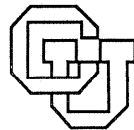Goal Based Models of Groupware

Clarence A. Ellis and Jacques Wainer

CU-CS-668-93   August 1993

University of Colorado at Boulder

DEPARTMENT OF COMPUTER SCIENCE

Goal Based Models of Groupware

Clarence (Skip) Ellis
University of Colorado
Boulder, Colorado, USA

Jacques Wainer
Universidade Estadual de Campinas
Campinas, Brazil

Abstract: In numerous group activities, we find that people do not strictly follow rules, nor strictly follow procedures. People know the personal and group goals, and act accordingly. Thus we see goal based behavior rather than rule based behavior. This paper introduces, justifies, and formalizes notions of goals and beliefs. This paper also introduces groupware paradigms of keepers, synchronizers, communicators, and agents. The primary thesis of this paper is that all four of these types of groupware systems could greatly benefit from incorporation of a goal based model of collaboration. We provide an infrastructure, and a modal logic of goals that enables modelling of group beliefs, preferences, and goals. We define and discuss notions of personal goals, group goals, common goals, and conflicting goals. In the process of presenting an example of the application of this logic, we show how it is possible to model communication, cooperation, conflict, false beliefs, and personal preferences. All of these need to be taken into account in the conception, design, and installation of groupware.

I.   Introduction

Within our societies, we see technologies which appear to greatly advance the human condition (e.g. water purification technology), and others which seem to be questionable in their societal effect (television technology). Convergence of technologies recently has appeared to bring the world closer together, both physically, and conceptually. For example, transportation technology has progressed tremendously, so that physically, we can travel to more places, more safely, in less time, with less effort. Conceptually, the telephone and other communication technologies have made it possible for families, communities, and interest groups to feel closer together although they may be separated by great distances. Groupware is an emerging technology with great promise of bringing people closer together conceptually. Whether people are in the same conference room or scattered around the world, groupware can potentially help them to coordinate, collaborate, and cooperate. However, like many emerging technologies, if not carefully directed, applied, and assessed, it can impose significant negative societal effects. Groupware is the generic name for technologies that support groups of people working (and playing) together. The term was first defined and published by Johnson-Lenz [Johnson82] to refer to computer-based systems plus the social group processes that the systems support. In this paper, we follow the definition published in a groupware overview paper [Ellis91a] that defines groupware as "computer based systems that support groups of people engaged in a common task or goal, and that provide an interface to a shared environment." It is important to note that the system and the group are intimately interacting entities. In either of the above definitions, it is useful to have a firm understanding of the group interactions, and the factors that influence group behavior. This paper attempts to contribute to that firm understanding by carefully defining notions such as common goals, group beliefs, cooperation, and conflict. Successful technological support of a group task or goal is heavily dependent upon a balance between good social processes and appropriately structured technology.

Examples of groupware range from meeting room technology [Nunamaker91] which is typically used by a group at the same time, same place, to electronic mail technology which is typically used by a group at different times, differ-

ent places [Malone87]. This time space taxonomy of groupware has been discussed by numerous authors [Johansen88],[Ellis91a]. In section 2 of this paper, we introduce a functional taxonomy of groupware components. We explain the concept of a goal based model of collaboration, and argue that each category in our taxonomy could greatly benefit from such a model. In section 3 we present one particular goal based model of collaboration within the workflow domain. We illustrate how the model can augment a standard workflow system model, and we also give examples of the benefit of doing this. Section 4 discusses related literature and work. Section 5 presents formal definitions and theorems of group beliefs, goals, and related terms. Our approach is to utilize a possible worlds semantic. We point out that not only is it important for everyone in a group to have (some of) the same goals [*group goals*], but also it is useful for everyone in the group to believe that everyone in the group has the same goals, and that everyone believes that everyone believes that everyone has the goals, and so on [*synergistic goals*]. Group goals that are also synergistic goals are called *common goals*. Section 6 gives an example of all of these definitions, and section 7 presents a summary and conclusions.

We end this section by briefly illustrating how a goal based model can enhance a typical groupware system. The GROVE system is a real-time group editor which allows a distributed group of people to view, discuss and edit a document concurrently [Ellis90]. The system allows participants to create private, shared, or public windows on their display screens to see differing views of different parts of the document. The default data setting is that all information is public, and no data items are locked. This offers the possibility to the group to work over long distances in a more tightly coupled, less inhibited manner than ever before. With this data setting, the system implements an optimistic concurrency resolution mechanism that guarantees consistency [Ellis89]

Our studies and observation of GROVE usage showed that after a few sessions, some groups developed a work style (or group protocol) which allowed them to edit a document concurrently very rapidly, and very synergistically. Conflicting simultaneous edits occurred very infrequently although, at times, the group would all be editing within the same paragraph of a document. When queried about this, they said it was easy and natural; and worked partly because they knew the preferences and goals of the other participants.

This suggests a possible embellishment to GROVE in which it also knows some of the preferences and goals of the participants. At a detailed level, this might mean that the system knows that participant 1 has a preference for paragraph 1, and participant 2 has a preference for paragraph 2. This would be useful information to incorporate within GROVE's concurrency resolution algorithm. Decisions about mediation of conflicting simultaneous edits could be handled more intelligently by a groupware editor which knows some of the goals of the group and the individuals. At a higher level, a model of collaboration for GROVE would incorporate characteristics of documents which are appropriate for different audiences. When we refer to group goals, we are thinking of answers to questions such as "Why is the group composing this document?". The answer to this question can effect the actions and reactions of the group editor. A document written with a goal of convincing management is quite different from a document written for publication in a highly technical international journal. Aspects ranging from the executive summary to dangling citations (items in the reference list which are not referenced in the paper) may be treated quite differently in these different cases.

II.   Groupware Paradigms

In this section, we offer and motivate a new, functionally based taxonomy of groupware components. This taxonomy, which categorizes groupware as keepers, synchronizers, communicators, and agents, is useful to elucidate and understand the spectrum of groupware products currently on the market, and to suggest useful groupware directions which have not yet emerged. In reality, many groupware systems are, and should be, a mixture of the above categories. We also give explanation and examples of groupware in each category. Other taxonomies based upon different criteria have been discussed in the literature; and found useful for understanding and analysis within the groupware area. See section 4 (related work) of this paper for further discussion. Our categorization is functional, and supports our thesis that goal based modelling can be very beneficial. For each category, we suggest some motivating reasons why a goal based model of collaboration would be useful and appropriate. In a later section we explore workflow technology in depth. It is an example of groupware which is currently a hot topic in the marketplace; we show details of a goal based model of collaboration for workflow.

2.1.   The Keeper

One of the functional categories of our groupware taxonomy is the function of the "keeper of the artifact." When a group of designers are working on a complex design, a primary means of interaction for them is to interact through the design artifact. Consider, for example, a group CAD (Computer Aided Design) system which allows a group of designers to concurrently view and edit design diagrams. One designer might say "I implemented a super low cost solution to our cross-over noise problem. Take a look at the auto-CAD." The second designer, upon hearing this might bring up the appropriate working diagram, study it, and understand exactly what the first designer did. Note that this detail was communicated via the auto-CAD diagram - by the work artifact itself. Thus, the groupware system acts primarily as a repository for, and a controller of access to the artifact. Of course, most keepers cannot answer the question "Why are these people manipulating the artifact; what is the ultimate purpose?". This is the purpose of a goal based model; when this question can be answered, the system can help with merging of simultaneous edits, with maintenance of consistency, and can give assistance in attaining the goals. Other examples include group document editors where the artifact is the document, and the graphical issue based information system, gIBIS, where the artifact is the design rationale [Conklin88]. Associated with a keeper is an explicit or implicit object model. This is a description of the repositories, the information items (or types), and the operators on these that the system provides. One can view the artifact as a database of objects, and the keeper as the manager of these objects. The group editor may have text objects (paragraphs, sections, etc.,) and gIBIS has issues, positions, and arguments as its object classes. Generally, object oriented databases and shared hypertext systems fall into this category. We insist that systems falling into this category are for groups' shared usage. These systems pose some vastly different challenges than single user systems [Rodden91]. It has, for example, been pointed out that groupware systems impose different requirements, and are built upon a different philosophy than typical database systems. Databases employ mechanisms such as locking and transactions to insure that simultaneous users are undisturbed by each other; in contrast the philosophy of groupware is to encourage cooperation by making it known and instantly apparent to all who is sharing what with whom. We find that correctness criteria such as database serializability are no longer the appropriate criteria, and need to be re-thought [Ellis89]. Thus, notions of group context and shared environments play a much more central role within groupware.

2.2.   The Synchronizer

Another functional category of groupware is the function of "synchronizer of the group activities" where activities are work tasks or procedure steps. In many group situations, the work or task of one person cannot be started until another person's task is completed. For example, yearly income tax cannot be calculated until after total year's income has been reported. Activities are frequently defined within the context of a procedure which is some partial ordering of work steps. The notion of precedence of activities within a procedure, together with parallelism and coordination, are primary concerns of the synchronizer. Groupware systems which typically fall into this category include group PERT chart programs, and workflow systems. A PERT chart program describes graphically which activities must precede or follow which others, and can calculate quantities such as critical path [Stilian62]. Although this program has no knowledge of the information objects used by the various activities (its not a keeper!), the precedence information imbedded in its display makes the PERT charter a useful synchronizer. Synchronizers such as workflow systems actually utilize the precedence information to synchronize the enactment (execution) of the associated activities. A workflow system that has a specification of activity x precedes activity y may have a feature which inhibits the users from processing activity y on their computers until activity x has been completed and terminated. This feature, depending upon the situation, may be very helpful or a barrier to productive work. For example, if the largest, most important customer requests that her work order be expedited, the fact that the system prevents x and y to be done in parallel may be a severe barrier. We see goal based workflow as a means of avoiding this problem. We will discuss this issue further in a later section when we elaborate a workflow example.

Synchronizers encapsulate control flow. Associated with a synchronizer, there is always an explicit or implicit coordination model. This is a description of the activities, and the precedence relation between them. Some of these coordination models allow parallel activity execution and some do not. Some rely upon and utilize nesting of activities. Finally, some explicitly represent fork, join, choice, and decision making. Most of these models are static - i.e. they do not allow change of this structure during execution. In a related paper, we discuss in detail the issue of dynamic change in groupware systems [Ellis93a].

2.3. The Communicator

A third category is the communicator. More than other categories of our taxonomy, communicators recognize the importance and pre-eminence of humans in groupware, and specifically support human to human communications. Groupware systems are people systems! Somehow groupware must support and enhance people communicating with people. Communicators implement a communications paradigm. In these systems, instead of the user thinking in terms of control flow (synchronizer) or data repositories (keeper), she typically thinks in terms of communication messages, and operations such as send and receive. A salient example is electronic messaging systems (e-mail) which typically allow users to send and reply to text messages asynchronously. It is primarily concerned with getting messages from people to people. As with the other paradigms presented it can be used for other functionality, but this is subsidiary. E-mail may also be used as a keeper by storing and organizing information as messages. It may also be cleverly used as a synchronizer if specific activities are associated with specific messages; the messages can help to control the order of those activities. Nevertheless, the dominant paradigm and the primary built-in functionality of an e-mail system is communication. E-mail systems that understand goals of participants can be very helpful. This can be the gateway to systems which address the social and organizational aspects of group work. Suppose that I send an electronic mail message to the operating systems specialists in my corporation announcing the availability of a technical report which I recently completed. Consider the goals of my communication act. This may have a technical goal of dissemination of relevant information, but also the side benefit to me of getting expert feedback and opinion on my ideas. An organizational goal of stimulating interaction and collaboration among these specialists may be furthered. There may also be a social goal of becoming friends with the recipients. Thus, there can be many goals imbedded in the sending of a message.

There are numerous other examples of communicators. Real time videoconferencing systems fall within this category because they are explicitly designed, built, and bought for communication among people. Even for people involved in a large face to face meeting, technology is available to help them communicate [Nunamaker91]. Video windows and desktop conferencing systems [Ishii91] are recent examples of creative communicators. An important subset is the category of context communicators. The Xerox shared video coffee room [Goodman86] was an example of this - it employed continuous running video cameras and video monitors in the coffee rooms of two Xerox research labs in two different states. This allowed people at remote locations to meet and informally interact with fellow employees. The Cruiser system enabled researchers to electronically scan the offices of coworkers whenever they want to formally or informally have a conversation [Root88]. Office rooms were equipped with video cameras and video windows on the workstation screens off researchers. Each participant could choose to set their video door to open (available to talk,) closed (unavailable,) or partially open (in office, but busy.) Whenever someone chose to go cruising from their own office, they would see a few seconds of scan of each office that was connected and could optionally choose to remain in contact with some office, and start an audio visual conversation. The NICK meeting room [Ellis87] experimented with mood buttons available to all participants during meetings. By hitting a button, a participant could at any time convey to the facilitator a message such as "I'd like to speak" or "I'm bored, lets move on" or "this speaker / topic is really important." Another experiment explored by NICK was to continuously display the averages of this information via meeting mood meters on the electric blackboard at the front of the meeting room. Work by others has continued to explore sharing of social and organizational context. A good communicator is useful for formal and informal communication; it delivers social as well as technical information.

Associated with each communicator is an (implicit or explicit) ontological model of communication which specifies classes of communication possible (see for example [Flores88]) The ontological model may include, among other things, senders, receivers, possible message types, and the set of available operations. Typical operations include send, receive, and possibly edit. These apply to messages, to packets, to frames, or to some other units of transmission. Specializations of the send operation such as reply and forward are prevalent in e-mail systems. If appropriate, display, print, and delete may be available. Transmission of the information, which we said may be text, voice, video, etc., will be synchronous or asynchronous, and may be implemented using concepts such as mail servers, addresses and routes, envelopes, message headers, and message bodies. One example of a model of this type is the model specified by the ISO 4000 messaging standard; many other models have emerged in recent years.

2.4. The Agent

The fourth category is the agent. Besides groupware modules and functional units which are concerned with operation of the whole groupware system. There are modules which are built to perform specific, non-global subtasks. These frequently involve specialized domain knowledge; we call these modules agents. Agents are subsystems (frequently automated, but not necessarily) which implement a specific set of functionality. Examples include user interface agents [Lee93], electronic meeting participants [Gibbs89], and critics [Fischer93]. None of these are concerned with the overall workings of the total system, but each contributes useful functionality in a specialized domain. Thus each is an agent which implies certain characteristics. Characteristics frequently associated with agents include autonomy, distribution, encapsulation, high level interface, and pro-active interaction.

As an example, the notion of a critic has been introduced into the literature as a knowledge based software subsystem which acts as an intelligent automated co-worker who offers some criticism of work which it has been asked to evaluate. In some cases, the criticism may be inappropriate, and humans are free to ignore the criticism. Fischer [Fischer90] writes about the "kitchen critic" which is a software system imbedded in a larger building design aid system. When a team of architects design a kitchen for the home of a client, they specify placement of ovens, sink, etc., within the space of the kitchen. The automated kitchen critic will then look at the design and compare what was specified to its many rules of good kitchen design. It may notice that the oven was placed directly below the window, which is a violation of one of its tenants of good kitchen design. The system will present this finding to the designers, and they are free to alter the design to alleviate this, or to ignore the critic because the owner specifically asked for or needs this arrangement. The non-procedural nature of many agent implementations makes a goal based model very natural. The rules of a critic such as "do not place the stove below the window," and "keep the appliances within a small radius," are particular instances of goals of safety and convenience. Clearly an agent can be much more helpful if she knows these goals as well as the particular rules. Systems such as critics clearly do not fit neatly into one of the categories of keeper, synchronizer, or communicator, but it clearly is groupware. The category of agent is a fitting place for this type of system, because it is a distributed autonomous subsystem, concerned with a specialized domain rather than the general concern of the total design. Some of the typical operations associated with agents are assert, observe, and update knowledge. Numerous papers about agents are available in the artificial intelligence literature; especially distributed AI.

III. Goal Based ICN

3.1. Workflow and ICNs

Workflow is defined as "systems that help organizations to specify, execute, monitor, and coordinate the flow of work items among a work group or organization." [Bull92] These work items are frequently specified as procedures. Flow-PATH is a typical workflow system; it presents forms on the display screens of appropriate users, and assists them to fill in the forms. After a user specifies that she is finished filling out her part of a form (or document), the system directs the form to the display screen (or electronic in-box) of the correct next person(s). It assists and monitors these people in the next steps of the procedure. The system can verify the content of fields of the form, send reminders when deadlines draw near, synchronize access to forms and data, and automatically perform computations such as fill in some fields of forms. Workflow systems need some kind of language or model to allow administrators to create and update procedures and other entities. For this purpose, FlowPATH incorporates the Information Control Net (ICN) model of organizational activity flow. After presenting the ICN model, we indicate some of its limitations, discuss some of the history of workflow, and then suggest an extended ICN model which includes the concept of goals.

The Information Control Net (abbreviated ICN) is a simple, but mathematically rigorous formalism created and designed during the 1970s to model office procedures [Ellis79]. ICNs have been studied in numerous Universities [Dumas91], and used in industry [Bull92]. They have been valuable for capturing office procedures, for mathematical analyses, and for simulation. They have also been successfully used (in their graphical form) as a communications vehicle among modelers and with end users. Documented analyses using ICNs include throughput, maximal parallelism, organizational redesign, and streamlining [Cook80].

The premise upon which ICNs are built is that many types of office work can be described as structured recurring tasks (called procedures) whose basic work items (called activities) must be performed by various people (called

actors) in a certain sequence. Associated with activities are information repositories, data items, and possibly other resources; we do not discuss these in this document. A particular workflow application is created by specifying an ICN description of a set of procedures and activities. We next give a full description of the basic concepts that embody an ICN.

3.2. ICN Definition

Definition: A *procedure* is a predefined set of work steps, and a partial ordering of these steps. Steps can be related to each other by conjunctive logic (after step 1, do steps 2 and 3), or by disjunctive logic (after step 1, do 2 or 3, but not both.) A *work step* consists of a header (identification, precedence, etc.) and a body (the actual work to be done.)

Examples include the "order processing procedure" within an engineering company, and the "claims processing procedure" within an insurance company. Both of these are relatively standardized and structured, and each can be described by a sequence of steps. Different steps of a procedure may be executed by different people or different groups of people. In some cases, several steps of a procedure may be executed at the same time or in any order. In general, we therefore define a procedure to be a partially ordered set of steps rather than a totally ordered set. Partial ordering means that all steps do not necessarily need to be executed sequentially, and that loops are allowed. Procedures typically have attributes, such as name and responsible person, associated with them.

Definition: An *activity* is the body of a work step of a procedure. An activity is either a compound activity, containing another procedure, or an elementary activity.

An *elementary activity* is a basic unit of work which must be a sequential set of primitive actions executed by a single actor. Alternatively, an elementary activity may be a non-procedural entity whose internals we do not model within our structure. An activity is a reusable unit of work, so one activity may be the body of several work steps. For example, if "order entry" and "credit check" are procedures, then the activity "send out letter" may be an activity in both of these procedures. In this case, these are two distinct steps, but only one activity. an activity instance associated with the body of a particular work step is called a work step activity.

Activities typically have attributes such as description and mode associated with them. An activity has one of three modes. Some work step activities may be automatically executed (auto mode.) some completely manual (manual mode.) and some may require the interaction of people and computers (mixed mode.) As an example, if the procedure is "order equipment" then there may be work steps of:

1) order entry,

2) credit check,

3) billing,

4) shipping.

The billing step activity may be automatic, but the credit check step activity probably requires human decision making.

This level of detail of description is typically adequate for an engineering manager, but is not enough detail for an order administrator. The order administrator would like to look inside of the work step called order entry, and see a procedure that requires logging data, and filling out of a form. Thus, the body of this step is itself a procedure with work steps of:

1.1) log name and arrival time.

1.2) fill out the order form,

1.3) send out acknowledgment letter.

Furthermore, the step 1.2 of filling out the order form may itself consist of work steps to fill out the various sections of the form. This example shows that it can be useful to nest procedures within procedures. Thus, a work step body has been defined to possibly contain a procedure. Work steps typically have attributes, such as unique identifier and executor, associated with them.

By definition, a workflow system contains a computerized representation of the structure of procedures and activities. This also implies that there is a means for someone (perhaps a system administrator) to specify and input descriptions of procedures, activities, and orderings into the computer. these specifications are called scripts.

Definition: A *script* is a specification of a procedure, an activity, or an automatic part of a manual activity. the composition or building of this script from available building blocks is called scripting.

Once procedures and activities have been defined, the workflow system can assist in the execution of these procedures. We separate the concept of the static specification of a procedure (the template) from its execution.

Definition: A *job* is the locus of control for a particular execution of a procedure. In some contexts, the job is called a transaction or work case; if a procedure is considered a Petri net, then a job is a token flowing through the net. If the procedure is an object Class, then a job is an Instance.

In our example, if two customers submit two orders for equipment, then these would represent two different jobs. Each job is a different execution of the procedure. If both jobs are currently being processed by the order entry department, then the state of each job is the order entry state. Jobs typically have parameters such as state, initiator, and history associated with them.

One element incorporated in the ICN which sets it apart from flowcharts and other algorithm specification models is the indirect association of people with activities via the concept of roles.

Definition: A *role* is a named designator for an actor, or a grouping of actors which conveniently acts as the basis for the partitioning of work skills, access control, execution control, and authority / responsibility.

Thus, instead of naming a person as the executor of a step, we can specify that it is to be executed by a particular role. For example, instead of specifying that Anna executes the order entry activity, we can specify that

1) the order entry activity is executed by the order administrator, and

2) Anna is the order administrator.

There may be a very large number of work steps in which Anna is involved. When Anna goes on vacation, it is not necessary to find and change all procedures and work steps involving Anna. We simply substitute Anna's replacement, named Mary, in the role of order administrator by changing (2) as follows:

2) Mary is the order administrator.

A role may be associated with a group of actors rather than a single actor. Also, one actor may play many roles within an organization. If there are many order administrators within our example, then these can be defined as a group, and it is easy to send information to all order administrators. In this case, an option may be available to "send to any" administrator, and the system might use some scheduling algorithm to select one. Other flexible scheduling algorithms are possible, including the notification of all members of the group that a job is available, and allowing the first responder to handle the job. In this discussion, we have considered an actor to be a person, but this is not a required interpretation for ICNs. For example, the credit check activity in our example is really executed by the credit department, not by any single person. And the print operation is really executed by one of many print servers that might be

non-human actors with the role of "printer."

Definition: An *actor* is a person, program, or entity that can fulfill roles to execute, to be responsible for, or to be associated in some way with activities and procedures.

Access attributes or capabilities may be associated with actors and with roles. Other attributes, parameters, and structures can be associated to create an enhanced organizational sub-model to capture more of an important dimension within the model. As an example, the role of manager is perhaps only played by Mary within the order entry department. When Mary is sick or absent, the system can determine this manager's manager, and reroute important transactions to her.

Definition: An *Information Control Net (ICN)* is a set of procedures, steps, activities, roles, and actors with a valid set of relations between these entities. Relations include the precedence relation between steps; the part-of relation between activities and procedures; the executor relation between activities and roles; and the player relation between roles and actors. The entity relation diagram in figure 1 graphically shows entities and relations of this definition. One important element of ICNs missing from the diagram is the nested functional abstraction capability [Nutt89] allowing an activity to be recursively expanded as a sub-procedure. More mathematical characterizations of ICNs can be found in other literature, e.g. [Ellis79].

We note in passing that an ICN is also intended to capture a dynamic picture of an organization in action. One can picture the ICN model at any particular time as modelling a snapshot of the organization with various transactions in various states of completion. This is accomplished in the ICN model by formally introducing the concept of tokens, which represent jobs or actions that capture the dynamic state of the system. We refer the reader to an earlier paper [Ellis93b] for a rigorous definition of this aspect of ICNs.
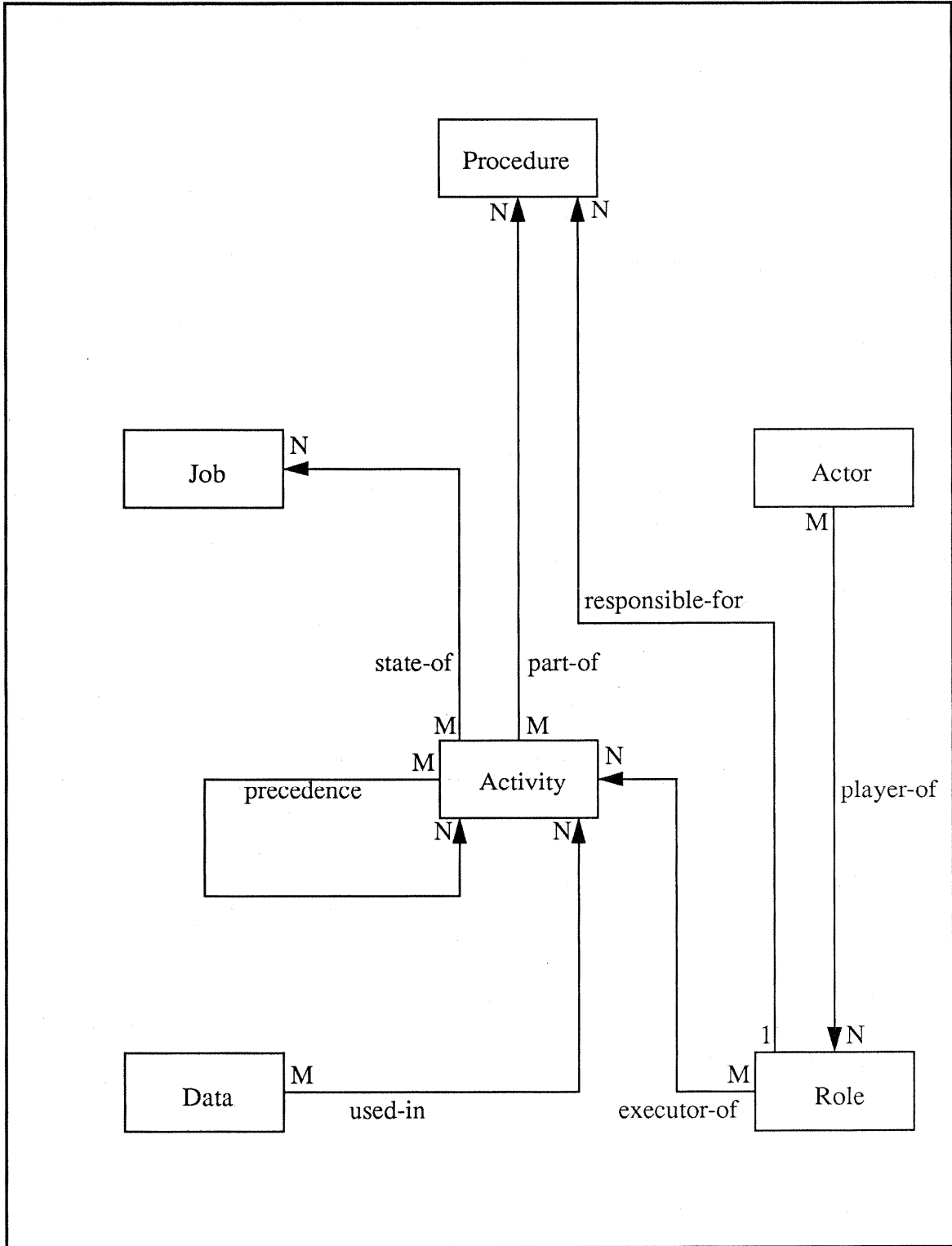
Figure 1: Workflow Conceptual Architecture

Workflow technologies have existed for decades, but despite progress in many areas, useful industrial strength workflow systems are difficult to implement, and not well established. The history of workflow systems in the USA has been mixed. More systems have silently died than been successful [Bair81]. It is frequently found that organizations flourish only if people creatively violate, augment, and circumvent the standard procedures when appropriate. Studies have shown that imbedded in highly structured office procedures is a lot of exception handling, brainstorming, problem solving, and creativity [Strong88]. All of this can be severely inhibited by a workflow system that is overly structured, inflexible, and dictatorial. Lucy Suchman gives a typical, and very understandable example [Suchman83] in which a form arrives on the desk of an order administrator in a sales company. The form has a phone number is the address field. The human immediately recognizes this and telephones to the number; the workflow system simply calculates that this is an invalid address and rejects this transaction with an error indicator. This illustrates the flexibility and creativity of people in handling exceptions which cannot all be preconceived nor preprogrammed into a workflow system. It is a good maxim to have a target of computer assistance of people rather than replacement of people. In many cases, people work through goals rather than only through procedures. If one method of attaining the goal fails, then they creatively try or invent another. A goal based workflow system could be much more amenable and fitting for this work style than the typical workflow system available on the market today. We know of more than 40 workflow products on the market; none of these embody goal based approaches. (One outstanding exception to this is the research work of Bruce Croft during the 1980s on the use of AI planning systems to support loosely structured routine tasks [Croft88]. However, this is research, not a product.) Our research group at the University of Colorado (the Collaboration Technology Research Group) is currently researching goal based concepts, and building a prototype goal based workflow system. We next show one way in which the ICN previously defined can be extended by the notion of goals.

3.3.   Goals within ICNs

There is an evolving understanding that a truly useful workflow model must capture much more than the steps of procedures. Many social and organizational factors play an important role in the day to day working of any organization. Factors include social communication networks, corporate structures and strategies, personal and corporate goals, and other intuitions, morale, and beliefs. These observations lead us to propose the following definition of extended ICN as an important step in this direction. Instead of choosing procedures and activities as the starting point, we choose people and goals. Indeed, when a company is just forming, the procedures and roles are not well formed yet, but there must be motivated people and specific goals for what the company will strive to attain.

Definition: An *Organizational Framework* is a tuple, F = [G, H, R] where G is a set of goals, H is a set of actors, and R is a set of resources.

Examples of classes of resources are information, money, repositories, telephones, office forms, furniture, and word processors. Analysis within this goals and resources model is made feasible by attaching attributes to objects. For example, given the set of people available to do work within an office, it is useful to know each person's skill level, experience, pay expectation, location (if the organization is distributed), personal goals, etc. In general, attributes are associated with classes, so all objects in the same class have the same set of attributes; e.g. all members of the class telephone have a phone number attribute, but office furniture may not. Of course classes can be subdivided into subclasses, so for example in a university, people may be subclassed into categories such as students, faculty, etc., and each subclass has different attributes as in typical object oriented structures.

The second component of our ICN definition adds to the first (framework) part by introducing procedural objects to model structured activities and mappings to model refinement, precedence, and other relations.

Definition: An *Extended Information Control Net* is a tuple, S = [F, O, f_s] where F is an organizational framework, O is a class of procedural objects (e.g. activities) and nonprocedural objects (e.g. roles), and f_s is a set of mappings over F and O. O and f_s capture the procedural definition of ICN given in the previous subsection. As in the previous definition, functional abstraction allows any activity to itself be defined as a procedure or a goal.

In summary, our extended model recognizes that an organization, fundamentally speaking, is comprised of resources (people, money, etc.) and goals. On top of this, we add a second layer which allows these resources to be organized as

roles and procedures. An advantage of this organization is that we can model unstructured as well as structured activity. We illustrate this point by two examples modelled by extended ICNs.

Figure 2 is an example of the graphical ICN model of a simple order processing procedure. Circles represent activities, and arcs represent precedence relations between the activities. Small hollow dots are OR nodes, and small filled dots are AND nodes. The diagram describes that the procedure starts with an order entry activity. Afterwards, a decision is made to either begin credit check, or to skip credit check and begin the billing activity. When billing is completed, and the timer has expired, the final activity of shipping is performed. Notice that the ICN represents who does what by the actor (actually role) boxes, but it does not represent goals. Using our definition of goal based ICN, we can further model this corporation and its work by adding triangles which represent goals of the organization. As we see in figure 3, there are triangles denoting goals of the organization. A high level goal of the organization is to make profit. Two sub-goals of this are shown: maximize customer satisfaction, and expedite sales. The scope of these sub-goals are shown by dashed lines: the maximize satisfaction goal is applicable to the first two activities which are encompassed within its dashed lines, and the expedite sales goal is applicable to the last three activities. In this model, we explicitly model organizational goals, and specify the scope of activities to which they apply. It is also important to model the goals of actors - we address this in a later section. Note that the actor performing the credit check activity is supposed to uphold both of these (potentially conflicting) goals. Thus, we have modelled goals in this simple example, and have noted that goal conflicts can occur. We discuss this further in section 5.

Realizing that there is much work which is unstructured, the goal based ICN is also capable of modelling unstructured activity. Figure 4 shows a very simplified example of a University model which shows unstructured activities as circles without precedence arcs. The actors in this example are students, professors, administrators, etc.; they are not in the picture, but would be represented by role and actor boxes. The goals of the University which are shown are knowledge, education, and research. Education, for example, is usually defined as a quality of mind and character which cannot be guaranteed to appear in every student via the same procedure. Sometimes education is acquired by structured exposure to, and critical evaluation of, a diversity of subjects. For others, education is the process of assimilating the in-depth knowledge and techniques of a specialized discipline. Within structured organizations, it is sometimes appropriate to focus upon the output of the organization. In the University, the output quantity of "educated people" is hard to quantify and measure., so other modelling objectives might be more appropriate to emphasize. The answer to the question "What activities ought to be performed within the University to realize these goals?" depends upon many factors including the qualifications and interests of the students, and availability / capabilities of the faculty, etc. In figure 4, three goals are shown by triangles, and circles show activities such as classes, seminars, examinations, and cultural events, which can occur simultaneously, or in any order.
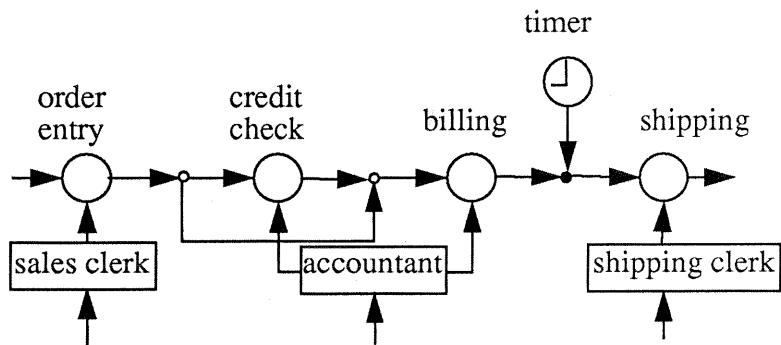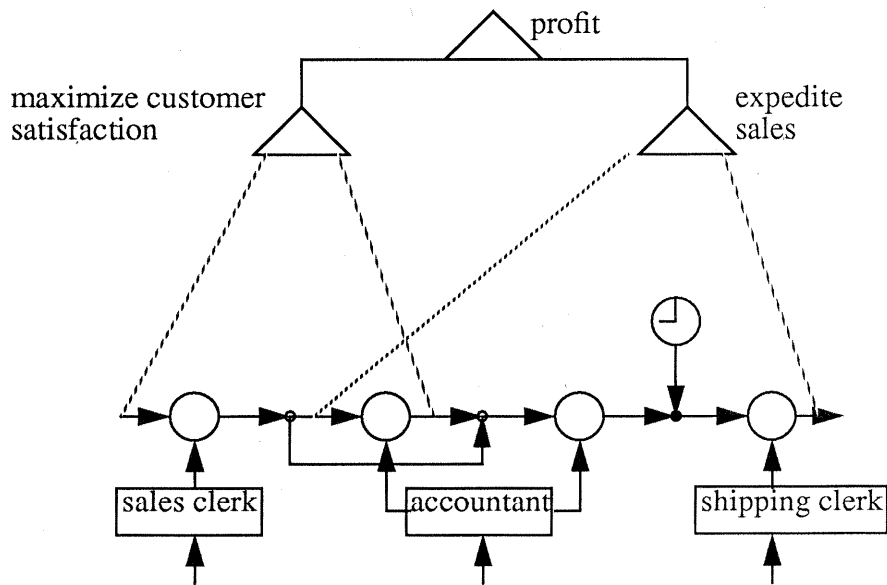
Figure 2: ICN of Order Processing
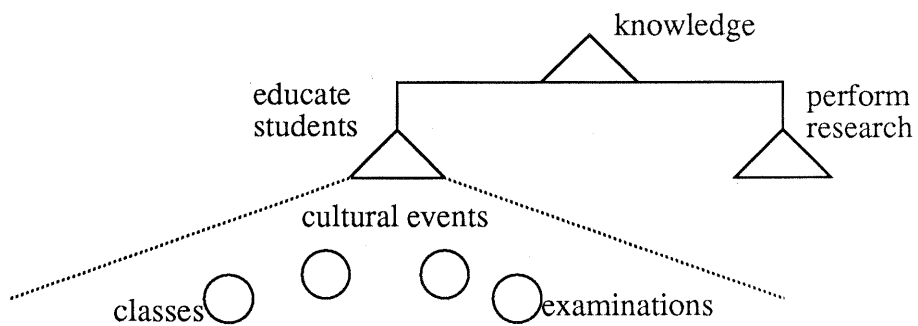

Figure 3: Goal Based ICN of Order Processing

Figure 4: Goal Based ICN of University

IV.   Related Work

The work in this paper is a hybrid subject within the CSCW discipline which itself is highly interdisciplinary. Our research draws upon other work of ourselves and of others in various disciplines. In this section, we overview related work. We have found no work covering the eclectic combination that we discuss, but there have been influential works in related sub-areas that we mention. There has been related work on goals, taxonomies of groupware, modelling of groupware, workflow, and goal based systems.

Goals: Work in this area has a long history and spans multiple disciplines. The work in general problem solvers in the 1960s and 1970s has repeatedly stated the premise that goals are an important and primitive concept in the solving of unstructured problems. In artificial intelligence, goals are an important component behind many kinds of heuristic programming [Newell69], and rule based production systems [Laird86]. There is a large body of literature that defines and applies notions of goals within robotics [Schoppers87]. Within the planning literature, there has been much work to understand and characterize systems and logics of conjunctive goals [Chapman87]. Our notion of goals is also a formal one based upon propositional logic.

Groupware Taxonomies: We put forth the claim that concepts of goals should be integrated with many different classes of groupware. This claim can be supported by showing that it is useful in each category of a taxonomy; however, there are many taxonomies that can apply to groupware. We mention here a subset of them. The taxonomy that we offer in this paper is a functional taxonomy. It classifies according to the type of generic function that a system provides. Closely related is the taxonomy of application domains that has been proposed for groupware [Ellis91a]. This can be contrasted with the time - space taxonomy [Johansen88], the typology of group tasks [McGrath84], the organizational structures taxonomy [Mintzberg84], the office models taxonomy [Newman79], and others.

Groupware Modelling: When a group of users invoke a groupware tool, they implicitly or explicitly are also invoking a model of the purpose and workings of that tool. Frequently these are informal internal unarticulated models in the heads of individuals. We believe that the implicit nature of this model sometimes leads to dissonant and unsuccessful results. It frequently means that the designers' model is far from the users' model, and so the designed functionality is not informed by the user needs; furthermore, the user is not informed of the usage patterns and capabilities that the designer may have insightfully enabled. Therefore it is often very useful to have an explicit model of functionality / collaboration associated with each groupware system and tool. Although the majority of systems do not, we review a few systems which do incorporate a collaboration model.

One recent product which has a very explicit model of collaboration is the CM/1 system marketed by Corporate Memory Systems [CMS92]. They also make available to potential users a three day course because they feel that the system is not simply a new tool; it is a new way of thinking about design and problem solving. Some meeting room technology (e.g. the Arizona/IBM Group Support System [Nunamaker89]) requires that participants adopt set roles, and have an understanding and preparation for the system, its style, and its requirements before meeting. The ForComment asynchronous group editor [ForComment89] structures the interaction so that there is a single author, who releases a document to a set of referees for simultaneous review (in read only mode.) These referees can append comments, and after they are finished, the author can read and selectively incorporate the content of comments. This collaboration model explicitly states and strictly controls who can read or write what at which stages of document processing.

Role Interaction Nets (MCC) [Rein93] are a visual formalism for the design, specification, and enactment of work processes. This research is being worked upon by the coordination research group at MCC. The RIN formalism, which is based upon Organizational Role Theory [Thomas79], describes processes as a collection of organizational role types and interactions among the role types. Although it does not embrace goals, the project has produced several interesting graphical languages which try to carefully match the end user concepts to the multi-dimensional graphics. This work incorporates a rather comprehensive graphical model of work interactions, it emphasizes graphical end user computer interaction environments.

The Comic Project, computer based mechanisms of interaction in co-operative work, is an ongoing ESPRIT basic

research project which aims to develop the theories and techniques necessary to support the development of future groupware systems [Rodden92]. Comic is primarily interested in models for real-time synchronous groupware; it is clear that at some time in the future, the synchronous and asynchronous features and architectures must come together. There is a documented need within workflow to have a system which allows efficient ad-hoc distributed meetings in the middle of workflow problem solving.

Workflow Modelling: We view workflow as one type of asynchronous groupware, so the comments above concerning explicit models of collaboration are applicable to workflow. Some types of workflow are unaware of the organizational structures behind the work that is being done. This can be good to the extent that it insures that the computer systems will not force structuring of activities upon users unnecessarily. In contrast, workflow is organizationally aware, because it must have a representation of the procedures, activities, and actors within the organization. There is also a large literature concerned with the modelling and implementation of systems to assist in the understanding and execution of tasks in the workplace. The European ESPRIT research project Prominand (IABG)[Karbe91] is closely related to our work since it investigates workflow architecture and design with an emphasis on exception handling. The Prominand system is based upon an "electronic circulation folder" paradigm. It is a system which circulates office tasks consisting of steps to be carried out by persons playing office roles. The system attempted to categorize all of the types of exceptions (e.g. lost information; wrong recipient; etc.), and provide handlers for exceptions (e.g. skip next step; return to sender; etc.). Their results illustrate that it is not feasible to identify and program handlers for all exceptions that will occur in the life of a workflow system a priori. In this sense, our approach (goal based) to workflow architecture and exception handling is quite different.

The workflow system by Action Technologies is called Action Workflow [Dyson92]; it is a system based upon the speech act theory of interpersonal communication. It presents an explicit model of coordination in which users are constrained to use one of a small number of types of utterances in their (electronic mail like) conversations. Users are urged to learn about this explicit underlying model before using the system. It is based upon Winograd's work on Speech Act theory [Winograd86]. The theory and the system are built upon a small well defined set of "language acts," and a notion of "conversations" which are coherent sequences of language acts with a regular structure. Within this approach, there are many workflows in an organization, each of which constitutes a basic unit of work. In each, a "performer" commits to produce "conditions of satisfaction" defined by a "customer," on or by a specified time. This system is a follow-on to their previous product; The Coordinator was a mail based product that did not have workflow, and which was a very controversial introduction of speech act theory. Clearly, the jury is still out concerning this approach. Although we are taking a much different approach, we have many goals in common.

There has been other work which addressed systems aspects of workflow. Much of this thrust has roots in the author's early work on Officetalk [Ellis82] and ICNs [Ellis79] during the 1970s. Officetalk was an experimental office information system developed in the Office Research Group at Xerox PARC [Ellis80]. Officetalk was the first system that we know of that provided a visual electronic desktop metaphor across a network of end users' personal computers. It also provided a set of personal productivity tools for manipulating information, a forms paradigm, and a network environment for sharing information. This system was created, evolved, and used extensively within the research lab, and was also tested in selected sites outside of PARC. It evolved into Officetalk-D (database back end, with ICN front end), and Officetalk-P (migrating processes as intelligent forms.)

GMD has implemented several versions of Domino [Kriefelts84], a Petri net based prototype office information system. Usage reports detail numerous problems and reasons for user rejection of the system -- this typifies problems of current workflow. More recently the GMD work has focussed upon tools for assisting more unstructured office work. Other workflow efforts include the Xerox "Collaborative Process Model" [Sarin91], and the WooRKS workflow prototype within the ITHACA ESPRIT project [Ader92]. There are numerous other workflow modelling systems and methodologies - many incorporating techniques for organizational redesign. There are a surprisingly large number of recent corporate efforts to produce workflow products.

Goal Based Systems: None of the systems that we have discussed above consider goal based workflow. The one salient exception to this is the Polymer work of Professor Bruce Croft and his colleagues at the University of Massachusetts [Croft84]. Polymer is a goal based planning system to assist in the performance of multi-agent, loosely structured, underspecified tasks [Croft89]. This work is an excellent starting point for our current research, since it

addresses problem solving and domain representation languages to assist in office work. It uses techniques of AI planning, inferencing, and backtracking. Polymer has served as a testbed to support further research and systems at the University of Massachusetts, including the Spandex, Dacron, and Geneva systems. Earlier related AI work was performed by Fikes at Xerox PARC. His Odyssey system [Fikes81] was a single user inferencing system. He observed that a system which does inferencing and takes actions for the user without informing the user is confusing, even to a single user, especially as complexity increases. Several theses out of MIT have contributed significant useful organizational ideas to goal based modelling and systems design. Li investigated the use of Shank's MOPS as an AI technique for office systems [Li90]. AI work related to office systems has been summarized by one of the authors elsewhere [Ellis88].

V.   Formal Definitions of Group Beliefs and Goals

We have seen that there are various groupware taxonomies and various groupware paradigms. Some researchers and designers have created models to correspond to the paradigm incorporated in their groupware system. A few of these models are even goal based, but what are goals? This section presents a fundamental conceptual definition of group beliefs and group goals.

5.1.   Beliefs

We will start this section by explaining our notation, and discussing a formal definition of belief, and then a formal definition of goal, since these are necessary steps in the definition of group goals, common goals, and implicit goals. Reasoning about knowledge, and thus modeling knowledge has been a research issue in many fields, including philosophy, artificial intelligence [Halpern92], distributed systems [Halpern87], and database theory [Reiter88]. Our work is inspired by the modal theory of knowledge developed within these fields for the formalization of knowledge and belief. We use terminology, symbolism, definitions and techniques very similar to Halpern's to define belief [Halpern and Moses, 1992]. Then we go beyond this work to define and prove theorems about goals and group goals.

5.1.1.   A Formal Definition of Belief

We use some standard terminology of propositional logic; we use Φ to denote a nonempty set of primitive propositions; individual primitive propositions are denoted by $p, q, p', q',...$ Given a set of n actors (people or processes), we will use the symbol B_i to denote the belief of actor i where $B_i(p)$ is true if and only if actor i believes the proposition p. We also denote by $L_k(\Phi)$, the least set of formulas containing Φ which is closed under negation, conjunction, and the modal operators $B_1, ..., B_n$. We denote arbitrary propositions (formulas of $L_k(\Phi)$) by $\phi, \psi, \nu, \mu, ...$ Thus, if ϕ and ψ are formulas of Φ, then so are $(\neg\phi), (\phi\wedge\psi), (\phi\vee\psi), (\phi\rightarrow\psi)$ and $B_i(\phi)$ for $i = 1, ..., n$. The symbol *true* abbreviates any valid formula such as $p\vee\neg p$, and *false* abbreviates $\neg true$. Whereas we define $B_i(\phi)$ to mean that actor i believes ϕ, a related notation, $K_i(\phi)$ means that actor i knows ϕ. In this paper, we use the notion of belief rather than knowledge, because knowledge carries the supposition of being true whereas belief does not. So $K_i(\phi) \rightarrow \phi$ (an actor can only really know a proposition if it is true), whereas it is possible to have $(\neg\phi \wedge B_i(\phi))$ and $(B_j(\neg\phi) \wedge B_i(\phi))$. Thus, an actor can believe a proposition and it is false; also, two actors can have contradictory beliefs - this is not possible with knowledge. Since we are also interested in modelling situations in which group members' beliefs change due to group interaction and influence, the notion of belief is more appropriate than the notion of knowledge which is much more static.

The logic for our belief system will follow the modal logic KD45. The axioms of this logic are:

ϕ if ϕ is a propositional tautology   (Prop)
$(B_i\phi \wedge B_i(\phi\rightarrow\psi)) \rightarrow B_i\psi$ (K)
$B_i\phi \rightarrow \neg(B_i\neg\phi)$ (D)
$B_i\phi \rightarrow B_iB_i\phi$ (4)
$\neg B_i\phi \rightarrow B_i\neg B_i\phi$ (5)

and the following two inference rules:

from ϕ and $\phi\rightarrow\psi$ derive ψ (MP)
if ϕ is an axiom derive $B_i\phi$ (Nec)

Axiom Prop states that all tautologies are axioms. Axiom K states that actors believe all logical consequences of their beliefs. Axiom D states that the belief of any actor is consistent. This means that if an actor believes something is true, then the same actor cannot believe that it is false. Axiom 4 states that actors have positive introspection. This means that if an actor believes something, then she believes that she believes it. Axiom 5 states that actors have negative introspection. This means that if an actor does not believe something, then she believes that she does not believe it.

Inference rule MP states that if a proposition ϕ is true, and $\phi \rightarrow \psi$ is true, then we can infer that ψ is true and inference rule Nec states that if ϕ is an axiom of the logic, then all actors believe that ϕ is true.

5.1.2. Formal Semantics of Belief

It is interesting and useful to describe a formal semantics for belief since similar concepts will be used later to describe goals. The formal semantic model of belief is based upon the possible worlds semantics of Hintikka [Hintikka62], Sato [Sato77], Moore [Moore85], and others, which can be understood as "alternate realities," where things may be different than the "real reality." The basic idea is that the agent selects among the set of all *possible worlds* the worlds that she believes to be possible descriptions of the reality, or "believable worlds." One says that the actor *believes the proposition* ϕ if ϕ is true in all worlds which the actor labels as believable worlds.

Formally, we define a model M as a structure $\{W, \pi, A\}$ where W is a set of possible worlds, π is a valuation function, and A is a set of accessibility relations, $A_1, ... A_n$. The set of possible worlds, W, can be finite or infinite. The valuation function π assigns for each world w_i, the value *true* or *false* to each formula ϕ in $L_k(\Phi)$. This means that a truth value is assigned to every proposition in the world w_i. The accessibility relation A_k (k = 1, 2,...,n) is a binary relation on W x W which captures the believability relation according to actor k. If (w, w') is a member of A_k, then actor k when in world w considers world w' as a believable world.

In order for a model M to be "logically reasonable," it's truth assignments should fulfill certain criteria. We formally define the *satisfiability relation* $\Rightarrow$ between a formula ϕ and a pair (M,w) consisting of a model M and a possible world w within that model as follows:

$(M,w) \Rightarrow p$	iff	$\pi(w,p) = $ true and p is a primitive proposition
$(M,w) \Rightarrow \neg\phi$	iff	$(M,w) \neg\Rightarrow \phi$
$(M,w) \Rightarrow \phi\wedge\psi$	iff	$(M,w) \Rightarrow \phi$ and $(M,w) \Rightarrow \psi$
$(M,w) \Rightarrow \phi\vee\psi$	iff	$(M,w) \Rightarrow \phi$ or $(M,w) \Rightarrow \psi$
$(M,w) \Rightarrow \phi\rightarrow\psi$	iff	$(M,w) \Rightarrow \phi$ and $(M,w) \Rightarrow \psi$ or $(M,w) \neg\Rightarrow \phi$

We define *belief* by saying that within a given model M, actor k in world w believes proposition ϕ if ϕ is true in all believable worlds. Formally:
$(M,w) \Rightarrow B_k\phi$ iff $(M,w') \Rightarrow \phi$ for all w' such that $(w, w') \in A_k$.

In order for the axioms of belief described above to hold there is the need for some further constraints on each of the relations A_k. The relations must be transitive, serial, and Euclidean. Formally:

transitive	if $(a,b) \in A_k$ and $(b,c) \in A_k$ then $(a,c) \in A_k$
serial	if for all worlds w, there is a world w' such that $(w,w') \in A_k$
Euclidean	if $(w,a) \in A_k$ and $(w,b) \in A_k$ then $(a,b) \in A_k$

5.2. Goals

5.2.1. Goal Concept

To explain the concept of goal, we begin with a semantic definition of goal, and then present the axioms of goals afterwards. One of the elusive aspects of goals is the subjective nature of them. They often reflect idiosyncratic priorities and desirabilities in a complex world. Thus, we define the notion of preferred worlds, and we say that an actor

has a goal of φ if in all of the most preferred worlds, φ is true. Using the possible worlds approach, we define P as the set of preference relations, $P_1,...,P_n$. The preference relation P_k (k = 1, 2,...,n) is a binary relation on W x W which captures the preferred worlds according to actor k. If (w, w') is a member of P_k, then actor k when in world w considers world w' as preferable to her current world. The relation P can be interpreted as identifying "more desirable worlds" or "where the actor would rather be." P_k is obviously transitive because if w' is more desirable than w, and w" is more desirable than w', then w" is more desirable than w. Intuitively the goals of the agent are the formulas that are true in all maximally preferred worlds. More formally, we will define the transitive closure of each relation

if there is no w' distinct from w, such that (w,w') ∈ P_k, then w is defined to be maximal, and (w,w) ∈ P_k. This simply says that for worlds that are maximally preferred, the actor would rather stay there.

5.2.2. A Formal Definition of Goal

By adding the set P to our definition of model M and adopting the notation $G_k(\phi)$ to mean that actor k has goal φ, we can now define goal in a similar manner to belief.

We define w to be *maximal* if there is no world w' such that w' $\neg= $ w, and ((w,w') ∈ P_k). We define *goal* by saying that within a given model M, actor k in world w has a goal of proposition φ if φ is true in all maximally preferred worlds. Formally:

$(M,w) \Rightarrow G_k\phi$ iff $(M,w') \Rightarrow \phi$ for all w' such that (w, w') ∈ P_k and w' is maximal.

Given the constraints on P and the definition of goal, the following axioms hold for all goals and all actors:

$G_k\phi$ if φ is a tautology	(Prop$_g$)
$G_k\phi \land G_k(\phi \rightarrow \psi) \rightarrow G_k\psi$	(K$_g$)
$G_k\phi \rightarrow \neg G_k\neg\phi$	(D$_g$)
$G_k\phi \leftrightarrow G_kG_k\phi$	(4'$_g$)
$\neg G_k\phi \leftrightarrow G_k(\neg G_k\phi)$	(5'$_g$)

Axioms (Prop$_g$) and (K$_g$) derive from the fact that the semantics of goals is based on possible worlds: tautologies are true in all possible worlds and thus in all preferred worlds. Similarly, if φ and φ → ψ are true in all preferred worlds, then ψ is true in all of them.

Axiom (D$_g$) states that goals are consistent. Axioms (4'$_g$) and (5'$_g$) state respectively that having a goal is equivalent to having a goal of having that goal; and that not having a goal is equivalent to having a goal of not having that goal. Or looking at the ← direction, the goal of having a goal is trivially achieved, and so is the goal of not having a goal.

5.2.3. Goals and Beliefs

The axioms above describe the properties of the goal operator by itself, and are derived from the constraints on the relation P. We now describe relations between goals and beliefs, first from an axiomatic point of view, and then from the point of view of the joint constraints on the relations A_k and P_k.

The goals of a rational actor must be realistic; that is, goals of an actor must not be believed to be impossible by that actor. In other words, an actor has a goal only if she believes that the goal is possible.

$$G_k\phi \rightarrow \neg B_k\neg\phi \qquad \text{(Rea)}$$

The goals of an actor must be closed under believed implication:

$$G_k\phi \land B_k(\phi \rightarrow \psi) \rightarrow G_k\psi \qquad \text{(K'g)}$$

An actor believes the goals that she has and has the goals she believes:

$$G_k\phi \leftrightarrow B_kG_k\phi \qquad \text{(GB}_1\text{)}$$

The goal of believing something is trivially achieved:

$$G_kB_k\phi \rightarrow B_k\phi \qquad \text{(GB}_2\text{)}$$

These properties will hold if the relations A_k and P_k satisfy the following joint constraint: all maximal worlds in P_k are part of the relation A_k. Formally:

if w is maximal in P_k then there is a world w' such that $(w',w) \in A_k$

5.3. Groups

A group is basically defined as a set of participants (its members) called actors. A group is sometimes more than (and sometimes less than) the sum of its members. The success and synergy of a group depends upon many factors including group beliefs and group goals. Throughout our presentation of goals and beliefs, we have insisted upon a notation which allows us to consider more than one actor. In this subsection, we will utilize that notation to present definitions, theorems, and interpretations which model the group as an entity in and of itself.

5.3.1. Group Beliefs

If members of a group have identical or similar beliefs, there is more likelihood that they can communicate, understand, and empathize. But this depends upon group members having some knowledge or belief of the beliefs of others. For some groups, differing beliefs can cause members to grow and broaden as they learn and adopt beliefs of others. For others, differing beliefs can cause strife and stalemate. We extend the concepts developed above to capture these ideas of the union and intersection of beliefs.

We first define the concept of *group belief* (E_b) as:

Definition 1. $E_b\phi \equiv B_1\phi \wedge B_2\phi \wedge ... \wedge B_n\phi$

A proposition is *common belief* (C_b) if everyone (every actor) in the group believes it, and everyone believes that everyone believes it, and everyone believes that everyone believes that everyone believes it, and so on. Formally:

Definition 2. $C_b\phi \equiv E_b\phi \wedge E_b^2\phi \wedge ... \wedge E_b^j\phi \wedge ... ,$
where E_b^j is defined recursively as $E_b^2 \equiv E_b E_b$, and $E_b^j \equiv E_b^{j-1} E_b$ for $j > 2$.

Although the common belief operator as defined above is "infinite" it can be captured in finite ways by the following axiom and inference rule. The axiom is:

$$C_b\phi \leftrightarrow E_b(\phi \wedge C_b\phi)$$

and the inference rule:

if $\phi \rightarrow E_b\phi$ is an axiom then derive $\phi \rightarrow C_b\phi$

Another concept of interest is that of *implicit belief* of the group (I_b). Implicit belief is the sum belief of the group if they "put their heads together", that is the union of the beliefs of all actors in the group. It is formally defined as:

Definition 3. $I_b\phi \equiv B_1\phi \vee B_2\phi \vee ... \vee B_n\phi$

5.3.2. Group Goal

In order to define common and implicit goals, we follow the framework used for belief. We define *group goal* (E_g) in a similar manner to group belief, E_b.

Definition 4. $E_g\phi \equiv G_1\phi \wedge G_2\phi \wedge ... \wedge G_n\phi$

ϕ is a *common goal* if it is a group goal, and everyone believes that it is a group goal, and everyone believes that everyone believes that it is a group goal, and so on. We again use the finite representation as follows:

Definition 5. $C_g\phi \equiv E_g\phi \wedge E_b E_g\phi \wedge E_b E_b E_g\phi ...$ or in a finite representation $C_g\phi \equiv E_g\phi \wedge C_k E_g\phi$

ϕ is a *synergistic goal* if everyone believes that it is a group goal, and everyone believes that everyone believes that it is a group goal, and so on. Note that this omits the requirement that ϕ is a group goal. We again use the finite represen-

tation as follows:

Definition 6. $S_g\phi \equiv E_bE_g\phi \land E_bE_bE_g\phi ...$ or in a finite representation $S_g\phi \equiv C_b E_g\phi$

Next, *implicit goal* I_g is defined similarly to implicit belief as the OR function:

Definition 7. $I_g\phi \equiv G_1\phi \lor G_2\phi \lor ... \lor G_n$

A *personal goal of actor k*, $\Pi_k\phi$, is a goal of actor k which is not a goal of any other actor in the group:

Definition 8. $\Pi_k\phi = ((G_k\phi \land \neg G_i\phi)$ for all $(i \neq k))$

A *private goal of actor k*, $\Lambda_k\phi$, is a personal goal which is not known (believed) by any other actor:

Definition 9. $\Lambda_k\phi = ((\Pi_k\phi \land \neg B_iG_k\phi)$ for all $(i \neq k))$

5.3.3. Theorems
We next state a few theorems that can be proved about the concepts above.

Theorem 1. $S_g\phi \leftrightarrow C_g\phi$

This theorem states that the concept of synergistic goal is equivalent to common goal. $C_g\phi \to S_g\phi$ is trivial, because of inclusion of the definitions. $S_g\phi \to C_g\phi$ hinges upon the observation that $B_kG_k\phi \to G_k\phi$. If all actors believe that ϕ is a group goal, then actor k must believe that ϕ is a group goal. This, in turn, implies that actor k believes that ϕ is a goal of actor k (herself.) This means $B_kG_k\phi$, and therefore $G_k\phi$. Applying this to all actors results in $S_g\phi \to S_g\phi \land E_g\phi$ which is the definition of $C_g\phi$. Q.E.D.

Theorem 2. ϕ is a tautology $\to C_g\phi$

This theorem states that tautological formulas are common goals. This derives from:
1.   if ϕ is a tautology, then it is a goal of each actor, $E_g\phi$ (by Prop$_g$)
2.   each actor knows (and thus believes) $E_g\phi$, and therefore $E_bE_g\phi$ (by Nec)
3.   true $\to C_bE_g\phi$ implies $C_g\phi$ (by common belief inference rule applied to true $\to E_bE_g\phi$). Q.E.D.

5.3.4. Set Interpretation
In order to present some further definitions, we will use a set notation for goals. We will use the * operator to define the set of propositions that satisfy a certain modal operator. Thus:

$$G^*_i = \{\phi \mid G_i \phi \text{ and } \phi \text{ is not a tautology}\}$$

The reason to leave out tautologies is that we would like to restrict our attention to "real goals," that is the non-tautological goals that actors may have. We define these non-tautological goals as *proper goals*.

$$E^*_g = \{\phi \mid E_g \phi \text{ and } \phi \text{ is not a tautology}\}$$
$$I^*_g = \{\phi \mid I_g \phi \text{ and } \phi \text{ is not a tautology}\}$$

The cooperativeness of a group can be classified based upon whether E^*_g is empty or not, and whether I^*_g is contradictory or not. If the set E^*_g is empty, the actors do not have a proper goal in common, that is, there is not a single proper goal that all actors will agree that should be pursued. We call this an *orthogonal group*. If the set I^*_g is contradictory, then at least 2 actors in the group have opposing goals, that is, there exists at least one ϕ such that one actor wants to pursue ϕ, and another wants to pursue $\neg\phi$. We call this a *polarized group*; see [Bales79]. Although groups which are orthogonal and polarized have no direct incentive to work together, there are many types of strong indirect incentives which means that these groups often, in practice, do work together.

One indirect incentive is social structures, such as peer pressure and friendship; group participant Mary may say "Anna is my friend; I will help her attain her goals." Another indirect incentive is cultural and organizational norms, such as company policy and societal ethics; participant Catherine may say "I am committed to doing good work for my company. It is the right and decent thing to do."

In real settings, it is rare for a group to be completely incapable of working together because participants are frequently *helpful*. We say that an actor is helpful if she adopts as her preferences, those worlds that she believes fulfill another actor's goals, other things being equal. Formally, actor i will be helpful in relation to actor j iff:

$$(B_i G_j \phi \wedge \neg G_i \neg \phi) \rightarrow G_i \phi$$

Thus, if at least one of the actors within a group is helpful in relation to the others, and the others communicate their goals, the group will likely not be orthogonal. We define a group to be *cooperative* if all of its actors are helpful. Also, if a group is polarized, it means that there will likely be moments during which different members of the group may pursue different lines of action, but they may still at other times work well together. Frequently group members have different goals of differing priority, and so may choose not to pursue a lower priority goal in order to pursue a higher priority goal. We illustrate the developed concepts of beliefs and goals; and show that priorities can be incorporated within our model in the following simple example.

VI.   An Example of Group Goals

6.1.   Introduction

As a very simplified example of the concepts defined in the previous section, consider an organization which contains a group with two members consisting of a service person, Mary, and a sales person, Anna. The service motto adopted by Mary is the goal "maximize the number of customers served per hour." The sales motto adopted by Anna is the goal "maximize customer good feelings." If these are the only goals, then Mary and Anna may be considered to form an orthogonal group because the intersection of their goal sets is empty, $I^*_g = \{\}$. Suppose that Mary and Anna also have a goal of "maximize corporate profits;" in this case, they have a group goal, $I^*_g \neq \{\}$. We can formalize this simple example as shown next.

6.2.   Example Definitions and Interpretations

The set of participants (members of the group) are:
        actor 1 = Mary,
        actor 2 = Anna.

The propositional statements are:
        proposition ϕ = "maximize the number of customers served per hour"
        proposition v = "maximize customer good feelings"
        proposition ψ = "maximize corporate profits"

The goals of actors are:
        $G_1 \phi, G_1 \psi$
        $G_2 v, G_2 \psi$

Given a universe of 3 propositions, there are 8 distinct possible worlds:
        $w_1 = \{\phi, v, \psi\}$
        $w_2 = \{\phi, v, \neg\psi\}$
        $w_3 = \{\phi, \neg v, \psi\}$
        $w_4 = \{\neg\phi, v, \psi\}$
        $w_5 = \{\phi, \neg v, \neg\psi\}$
        $w_6 = \{\neg\phi, v, \neg\psi\}$
        $w_7 = \{\neg\phi, \neg v, \psi\}$
        $w_8 = \{\neg\phi, \neg v, \neg\psi\}$

The set of preferences P can be used to describe individual preferences as well as priorities among goals if they exist. Assuming that Mary and Anna are both helpful (defined in the previous section), our definition of P imbeds the specification that a world in which all goals are true is preferable to any world in which only one of the goals is true. Then included in P_1 and P_2 are (w_5, w_1), (w_6, w_1), and (w_7, w_1). As an expression of individual preferences, if actor 1 feels that corporate profit (ψ) is more important than maximizing number of customers served (ϕ), then this could be represented within the model by specifying that a world in which ψ is true but ϕ is not ($\psi \wedge \neg\phi$), is more desirable than a world in which the converse ($\phi \wedge \neg\psi$) is true. This means that $(w_2, w_4) \varepsilon P_1$ and $(w_5, w_7) \varepsilon P_1$.

Note that within our universe, w_1 is a maximally preferred world for both actors. However, in some cases, there may be no joint maximally preferred world. Further semantic embellishment of our example could include another proposition μ = "minimize customer talk time" which suggests that all phone calls and face to face discussions with customers should be very short to allow more time to do more other work. Then it becomes clear that we could also add two more axioms:

$$\phi \rightarrow \mu$$

The above says that if we maximize the number of customers per hour, then this minimizes the time per customer.

$$\nu \rightarrow \neg\mu$$

The above says that if we maximize customer good feelings, then we are spending a lot of time getting to know the customer, chatting with the customer, and helping the customer by talking about any customer concerns, problems, issues, etc. This implies that we are not minimizing customer talk time. Given these two axioms, we can obtain:

$$G_1\phi \rightarrow G_1\mu$$
and
$$G_2\nu \rightarrow G_2(\neg\mu)$$

Thus, $I^*_g = \{\phi, \nu, \psi, \mu, \neg\mu, ...\}$. Due to the derived goal μ of agent 1, and $\neg\mu$ of agent 2, the set I^*_g is contradictory, so this group is polarized. Again, this does not mean that the group members will not cooperate; cooperation can be strongly influenced by factors such as beliefs of group members.

6.3.   Example Beliefs
We look at our example under cases of "no belief," "full true belief," and "false belief."

Suppose that each of the two actors has no knowledge nor belief about the goals of the other. In particular, $\neg B_1 G_2 \psi$ and $\neg B_2 G_1 \psi$. In this case Mary and Anna have a group goal ($E^*_g = \psi$), but no common goal ($C^*_g = \{\}$). Via group communication, Mary and Anna may come to know and believe all of the actual goals of both actors. They will then realize that they have a common goal ($C^*_g = \{\psi\}$), and that there is a conflict of goals at μ. This enables them to provide help to each other, and to sense that they are a polarized group. Another scenario is the possibility of Mary and Anna having false beliefs. Suppose, for example, that Mary and Anna each believe that the other person also holds all of their own goals. This is modelled as:
$B_1 G_2 \phi, B_1 G_2 \psi, B_1 G_2 \mu$
$B_2 G_1 \nu, B_2 G_1 \psi, B_2 G_1 (\neg\mu)$

This situation could potentially stimulate very good cooperation temporarily. However, it is likely that by talking, interacting, and working together, there will be quick realizations that the beliefs were wrong, or there will be surprise rude awakenings at a later time in the interactions.

VII.   Summary and Conclusions
This paper has presented the case for goal based modelling. It was suggested that goals are a useful added ingredient in keepers, synchronizers, communicators, and agents. As an example, a typical workflow model was embellished to support concepts of goals. In the latter part of the paper, we carefully defined notions of beliefs and goals, and group / common beliefs and goals. An example application of this formalism showed how we can analyze a group for polarization, cooperation, conflict, and helpfulness. We believe this is significant in its own right, and as a basis for group-

ware modeling and development.

We feel that the study of goal based systems is just beginning. The model of goals that we presented makes an assumption of logical omniscience which is inappropriate in some situations. Thus, it is useful to study alternative axiomatizations such as T, S4, S5, and others. It is also useful to study properties of complexity and completeness for our logic system.

Our research group is currently architecting a goal based workflow system which we expect will reflect the definitions provided here, and will exhibit much goal based flexibility. We are also expanding the presented model to incorporate subgoals and priorities. The model has an obvious next step of specifying ways in which worlds can change states, thus making this a dynamic model. Besides this further theory work, there is a need to apply the model to organizational studies and to experiment with the incorporation of goal based notions in various groupware (as the group editor that was previously suggested.) Ultimately, we believe that the functional categories elucidated in this paper will merge, and that there will also be a seamless merging of synchronous and asynchronous groupware.

VIII.   References

Ader, M., Lu, G., "The WooRKS Object Oriented Workflow System," OOPSLA92 Exhibition, Booth 712-714, October 19-21, 1992. Developed as part of the ITHACA Research project within the ESPRIT Program.

Bair, J., *Office Automation Systems: Why Some Work and Others Fail,* Stanford University Conference Proceedings, Stanford University, Center for Information Technology, 1981.

Bair, J. "Methods for Success with New Workflow Systems," GroupWare'92, by D. Coleman (ed.), Morgan Kaufmann Publishers, San Mateo, Ca., 1992.

Bales, R. F., Cohen, S. P., *SYMLOG, A System for the Multiple Level Obesrvation of Groups.* The Free Press, 1979.

Barber, G. "Supporting Organizational Problem Solving with a Workstation" ACM Transactions on Office Information Systems (1,1) 1983.

Bull Corporation, *FlowPath Functional Specification,* Bull S. A., Paris, France, September, 1992.

Chapman, D., "Planning for Conjunctive Goals" AI 32, 1987.

CMS, "The CM/1 System Users' Manual" by Corporate Memory Systems, Inc., 1992.

Cook, C., "Office Streamlining Using the ICN Model and Methodology," Proceedings of the 1980 National Computer Conference. June, 1980.

Conklin, J., Begeman, M., "gIBIS: A Hypertext Tool for Exploratory Policy Discussion" in Proceedings of CSCW'88, Portland, Oregon, Sept. 1988.

Croft, W. B. and Lefkowitz, L. S. "Task Support in an Office System," ACM Transactions on Office Information Systems (2, 3) July, 1984.

Croft, W. and Lefkowitz, L. "Using a Planner to Support Office Work," Proceedings of ACM COIS'88, March, 1988.

Croft, W. and Lefkowitz, L. "Planning and Execution of Tasks in Cooperative Work Environments," in IEEE AI, 1989.

Davis, D. B. "Software That Makes Your Work Flow," Datamation, 37, 8, April, 1991.

De Jong, P. "Structure and Action in Distributed Organizations," Proceedings of ACM COIS'90, April, 1990.

Dumas, P., *La Methode OSSAD,* Les Editions d'Organization, 1991.

Dyson, Esther,"Workflow," Release 1.0, EDventure Holdings, New York, September, 1992.

Ellis, C., "Information Control Nets: A Mathematical Model of Office Information Flow," Proceedings of the 1979 ACM Conference on Simulation, Measurement and Modeling of Computer Systems, August, 1979.

Ellis, C. and G. J. Nutt, "Office Information Systems and Computer Science," *ACM Computer Surveys,* (12,1) March, 1980.

Ellis, C. "OfficeTalk-D, An Experimental Office Information System" in Proceedings of the First ACM Conference on Office Information System, June 1982.

Ellis, C. , et.al. "Project Nick: Meeting Analysis and Augmentation" in ACM Transactions on Office Information

Systems 5,2, April 1987.

Ellis, C., "Artificial Intelligence in Office Information Systems" with P. Cook, B. Desai, C. Frasson, J. Mylopoulis, N. Naffah, presented at the Joint National Computer Conference, 1988.

Ellis, C., Gibbs, S. "Concurrency Control in Groupware Systems" in Proceedings of the ACM SIGMOD'89 Conference on the Management of Data, May 1989.

Ellis, C., S.J. Gibbs, and G. Rein "Design and Use of a Group Editor" in Engineering for Human Computer Interaction, G. Cockton, editor. North Holland, Amsterdam, 1990.

Ellis, C. , S. J. Gibbs, and G. L. Rein, "Groupware: Some Issues and Experiences," Communications of the ACM, (34,1), January, 1991a.

Ellis, C., Rein, G., "rIBIS: A Real Time Group Hypertext System" in the International Journal of Man Machine Systems, 34, 1991b.

Ellis, C. "Workflow Doesn't Work" presented to the French Informatics Society, July 1991c.

Ellis, C., Keddara, K., "Dynamic Change Within Workflow Systems" University of Colorado Technical Report, July 1993a.

Ellis, C., Nutt, G. J.,"Modelling and Enactment of Workflow Systems" in Proceedings of the 1993 International Conference on Petri Net Theory and Applications, Chicago, June 1993b.

Engelbart D. "Toward High-Performance Organizations: A Strategic Role for Groupware," Proceedings of the GroupWare'92 conference, edited by D. Coleman, Morgan Kaufmann Publishers, San Mateo, Ca., 1992.

Fagin, R., Halpern, J. Y. "Belief, Awareness, and Limited Reasoning: Preliminary Report," in Proceedings of the 9th IJCAI Conference, 1985.

Fikes, R. "Odyssey: A Knowledge Based Assistant," in Artificial Intelligence, vol.16, 1981.

Fischer, G, et. al. "Using Critics to Empower Users," Proceedings of the ACM CHI'90 Conference on Human Factors in Computer Systems. Seattle, WA; April, 1990.

Fischer, G, et. al. "Critics: An Emerging Approach to Knowledge Based Human Computer Interaction," International Journal of Man AMachine Studies (in press), 1993.

Flores, F., Graves, M., Hartfield, B. and Winograd, T. "Computer Systems and the Design of Organizational Interaction," ACM Trans. Office Information Systems, 6, 2, April, 1988.

ForComment, "The ForComment System Users' Manual" 1989.

Gasser, L. "The Integration of Computing and Routine Work," ACM Transactions on Office Information Systems, (4, 3), July,1986.

Gibbs, S. "LIZA: An Extensible Groupware Toolkit" in Proceedings of the ACM SIGMOD'89 Conference on Human Factors in Computer Systems, May 1989.

Goodman, G., Abel, M., "Collaboration Research in SCL" in Proceedings of CSCW'86, Austin, Texas, Dec. 1986.

Grudin, J. "Why CSCW Applications Fail," Proceedings of the CSCW88 Conference, ACM,1988.

Halpern, J. Y., Moses, Y., "A guide to completeness and complexity for modal logics of knowledge and belief," Artificial Intelligence, (54,3), 1992.

Halpern, J. Y., "Using Reasoning about Knowledge to Analyze Distributed Systems," in J. F. Traub, B. J. Grosz, and B. W. Lampson (eds.), Annual Review of Computer Science, Annual Reviews Inc., 1987.

Hintikka, J., *Knowledge and Belief*, Cornell University Press, Ithaca, N.Y., 1962.

Ho, C., Hong, Y. and Kuo, T. "A Society Model for Office Information Systems," ACM Transactions on Office Information Systems, (4, 4), April, 1986.

Ishii, H., Miyake, N., "Toward an Open Shared Workspace" Communications of the ACM (34, 12), December, 1991.

Johansen, R., *Groupware: Computer Support for Business Teams*.The Free Press, 1988.

Johnson-Lenz, P., Johnson-Lenz, T., "Groupware: The Process and Impact of Design Choices" in Computer Mediated Communication Systems. Kerr, E.B., Hiltz, S.R. (eds.) Academic Press, 1982.

Konolige, K., " What Awareness Isn't: A Sentential View of Explicit and Implicit Belief"in Proceedings of the Conference on Theoretical Aspects of Reasoning about Knowledge. Los Altos, CA, 1986.

Kreifelts, T. "Coordination of Distributed Work: From Office Procedures to Customizable Activities," Verteilte Kunstliche Intelligenz und Kooperatives Arbeiten, 4. Internationaler GI-Kongress Wisensbasierte Systeme, Munchen, Germany, Oct., 1991.

Kreifelts, T., Licht, U., Seuffert, P. and Woetzel, G. "DOMINO: A System for the Specification and Automation of Cooperative Office Processes," Proc. EUROMICRO'84, edited by Wilson and Myrhaug, 1984.

Laird, J., *Universal Subgoaling and Chunking*, Kluwer Publishers, 1986.

Lee, K. C., et. al., "A Framework for Controlling Cooperative Agents," IEEE Computer, (26, 7), July 1993.

Leung, Y. "Workflow Products Market Report" B.S.A. Technical Memorandum, Bull Corporation, 1992.

Levesque, H. J., " A Logic of Implicit and Explicit Belief ," In Proc. of the National Conference on Artificial Intelligence, 1984.

Li, Jianzhong. *AMS: A Declarative Formalism for Hierarchical Representation of Procedural Knowledge*, PhD Thesis, L'Ecole Nationale Superieure des Telecommunications, Paris, France, December, 1990.

Luqi "A Graph Model for Software Evolution," IEEE Transactions on Software Engineering, (16, 8), August 1990.

Martial, F. "Activity Coordination via Multiagent and Distributed Planning," Verteilte Kunstliche Intelligenz und Kooperatives Arbeiten, 4. Internationaler GI-Kongress Wissensbasierte Systeme, Munchen,Germany, Oct. 1991.

Malone, T., et. al., "Semistructured Messages are Surprisingly Useful for Computer Supported Coordination," ACM Transactions on Office Information Systems (5,2) April 1987.

McGrath, J. E. *Groups: Interaction and Performance*, Prentice Hall, Inc. NJ, 1984.

Medina-Mora, P. et. al., "The Action Workflow Approach to Workflow Management Technology," Proceedings of ACM CSCW'92, Oct. 31-Nov. 4, 1992.

Medina-Mora, P. "Action Workflow TM Technology and Applications for Groupware," GroupWare'92, edited by D.Coleman, Morgan Kaufmann Publishers, San Mateo, Ca., 1992.

Mintzberg, H., "A Typology of Organizational Structure" in Miller, D., and Friesen, P. (eds.) Organizations: A Quantum View. Prentice Halol, 1984.

Moore, R., "A Formal Theory of Knowledge and Action" in Formal Theories of the Commonsense World, Ablex, Norwood, N.J., 1985.

Newell, A. "Heuristic Programming: Ill Structured Problems" in Progress in Operations Research 3, Wiley, 1969.

Newman, W. "Office Models and Office Systems Design," In Naffah, N. (Ed.) Integrated Office Systems, Burotics North-Holland, 1979.

Nunamaker, J., et. al., "Electronic Meeting Systems to Support Group Work," Communications of the ACM (34,7) July 1991.

Nunamaker, J., et. al., "Experiences at IBM with Group Support Systems," Decision Support Systems (5,2) June 1989.

Nutt, G. J., "A Flexible Distributed Simulation System" in Proceedings of the Tenth Annual Conference on Applications and Theory of Petri Nets. Bonn Germany, June 1989.

Poltrock, S., Grudin, J., "Tutorial on Computer Supported Cooperative Work and Groupware," Presented at ACM SIGCHI Conference on Human Factors in Computing Systems, New Orleans, April 27, 1991.

Rein, G., *Organization Design Viewed as a Group Proces Using Coordination Technology*, PhD Thesis Dissertation, Department of Information Systems, University of Texas at Austin. May 1992.

Reiter, R., "On Integrity Constraints" in Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge, 1988.

Rodden, T., Blair, G., "CSCW and Distributed Systems: The Problem of Control" in Proceedings of the Second European Conference on Computer Supported Cooperative Work, September, 1991.

Rodden, T. "The COMIC Project-COmputer-based Mechanisms of Interaction in Cooperative Work," ESPRIT Basic Research Project #6225, Technical Annexe, 27th, April, 1992.

Root, R., "Design of a Multi-media Vehicle for Social Browsing" in Proceedings of CSCW'88, Portland, Oregon, Sept. 1988.

Sarin, K. S., Abbott, K. R. and McCarthy, D. R. "A Process Model and System for Supporting Collaborative Work," ACM COCS'91, pp. 213-224.

Sasso, W. C., Olson, J. R. and Merten, A. G. "The Practice of Office Analysis: Objectives, Obstacles, and Opportunities," IEEE Office Knowledge Engineering, 1,1, 1987, pp.11-24.

Sato, M., "A Study of Kripke Style Methods for Some Modal Logics" Publ. Res. Inst. Math. Sci. (13,2) Kyoto University, 1977.

Schoppers, M., "Universal Plans for Reactive Robots" in Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1987.

Stilian, G. N., et. al., "PERT, A Management Planning and Control Technique" American Management Association publication #74, 1962.

Strong, D. M. *Design and Evaluation of Information Handling Processes,* Ph.D. Dissertation, Carnegie Mellon University, 1988.

Strong, D. M., Miller, S. M. "Exception Handling and Quality Control in Office Operations," Working Paper Number 89-16, Boston University, Boston, MA., September, 1989.

Suchman, L. A. "Office Procedure as Practical Action: Models of Work and System Design," ACM Transactions on Office Information Systems, 1, 4, October, 1983.

Thomas, E. J., Biddle, B. J., *Role Theory: Concepts and Research.* Krieger Publishing, 1979.

Tueni, M., Li, J. and Fares, P. "AMS: A Knowledge-based Approach to Task Representation, Organization and Coordination," Proceedings of ACM COIS'88, March, 1988.

Von Martial, F. "A Conversation Model for Resolving Conflicts among Distributed Office Activities," Proceedings of ACM COIS'90, April, 1990.

Winograd, T. and Flores, F. *Understanding Computers and Cognition: A New Foundation for Design,* Ablex Publishing Co., NY, 1986.

Zigurs, I., Kozar, K., "An Exploratory Study of Roles in Computer Supported Groups" submitted to the Journal of Management Information Systems, 1992.