to model work processes, which should then support enaction and integration with external tools, manipulating them as part of the process.

Workflow Based Approaches to Managing Exceptions

Systems supporting work processes have adopted a number of techniques to address the issue of exceptions. These fall into two general categories. One class of approaches attempts to correct deviations from the process model when they occur, the other seeks to provide appropriate levels of specification and sufficient flexibility to prevent exceptions from occurring. We briefly describe several approaches here; for a more complete examination of these issues see [23].

Recovering from Process Deviations

When actual work deviates from the process specification, one approach is to attempt to handle the deviation and bring work, as much as possible, back into a consistent state with the model process. This approach adopts a philosophy similar to exception handling in programming languages. Execution is redirected to an exception handling routine which attempts, most likely with human assistance, to recover from the exception and resume normal execution of the work process at an appropriate point. This may require the rollback and/or repetition of some completed operations. For a discussion of some exception handling approaches (along with extensive taxonomy and discussion of exceptions), see [31].

Evolving Process Models

Sometimes exceptions are indicative of systematic failures in the process model. While a deviation from the model in one instance of its use may be corrected for that instance, repeated incidents suggest the need to evolve the model to prevent similar problems reoccurring. Future instances of the process then reflect the change.

Dynamic Change and Composition of Process Models

Recognizing that process enactment does not consistently follow the same path, some systems address the problem of exceptions with strong support for dynamism in process modeling and execution. Dynamism describes the ability of the user to dynamically specify and modify the process definition as the process is executed. This permits instances of the guiding process to be tailored to individual circumstance. In particularly flexible systems, this also allows "just-in-time" processes to be constructed on-the-fly as needed, based on available information. A number of authors have addressed dynamism as a fundamental component of exception-tolerant workflow systems [5], [7], [10], [14].

Flexible Enactment Models

In contrast to attempts to correct deviations or perfect models of work processes, an alternative approach is to provide only loose specification of the sequencing of user tasks. By specifying processes using loose constraints, users are presented with a number of possible tasks to choose from, essentially constructing the process model as they enact it. FREEFLOW is one system that demonstrates this approach [13].

CONVERSATION TOOLS

Communication styles vary widely. We briefly describe several dimensions that distinguish conversational forms supported by communication tools. A particular tool may support different approaches to varying degrees. While our focus is on computer-mediated communications, these distinctions have analogy in communication artifacts typically found in organizations and we note these where appropriate. The characteristics, as well as the degree to which participants may identify and control them, strongly influence the character of the communication.

Our classification bears similarities to that used in [26] to describe differences between formal and informal communication. The emphasis in [26] leads the authors to include elements of scheduling and conversation content as well. Our focus is on describing the capabilities of communication tools, leading us to also include characteristics of persistence and synchronization. For a more extensive discussion of communication taxonomies, see [22].

One-to-one/one-to-many

Conversation may take place among two participants or may be broadly distributed to a group of people, some of which may merely spectate and not contribute to the actual conversation. A memo or an e-mail directed to a single individual will be interpreted differently, especially in terms of the recipient's responsibilities, than an item broadcast to a wide list.

Synchronous/asynchronous

Conversations may take place synchronously, in "real time," across one uninterrupted period of time with the opportunity for back-and-forth dialog, or asynchronously, where there is significant lapse between sending a message and obtaining replies. Conventional "chat" tools typically provide synchronous conversation, while bulletin boards or newsgroups are generally asynchronous. While many systems tend to support one or the other, several authors suggest advantages to combining the two approaches [15], [29], [32].

Persistent/transitory

A communication is persistent if it is stored for possible later examination. Transitory conversations are lost shortly after the communication has taken place (for example, once text scrolls out of a window, or audio has been heard). Newsgroups or e-mail, if archived, become persistent resources, whereas informal conversation, such as with chat tools or video/audio conferences is usually transitory.

The disposition of communication artifacts in organizations is often a matter of explicit policy. Such policy may dictate, for example, that memos be kept for a certain length of time, or even dictate that they be destroyed. Often an artifact, such as a form, will specifically indicate for how long a copy must be retained.

Inclusive/Exclusive

Informal and formal conversations both have flow and control characteristics. In a formal situation control may be vested in a single person. However, in informal situations control and turn taking in the conversation is distributed and shared among the participants. The characteristic of a given conversation that determines who may participate is the inclusiveness or exclusiveness of the conversation.

Moderated discussion lists and news groups are examples of formalized control over the inclusiveness of a conversation. In a moderated forum, messages are first sent to a person who approves or rejects dissemination to the full group. Theoretically, this prevents redundant or inappropriate material being sent to the entire list.

An analogous situation arises in organizations. Many organizations require that memos be approved by particular individuals. In this case the requirement satisfied is not so much one of avoiding irrelevant items, but rather of establishing authority and responsibility.

Filtering Conversation

Informal conversation in a public forum can produce long, quick flowing, interleaved interaction. Both interesting and uninteresting information will be intermingled in the flow of conversation with many participants. Such conversations are difficult when heard, but when presented as text they can be overwhelming. This situation is even more complicated when introducing automated notification of external events (e.g. weather reports, company notices, or workflow status) into the conversational stream.

In this difficult situation the informational impacts are relatively clear. The participants in the conversation miss key points and overhear irrelevant information. This can lead to conversational non-sequiturs and social faux-pas that can ultimately lead to the cessation of the conversation.

An effective integration of workflow systems and conversational tools will address both social and informational aspects of conversation maintenance. Sophisticated filtering techniques hold the promise of minimizing the detrimental social impacts while supporting the information needs of the participants. It is important, therefore, to provide mechanisms for filtering the conversation to highlight or alert the user to interesting or the most relevant content. Content filtering and social filtering are two techniques that apply to conversational messages [27], [20].

Content Filtering

Messages can be filtered in a number of ways based on their actual content. This involves comparing the message against a collection of key words or regular expressions, defined by the recipient, for items of interest. When a match is detected the message is highlighted or privileged in some manner.

Perhaps the simplest form of filtering is the use of channels. A mailing-list or newsgroup represents a single channel (with varying degrees of focus). The topic provides a high level guide for participants to choose channels in which the discussion is likely to be of interest. Channels do not provide a fine level of control over the material sent to the channel, so some material on a particular channel may not be of interest to all participants. In addition, rather than being based on the actual messages, filtering is based on the implicit agreement that only material relating to the channel's focus will be sent. Unless enforced by a moderator, the implicit agreement is only regulated through peer pressure.

Autonomous agents can perform content filtering when a user cannot explicitly state what is of interest. These agents are attached to a specific message stream and are trained by the user. After some period of training, the agent can be used to identify messages that are interesting.

Social Filtering

Messages can be filtered based on social characteristics surrounding the message. Techniques that filter based on the social space around a message are performing social filtering. For example, one simple means of social filtering is to select messages based on the sender of the message. Filtering based on sender can be used to distinguish messages from different individuals as well as from automated sources.

The intended recipient of a message is also an indicator of its significance. Messages that are targeted specifically to a single recipient, for example, are likely to be more significant to that recipient than messages directed at a channel or list of which the recipient is a part.

Messages may also be filtered based on who responds or who replies to a specific message. An individual message may become more interesting based on which individuals respond or reply to the message.

Communication Tools Supporting Work Activity

A number of communication tools have been used to support work activity. They vary in degree of integration with the work environment, attempts to support specific work activities, and conversational richness. We discuss briefly a number of sample approaches with increasing levels of complexity and integration with the work environment. None of the approaches discussed below have implemented more than the simplest content filtering.

Chat

Simple "chat" systems provide means for text based communication. Conversation takes place independent of work artifacts and is generally synchronous and transient. Filtering is based largely on a channel mechanism and enforced by social practice. One example of this sort of system is the ZEPHYR help instance [2], a successful use of a synchronous, channel based, chat style system to support distributed help and problem solving. BABBLE extends on the conventional chat system with mechanisms for awareness of other participants and support for a mixture of synchronous and asynchronous communication [15].

Notebooks

Electronic notebooks provide a mechanism for asynchronous persistent communication between workers, as well as a logging mechanism for work performed, problems encountered, and the like. While, in principle, they need not be much more technically complex than a electronic bulletin board, the organizational role implicitly links them to work context. Under this pretext, they can provide mechanisms for informal organizational memory and rationale capture. Kovalainen et al. [25] describe the use of a simple electronic diary in a papermill as an aid to communication and work process coordination. More complex forms of this metaphor provide for complex hyperlinked structures and the inclusion of objects from other applications in the notebook [18], [36].

Work Spaces

A number of systems use the metaphor of a work space to construct a work and communication environment. These systems combine external tools and communication infrastructure to provide an integrated view of tools, work artifacts, and other participants. These elements are presented in a coordinated view to participants in a work locale. Participants may move between areas, traveling from one work context to another. Examples of this approach include WORLDS [17] and TEAMROOMS [30].

Virtual Process Environments

Finally, Doppke et al. [12] describe the use of MUDs (Multi-User Dimensions), and the prototype system PROMO, to support the modeling of processes and associated communication. MUDs provide a (generally text based) environment for users to manipulate objects and interact with other users. In the approach adopted by Doppke activities are mapped to location, tools to objects, and process participants to players.

INCORPORATING CONVERSATION TOOLS IN WORK PROCESSES

In an earlier section we discussed the important role informal communication plays in work processes, providing both social interaction and means for coordinating work production. This role becomes more significant in the common situation where work does not follow a standardized procedure. The challenge is even greater with widely distributed teams and virtual enterprises composed of geographically dispersed participants. By strongly integrating the conversation and process modeling tools we have discussed, we can not only coordinate communication for normal work practices, but also improve coordination when the unexpected occurs.

Our research with workflow systems, suggests a number of advantages to representing communication tools as resources within the process model and including communication artifacts as elements of the process itself [7], [23]. This allows the process model not only to represent the flow of work and activities, but the interaction of participants, supporting not only the possibility of bringing people into touch when they need to be, but also of enriching the model of work as it is performed. In this section we discuss some of the specific details of how communication improves the use of work process tools, including the modeling of communication and coordination, the establishment of relationships between communication artifacts and process elements, and the provision of historical context to activities.

Communication in the Model: Bringing the Right People Together

The integration of conversation tools within the work process infrastructure allows a model of participant communication to be developed. By bringing together participants at times when they need to communicate, the workflow system can simplify the task of finding with whom a individual needs to converse to complete an activity. In combination with a model of participant roles, the system can reduce the effort required to search out coworkers with necessary expertise or responsibility.

Grounding the Conversation

Conversation may be grounded by explicitly associating elements of the work process with conversation itself [9]. The context serves as a basis for the communication. Participants can focus their discussion around a number of workflow elements, including those described in the earlier section.

Groups participating in the same activity will naturally need to discuss their task. They may coordinate work, discuss progress, or collaborate to solve problems. Communication linked to a resource (such as a meeting room or hardware) might include negotiation for its use or comments on its appropriateness for a particular task.

Artifacts provide a particularly rich source of context for communication. Often workflow centers on the routing, creation, or approval of artifacts. By connecting ongoing discussions to the artifact itself, the conversation is linked to the item being discussed and provides a method maintaining process history as it passes from activity to activity.

Each time one of these sequences of activities is enacted represents a separate instance of the processes execution. Each of these instances represents a distinct collection of tasks, a composite activity which may be discussed in, and annotated by, an associated conversation or a collection of conversations.

Conversation as a Mechanism for Building Historical Context

Associating conversations with process elements provides context for ongoing discussion. Making these discussions persistent creates a historical record for the work taking place. This improves process modeling by supporting both the user interaction and the process record.

History *provides context* for joining a conversation. Rather than being forced to infer history from current conversations, new participants may access a conversational history to obtain background, providing another mechanism for their participation in the discussion

Historical logs also support a *mixed interaction model* (synchronous and asynchronous). A message may be immediately available to be responded to or, if the recipient is not available, persistence provides for asynchronous communication as well.

These persistent notes attached to work elements provide an *annotation mechanism*. For example, a common workflow artifact is some sort of shared document, reviewed or edited by a number of participants. A persistent conversation tool can provide a mechanism for cooperative annotation of the artifact through discussion as it transitions through the work process, adding comment, describing problems, explaining unusual characteristics and so on.

Finally, maintaining history provides for *rationale capture* and feedback into the design of the work process. This builds organizational memory constructed around work activities. By examining conversation and annotations associated with workflow components qualitative insight can be acquired into the way in which work proceeds. This information can document why decisions are made and suggest ways in which to improve procedures in the future.

Giving Users Control Over Their Histories

Coupled with the benefits of recording historical conversation is the need to allow users to maintain some control over the storage of their communication. Logging

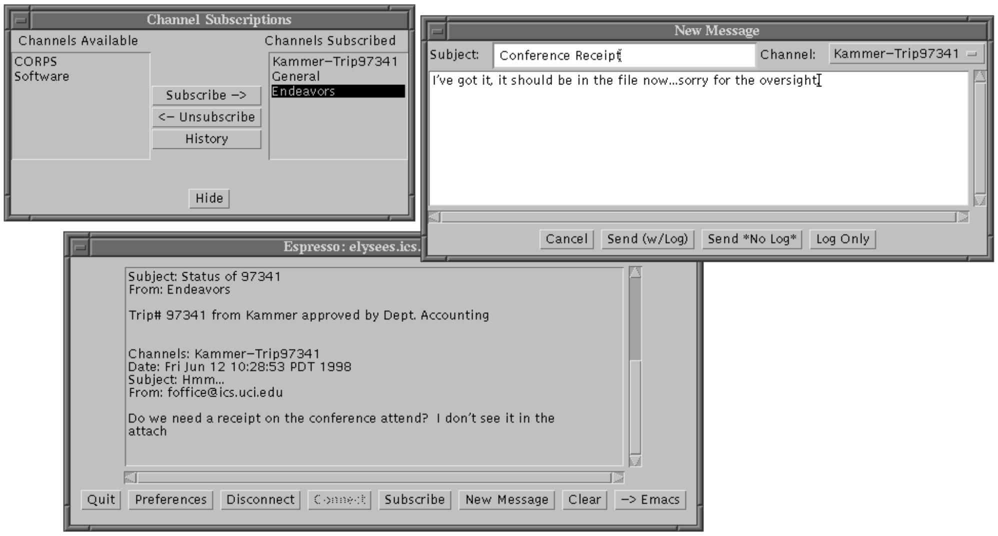

Figure 1. ESPRESSO client interface

of conversation is likely not only to make users uncomfortable with the tool, but also to collect information that is not pertinent. A number of different mechanisms may be used to distinguish persistent from transitory communication.

Controlling storage *by channel* selects what is logged based on the channel in which the communication takes place. This creates a course grain, relatively clear distinction between what will and will not be persistent. Historical logs contain all the conversation on a channel so the overall thread is likely to be coherently saved. This approach may, however, dilute the topical foci of individual channels and discourage the mixing of informal asides with more formal work related comments, as is common is face-to-face conversation.

Determining persistence on a *message by message* basis allows users to decide when they send a message (or perhaps even later) whether it will stored for later review. This allows fine grain control of what is and is not persistent. The interface must be designed carefully, however, to assure the user's confidence in their selection and determination as to whether the message being sent will be stored. Further, mixing persistent and ephemeral communication on the same channel diminishes the coherency of the record as portions of the conversation are not saved, leaving gaps in the exchange.

Communication Among Participants and System Agents

A conversation tool provides a mechanism for users to communicate with each other and for automated components of the process to communicate with the user. Work processes often involve both human centered activities (e.g. document editing) and automated activities (e.g. report generation, code compilation). Status reports from both these kinds of activity are of potential interest to users.

The workflow system itself may provide this sort of notification for events it recognizes, for example transitioning from one activity to the next. Alternatively if the system is extensible to allow programmatic description of reports from activities, the communication may be customized by the process specifier for a particular task.

Summary

Integrating conversation tools closely with the process model enhances the capabilities of both classes of application we described in the earlier sections. We have seen that there are a number of ways in which informal communication can assist participants in their practical tasks and enrich the process model. In addition the work context provided by the model can inform the conversation taking place. These capabilities provide the foundation for improved modeling and guidance of work tasks, particularly in unconventional situations where coordination between participants is essential. In the next section we will further explore these issues with a prototype integration of a communication tool and workflow system, examining the integration architecture and describing a small example of how the combination might be used.

A PROTOTYPE INTEGRATION: ESPRESSO AND ENDEAVORS

System Background

ESPRESSO

ESPRESSO is a Java based chat application derived from a larger communications toolkit, CAFECK [1]. ESPRESSO has a client-server architecture that supports multiple communication channels. A single server runs on a, potentially remote, system and interacts with multiple clients. Each client may subscribe to and thus monitor conversations on multiple channels at once. Messages are sent from the clients to the server to be broadcast on a particular channel.

The ESPRESSO client was initially designed for synchronous communication between users at their machines simultaneously. Once dialog scrolls beyond the

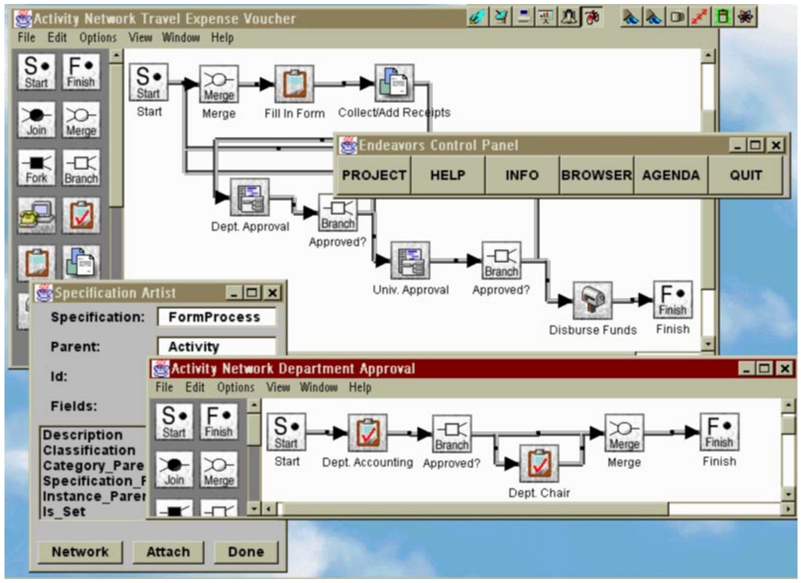

Figure 2. One view of an ENDEAVORS activity network

limit of a client's text area or a client is closed, the dialog is lost. This also means that when a new participant subscribes to a channel, they have no context for the conversation taking place. We extended ESPRESSO to provide a configurable persistence mechanism to log conversations based on channel. The server creates a log file for each channel unless the channel is excluded in the system configuration file. Upon request, the server will send this log file to a client so that a new participant may view previous activity on a channel. In addition to configuring the writing of log files based on channel, we added user controls to specify logging on a message by message basis.

Figure 1 shows the additional controls added to the ESPRESSO client to take advantage of the logs kept by the server and to give users control over whether their contributions are logged. The *New Message* dialog provides three choices for constraining how a message is sent or retained. *Send (w/Log)* sends a message and stores it to the log file. This is suitable for normal user interactions. *Send*No Log** sends the message but does not log it to a file, supporting unrelated social messages or other messages that the user does not wish to be recorded. Finally, the *Log Only* command does not send the message to the clients, but just stores it in the log file. This is appropriate for simple descriptive messages that serve as annotations, but do not need to be broadcast to other participants.

In addition to providing a means to select channels to monitor, the *Subscribe* dialog provides a mechanism for requesting the message log, or *History*, from the server. This is a way of "catching-up" on activity on a channel one has just joined. Providing additional material in the message window suggested the need for additional control. The buttons *Clear* and *Emacs->* provide convenient ways to clear out the buffer or send it to a text editing tool.

ENDEAVORS

ENDEAVORS is an open, distributed, extensible, workflow/process execution environment [7]. The system may be used on a single computer with a local file store, or distributed with remote file stores [24].

The ENDEAVORS layered object model includes a domain level view that models three primary classes of object: artifacts, activities, and resources. Artifacts are producible and consumable elements of a process, for example documents or program code. Activities represent tasks within a process, performed by people, groups of people, or computers. Resources are traditional management resources, rooms, equipment, and the like. Additional classes of objects are extended from these. Objects respond to events by invoking handlers, small sections of code that describe the object behavior. This behavior can include invoking and interacting with external tools, as well as manipulating internal constructs. ENDEAVORS is written with externally visible APIs that can be manipulated by the external tools.

Activities are associated in networks that describe process flow. ENDEAVORS provides six basic control flow primitives: start, finish, branch and merge (for decision), fork and join (for concurrency). Processes may be arranged hierarchically, with activities in one network causing the execution of an associated subnetwork. Figure 2 shows the ENDEAVORS "Network Artist" along with other components. The upper panel is a high-level process, the lower shows a subnetwork of activities associated with the top-level activity, "Dept. Approval." Also included in the figure are the main control panel and a dialog for specifying parameters for a work activity.

Conventionally, an interpreter executes the given network by traversing its control flow and sending the appropriate series of events to each activity. The interpreter maintains a reference to a modifiable execution context to retain

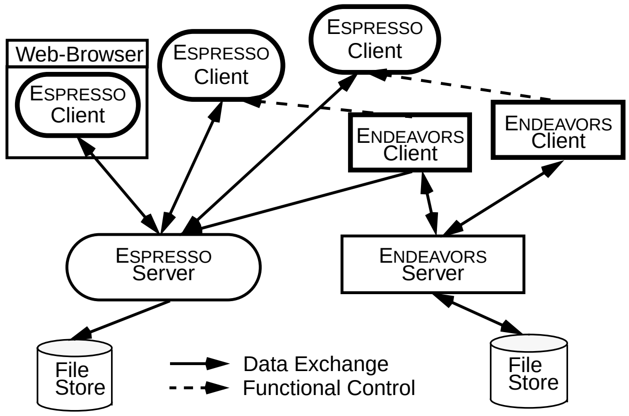

Figure 3. Architecture of ENDEAVORS / ESPRESSO integration

information specific to that invocation of the process. ENDEAVORS provides a flexible specification and interpretation model. Elements of the process may be created or changed dynamically at run time. Process interpretation may be manipulated, programmatically or through human intervention, to allow activities to be skipped, repeated or resequenced. Handlers, describing the behavior of automated activities, are loaded into the system as needed, so they may be altered dynamically as the work process is executed.

Details of the Integration

Strictly speaking, the ENDEAVORS system was not extended for this prototype. ENDEAVORS is designed as an infrastructure to support the integration of other tools through the use of handlers and its external interfaces. ESPRESSO was modeled within ENDEAVORS as two components of the *Tool* classification (*Tool* is a subclass of the *Resource* class described earlier.) One component models the connection to the ESPRESSO server. Using this component, process models may initiate a connection with the server, send messages, create new channels, and drop the connection. The other component was created to represent the ESPRESSO client within the process model. This component provides mechanisms to launch a client (as in Figure 1) on the user's desktop and subscribe it to an appropriate channel to receive messages. These capabilities are available to the process model, allowing it to create channels, send messages to be broadcast, and provide an interface to the user on the appropriate communication stream at the appropriate time.

Application of these extensions is further described in the next section. Figure 3 diagrams an integration of ENDEAVORS and ESPRESSO components. ESPRESSO clients are launched and controlled by running instances of ENDEAVORS. Clients may also be launched independently or through a web browser to participate in the conversation. The ENDEAVORS clients manipulate the ESPRESSO interfaces to provide users access to the appropriate channel when needed. They may also send status messages to the ESPRESSO server to be broadcast to its clients. The ESPRESSO server broadcasts communication to subscribed clients and logs to a file store as appropriate. The ENDEAVORS server provides remote storage to its clients and coordinates process enactment between them.

An Illustrating Scenario

Further illustrating the integration, we describe an example application of the ENDEAVORS system and how it may be improved by the integration of conversation support. An ongoing project with Endeavors has been an attempt to provide a model and support mechanism for the process of reimbursing University employees for travel expenses incurred on business for the school. This sort of forms processing is a traditional workflow application, common to many organizations. The modeling effort has been pursued through interviews and a series of prototype applications. Roughly, the procedure for receiving reimbursement is described as follows:

1. The traveller fills in a form and submits it, along with supporting documents (receipts, airline passes, and so on) to the department office.
2. The department's accounting group verifies the availability of the funding.
3. If applicable, the department's contracts and grants office verifies the appropriateness of the expense for the given funding source.
4. If validated, the request is passed to the department chair for approval.
5. From the department, the request passes to University accounting who also validates the expenses and verifies all necessary supporting documents are attached.
6. The University disbursements office issues a check or executes an electronic funds transfer to complete the reimbursement.

There are other possible branches. For example, often some or all of the expenses are prepaid. However, this scenario provides a simple illustrative description of the main process.

Support for informal communication tools and artifacts enhances this process model in a number of ways, though these elements may not be obvious in the explicit procedural model. Travelers often need assistance in satisfying the requirements and navigating the reimbursement process. A process facilitator, an experienced administrative staff member, often provides help to avoid glitches down the line. This advisory capacity is difficult to model conventionally, but is nonetheless an important element in the process. By modeling and supporting this communication role within the system, the traveler is guided in finding and communicating with a crucial process participant.

The reimbursement request artifact itself grounds the conversation relating to its approval. Individual participants working in parallel can discuss the document through conversation associated with it. This communication channel assists in collaboratively resolving problems or providing clarifications, preventing small issues from derailing the workflow. This mechanism not only provides pertinent information, but assists process participants in being aware of the related activities of their coworkers.

The associated communication provides a mechanism to make the process more transparent to the traveler and provides for ongoing involvement in the process. Through automated reports or communication from other process participants the traveler can monitor the status of the request and if a problem arises participate in the solution.

Not only can the traveler obtain greater visibility into the process, but other actors in the work process may as well. Through communication they can become aware of other participants in the work process and their roles, improving the ability to identify frequent problems and develop collaborative strategies for handling them when they occur.

CONCLUSIONS

Actual work activities rarely follow the path set out by ideal practice. Practical work is ad-hoc, requiring communication between participants to achieve the goals embodied in the standard procedures. This ad-hoc collaboration is particularly essential when unusual or unforeseen problems arise in the course of work activity and collective effort is required to bring solutions to bear.

Supporting these activities requires that workflow systems not only provide the flexibility to accommodate variation, but also the communication mechanisms to coordinate work execution. We have explored the design space integrating conversation tools with workflow modeling systems, finding a number of tangible benefits realizable from their integration.

Our implementation of a research prototype served as a promising beginning to explore the technical issues associated with this combination of tools. One requirement highlighted by the effort was the need for open interfaces to manipulate collaborative tools. While our choice of workflow system adopted this open system approach, our conversation tool did not, requiring that it be extended. Many similar systems (e.g. IRC) suffer the same shortcoming.

In our future work, we will further explore the design space of this tool combination. We envision the integration of asynchronous communication tools such as Usenet News style bulletin boards and electronic mail. As well, when more communication tools are integrated we see the need for more sophisticated content and social filtering to augment users ability to attend to communications most relevant to themselves.

ACKNOWLEDGMENTS

The authors would like to recognize the hard work and effort in the design and implementation of the Endeavors system by the Endeavors research group at UCI. Particular thanks also go to Mark Ackerman and Richard N. Taylor for their review and comments on this work.

REFERENCES

1. Ackerman, M. S., and McDonald, D. W. "Answer Garden 2: Merging Organizational Memory with Collaborative Help", in *Proc. of The Conference on Computer Supported Cooperative Work* (CSCW '96), Boston, MA, USA, Nov., 1996, ACM, New York, pp. 97-105.
2. Ackerman, M. S. and Palen, L. "The Zephyr Help Instance: Promoting Ongoing Activity in a CSCW System", in *Proc. of Human Factors in Computing Systems. Common Ground*.(CHI 96), Vancouver, BC, Canada, April, 1996, New York, NY, USA: ACM, pp.268-75.
3. Agostini, A., De Michelis, G., and Grasso, M. A. "Rethinking CSCW Systems: The Architecture of MILANO" in J. A. Hughes et al. (eds.): *Proc. of Fifth European Conference on Computer Supported Cooperative Work* (ECSCW '97), Lancaster, UK, Sep., 1997, pp. 281-295.
4. Agostini, A., De Michelis, G., Grasso, M. A., and Patriarca, S. "Reengineering a Business Process with an Innovative Workflow Management System: a Case Study" in *Conf. on Organizational Computing Systems*, Milpitas, CA, USA, Nov. 1993, pp. 154-165.
5. Bogia, D. P. and Kaplan, S. M. "Flexibility and Control of Dynamic Workflows in the wOrlds Environment" in N. Comstock and C. Ellis (eds.): (COOCS '95), August 1995, Milpitas, CA, USA, ACM Press, New York, pp. 148-159.
6. Bolcer, G. and Taylor, R. N. "Advanced Workflow Management Technologies", to appear in *Journal of Software Process Practice and Improvement*, accepted for publication January, 1999.
7. Bolcer, G. *Flexible and Customizable Workflow Execution on the WWW*, Ph.D. Thesis, Univ. of California, Irvine, September 1998.
8. Bowers, J., Button, G., Sharrock, W. "Workflow From Within and Without" in Proc. of Fourth European Conf. on Computer Supported Cooperative Work (ECSCW '95), Sep. 1995, Stockholm, Sweden, Kluwer Acad., London, pp. 1-16.
9. Clark, H. H. and Brennan S. E. "Grounding in Communication" in L.B. Resnick et al. (eds): *Perspectives on Socially Shared Cognition*, APA, Washington DC, USA, 1991, pp. 127-149.
10. Cugola, F. "Tolerating Deviations in Process Support Systems via Flexible Enactment of Process Models", in *IEEE Trans. on Software Engineering*, vol. 24, no. 11, Nov., 1998.
11. Curtis, B., Krasner, H., Shen, V., and Iscoe, N. "On Building Software Process Models Under the Lamppost" in Proc. of the 9th International Conference on Software Engineering (ICSE), Monterey, CA, USA, April 1987, IEEE Comp. Soc. Press, Washington DC, pp. 96-103.
12. Doppke, J. C., Heimbigner, D., and Wolf, A. L. "Software Process Modeling and Execution within Virtual Environments" in *ACM Trans. on Software Engineering and Methodology*, vol. 7, no. 1, January 1998, pp. 1-40.
13. Dourish, P. "Freeflow: Mediating Between Representation and Action in Workflow Systems" in M. S. Ackerman (ed.): *Proc. of the ACM 1996 Conference on Computer Supported Cooperative Work* (CSCW '96), Boston, MA, USA, Nov., 1996, ACM, New York, pp. 190-198.

14. Ellis, C. and Rozenberg, G. "Dynamic Change Within Workflow Systems", in N. Comstock and C. Ellis (eds.): (COOCS '95), Aug., 1995, Milpitas, CA, USA, ACM, New York, pp. 10-21.
15. Erickson, T., Smith, D. N., Kellogg, W.A., Laff, M., Richards, J.T., and Bradner, E. (1999): "A Sociotechnical Approach to Design: Social Proxies, Persistent Conversation, and the Design of Babble", To appear: *Proc. of Conf. on Human Factors in Computing Systems* (CHI'99), Pittsburgh, PA, USA, May 1999.
16. Fielding, R. T., Whitehead Jr., E. J., Anderson, K. M., Bolcer, G. A., Oreizy, P., Taylor, R. N., "Support for the Virtual Enterprise: Web-based Development of Complex Information Products". *Communication of the ACM*, vol. 41, no. 8, pages 84-92. August,1998.
17. Fitzpatrick, G., Tolone, W. J., and Kaplan, S. M. "Work, Locales, and Distributed Social Worlds", in Proc. of Fourth European Conf. on Computer Supported Cooperative Work (ECSCW '95), Sep. 1995, Stockholm, Sweden, Kluwer Acad., London, pp. 1-16.
18. Fowler, J., Baker, D.G, Dargahi, R., Kouramajian, V., Gilson, H., Long, K. B., Petermann, C., and Gorry, G.A. "Experience with the Virtual Notebook System: Abstraction in Hypertext", in *Proc. of The Conf. on Comp. Supported Cooperative Work*, Chapel Hill, NC, USA, Oct. 1994.
19. Gasser, L. "The Integration of Computing and Runtime Work", in ACM Transactions on Office Information Systems, vol. 4, no. 3, July 1986, pp. 205-225.
20. Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. "Using Collaborative Filtering to Weave an Information Tapestry", in *Communications of the ACM*, vol. 35, no. 12, Dec., 1992.
21. Grudin, J. "Groupware and Social Dynamics: Eight Challenges for Developers", in *Communications of the ACM*, vol. 37, no. 1, January, 1994, pp. 92-105.
22. Jablin, F. M., Putnam, L. L., Roberts, K. H., and Porter, L. W. "Process: Communication Behavior in Organizations", in F. M. Jablin et al., (eds.) *Handbook of Organizational Communication: An Interdisciplinary Perspective*, Sage, Beverly Hills, CA, USA, 1987.
23. Kammer, P. J., Bolcer, G. A., Taylor, R. N., Hitomi, A. S. and Bergman, M. "Techniques for Supporting Dynamic and Adaptive Workflow", to appear in *Computer Supported Cooperative Work*, accepted for publication March, 1999.
24. Kammer, P. J., Bolcer, G. A., Taylor, R. N., and Hitomi, A. S., "Supporting Distributed Workflow Using HTTP", in *Proc. of the Fifth International Conf. on the Software Process: Computer Supported Organizational Work*, Lisle, IL, ISPA, June 1998, pp. 83-94.
25. Kovalainen, M., Robinson, M., and Auramäki, E. "Diaries at Work", in *Proc. of ACM 1998 Conference on Computer Supported Cooperative Work* (CSCW '98), Seattle, WA. USA, November, 1998, pp. 49-58.
26. Kraut, R. E., Fish, R. S., Root, R.W. and Chalfonte, B.L. (1990): "Informal Communication in Organizations: Form, Function, and Technology", in S. Oskamp and S. Spacapan (eds.): *People's Reactions to Technology in Factories, Offices, and Aerospace*, Sage, London, 1990.
27. Malone, T. W., K. R. Grant, F. A. Turbak S. Brobst and M. D. Cohen. "Intelligent Information-Sharing Systems", in *Commun. of the ACM*, vol. 30, no. 5, May, 1987, pp. 390-402.
28. Medina-Mora, R., Winograd, T., Flores, R. and Flores, F. "The Action Workflow Approach to Workflow Management Technology" in *Proc. on Computer Supported Cooperative Work* (CSCW '92), Toronto, Canada, November, 1992, ACM, New York, pp. 281-288.
29. Posner, I.R. and Baecker, R. M. "How People Write Together", in R. M. Baecker (ed.): *Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration*, Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993, pp. 239-250.
30. Roseman, M. and Greenberg, S. "TeamRooms: Network Places for Collaboration" in M. S. Ackerman (ed.): *Proc. of the ACM 1996 Conference on Computer Supported Cooperative Work* (CSCW '96), Boston, MA, USA, November, 1996, ACM, New York, pp. 325-343.
31. Saastamoinen, H. *On the Handling of Exceptions in Information Systems*, Ph.D. Thesis, University of icJyväkylä, Jyväkylä, 1995.
32. Sakamoto Y. and Kuwana, E. "Toward Integrated Support of Synchronous and Asynchronous Communication in Cooperative Work: An Empirical Study of Real Group Communication", in S. Kaplan (ed.)*: Conference on Organizational Computing Systems* (COOCS '93), Milpitas, CA, USA, November 1993, ACM Press, New York, pp. 90-97.
33. Strong D. M. and Miller, S. M. "Exceptions and Exception Handling in Computerized Information Processes", in *ACM Trans. on Info. Systems*, vol. 13, no. 2, Apr. 1995, pp. 206-233.
34. Suchman, L. "Office Procedure and Practical Action: Models of Work and System Design", in ACM Transactions on Information Systems, vol. 1, no. 4, October 1983, pp. 320-328.
35. Suchman, L. "Do Categories Have Politics? The language/action perspective reconsidered" in *Computer Supported Cooperative Work*, vol. 3, no. 2, 1994, pp. 177-190.
36. Turner, P. and Turner, S. "Supporting Cooperative Working Using Shared Notebooks", Proc. of Fifth Euro. Conf. on Computer Supported Cooperative Work (ECSCW'97), Lancaster, UK, Sep., 1997.
37. Winograd, T. "Categories, Disciplines, and Social Coordination", in *Computer Supported Cooperative Work*, vol. 3, no. 2, 1994, pp. 191-197.
38. Winograd, T. and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design*, Ablex Pub. Corp., Norwood, N.J. USA, 1986.