A Task-Centered Theory of Cooperative Work

Nehru Bhandaru, W. Bruce Croft, Dirk E. Mahling
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 90-25
March 1990

A Task-Centered Theory of Cooperative Work [1]

Nehru Bhandaru, W. Bruce Croft, and Dirk E. Mahling
Collaborative Systems Laboratory
Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003
Telephone: (413) 545 0463
Electronic Mail: bhandaru@cs.umass.edu
croft@cs.umass.edu
mahling@cs.umass.edu

Abstract

The design of collaborative systems has been more of an art than a science, due to the lack of comprehensive theories. Even those collaboration support systems that are based on an implicit theory are often tailored to a domain. The absence of a generic theory of cooperation makes it difficult to formally describe issues in CSCW, engineer appropriate systems, and evaluate other systems.

We propose a task-centered theory of cooperation that is founded on a goal-based view of collaborative activities, action psychology, interface design, and computational theories. Our theory contains the major elements in cooperative environments (people and organizations, their surroundings, beliefs and goals, and the computer with its interface) as well as a hierarchy of generic activities. We present the structure of generic activities in a goal-based representation and then discuss the importance of combining problem solving and communication approaches. An example of cooperative work is described using the theory.

[1]This work was supported by the Office of Naval Research, under a University Initiative Research Grant, Contract # N00014-86-K-0764 and by Triumph-Adler, Germany

1 Introduction

The absence of a theoretical foundation for CSCW makes analysis and design of cooperative systems a difficult enterprise. Other fields, such as cockpit design and engineering psychology, show the benefits of scientific research and engineering based on theoretical frameworks. To build systems that can effectively support groups of people we need a general theory that allows the analysis of cooperation and the derivation of design specifications for systems and interfaces.

Traditional approaches to CSCW have focused on communication, sharing, and changes in organizational structures [MGL+87,FGHW88]. To support all aspects of cooperation, we need to expand this focus by acknowledging task context. People do work and want to see results. A task-centered theory of collaboration leads to systems that address the hard issues of getting group work done through efficient and effective support. Previous theories are mostly concerned with organizational structure and the introduction of computer systems in the organization [Gre88]. These theories are analytical and cannot be used to derive design implications for a CSCW system.

A number of approaches exist for designing cooperative systems. The first approach is technology-driven [SG85]. The system designer addresses a problem by using the most current technologies available to build a system. This usually leads to systems that are sophisticated and overloaded beyond user demand and comprehension. The second approach is user driven. Designers interact with users to build systems that fulfill the needs of the user community [SC88,Suc85,WF86]. This approach is prone to missing functionality not demanded or envisioned by the users, and duplicating research and development efforts.

A third approach is based on domain theory, user participation and sound engineering principles [Mah90,Ras86b,VR84,Lan87,WW87]. This approach starts by setting goals with the users. It develops a theory of the domain, drawing implications from the theory for a system, building the prototype and involving users in major design decisions. Such an approach has a number of advantages: It moves system and interface design from the domain of art to engineering; it allows for the systematic accumulation of knowledge; it allows the synthesis of other approaches; and it brings order to our understanding of the subject [Gre88].

We call this third approach Integrated System Design (ISD). It combines the approaches of software engineering and cognitive engineering, maintaining a focus on the proactive use of theories from all cognitive sciences for the design of systems. To build relevant systems, the focus has to be on identifying the goals of users. This can invoke traditional analytic methods or approaches like ethnography. To formalize goals, task-analytical methods can be used [Ras86a].

Dacron [Mah90] and Coordinator [FGHW88] are two of a small number of systems built using implications of very specific theories. In one case (Dacron), the system is based on a tailor-made theory of task recall, in the other (Coordinator), speech act theory is the focus. In both cases, the theories are too narrow to serve as a general basis for CSCW.

A task oriented theory that captures many aspects of cooperative work generically should address at least the user, the group, the context, the system's functionality and the interface. The users must be represented with their perceived world, their intentions and goals, their commitments, and their perception of the group. The theory must allow for design implications and proactive rather than explanatory use.

In the following section, we introduce a generic theory of cooperative work. The scope of the theory, its activities, states, and the flow of control are discussed. In section 3, an example is given for coordination in the purchasing domain using a hypothetical system based on the theory. Conclusions about the approach and the theory presented here are drawn in section 4.

2 A Generic Theory of Cooperative Work

We view cooperative activity as goal directed problem solving and communication among a group of people. We focus on three aspects for describing the structure of this interaction: generic activities in cooperation, states of cooperative activity, and relations between generic activities and states. We extend earlier work on activities in collaboration [KGE88], stages of human computer interaction [ND86,MCC90], and structure of conversations in group work [FGHW88], and integrate these perspectives into a design theory. The basic elements of our model are:

- *People* engaged in the activity.
- The *Real World.* We distinguish the real world and world as perceived by people.
- *Beliefs* of these people - e.g., their agenda of plans and intentions, perceived worlds, roles in activities etc.
- *Activities* of people.
- *Computer Systems* and *User Interfaces.*

People have multiple goals and perform many activities concurrently. The activities they perform to achieve these goals change their beliefs, along with the real world. The real world for a person consists of the objects in the work environment. In cooperative environments the worlds for different agents intersect. An example of this is a document that is shared between people.

The beliefs of a person are complex and varied. These include the perceived world, the agenda of plans and intentions, the roles people play in activities, the plans and agendas of others, and expectations about their actions. Sharing of beliefs is a central aspect of cooperative interactions.

Figure 1 shows the relations between the elements of the theory from the perspective of an individual, called the user. This user perceives the real world, with his organization and colleagues being part of the world. The user also perceives the interface of a collaboration system (*gulf of perception* [ND86]). In the case of an intelligent system, a link between the real world and the system exists, which allows the user to see the system's perception of the world. The current state of the user, the system and the interface constitute reality at a point in time. Any one of these four elements can now do something to change that state. We call that performing a generic activity. If the activity is initiated by the user, an expectation about the change in the real world or the interface will form in the user's mind. This expectation will influence the user's perception of the actual changes. After the activity is completed, a new state in time is reached, again comprising the world, the system, the interface and the user.

2.1 Generic Activities

Our theory is based on a set of activities common across all cooperative tasks [Mac69,Fik82,KGE88], which we call *generic activities.* Examples of these activities include planning, evaluation, establishing commitments for sharing task performance, and resource allocation. Sharing is a central aspect of cooperation. Generic activities make the sharing of resources, domain objects and beliefs possible. Communication and negotiation are fundamental to cooperation because they are part of all generic activities.

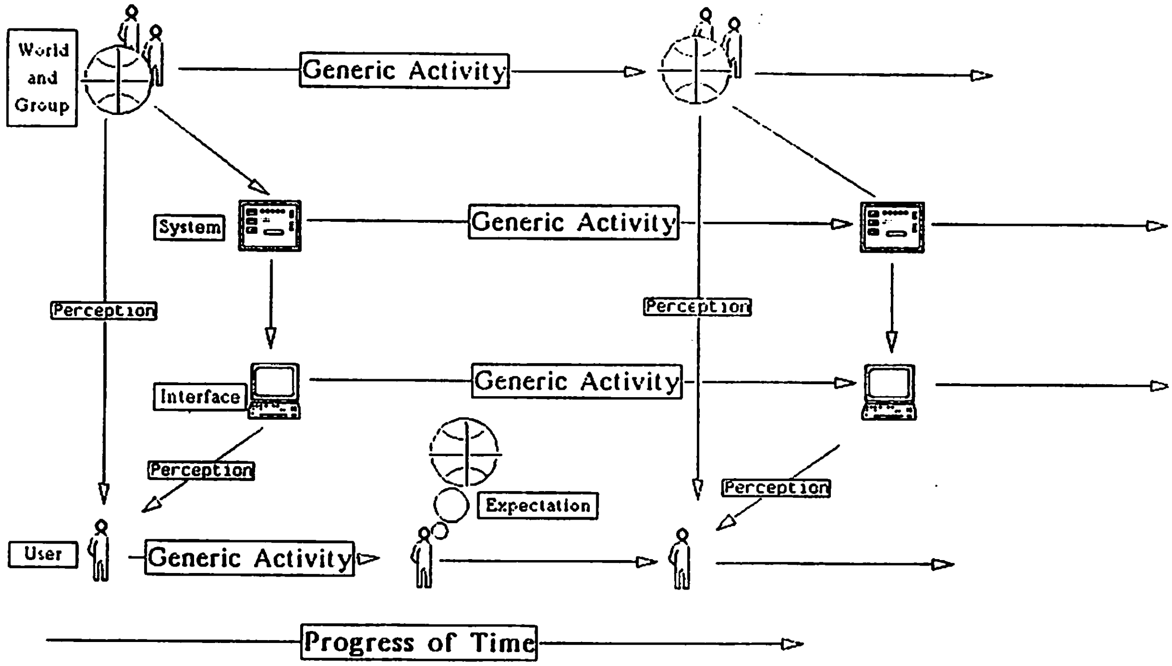

Figure 1: Elements of the Model

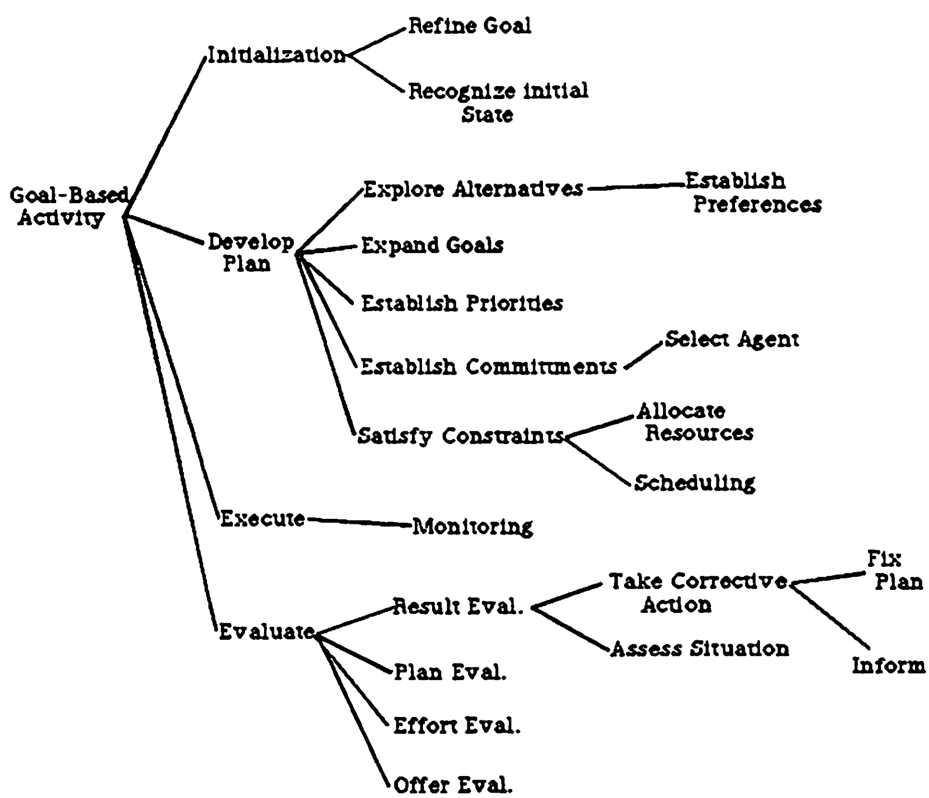

Figure 2: Generic Activities in Goal directed Cooperative Work

Figure 2 shows the hierarchical structure of these generic activities. Initialization results in the establishment of a goal for a person or a group. Formation of a goal from a context (or situation) and intentions of a person is a complex process [Boe80]. Goals may result from failed expectations or agents requesting one another to act. Goal establishment involves communication [2] to share beliefs about the situation and refine intentions into well-defined goals.

Planning establishes a structure for handling the complexity of the goals to be achieved [Fox81]. It involves exploration of alternative ways of achieving a goal. In this process people and computer systems pool their knowledge, communicate their preferences, and negotiate agreements between conflicting or differing evaluations. We refer to people and intelligent computer systems as "agents" of the activity.

Planning entails establishing commitments for sharing task performance between agents involved in the domain activity. Allocation of resources to a plan is also an integral part of the planning process. In the face of resource constraints, agents need to prioritize their domain activities, and allocate resources to those they consider important. The planned activity is performed and evaluated in the context of the agreements resulting from planning.

Execution of an action changes the real world, and hence the perceived world for the agents. Agents need to monitor their activities, and respond to the unexpected situations that arise. The effects of the execution of an action may be unexpected, as many activities are concurrent in the real world. [3]

We distinguish four kinds of activities in evaluation. Effort evaluation considers the number and duration of the resources used, the manpower required, and the like. Plan evaluation deals with analyzing a single plan, or comparing and contrasting multiple plans for achieving a goal. Agents evaluate the results of an action by comparing it to their expectations, which may lead to evaluation of current plans, and their modification. Finally, offer evaluation results in a preferred response for an agent, when another agent requests this agent to perform a particular activity.

2.2 States in Cooperative Tasks

The state of an activity can be characterized by beliefs of agents [4] and state of the real world. The state is a predicate over the 4-tuple of states of agent beliefs, state of the world, system, and interface. For example, A partial description of a state, where the agents have commitments established is:

$$\textit{has-goal(Harry, design-kitchen), is-plan(plan}_1\textit{,design-kitchen)}$$
$$\textit{role-in-plan(carpenter, plan}_1\textit{), role-in-plan(plumber, plan}_1\textit{)}$$
$$\textit{role-agent(carpenter, Joe), role-agent(plumber, Tom)}$$
$$\forall\ \textit{role (role-in-plan(role, plan}_1\textit{))}\ \exists\ \textit{agent (role-agent(role,agent))}$$

Here Harry has the goal of designing a kitchen, and has a plan for doing it ($plan_1$ above). He needs a carpenter and a plumber for this task. Harry finds Joe and Tom to fulfill these roles in the plan. Joe and Tom agree to fulfill these roles in Harry's plan. The last statement asserts that Harry has established commitments for all the tasks in $plan_1$, in this case the carpentry and the plumbing.

[2] See [KGE88] for a discussion of the implications of this aspect.

[3] All the effects of an action can not be known before the execution of the action. This is the frame problem in AI.

[4] An agent need not be a single individual. It can be a team, a department, an organization etc.

The agent beliefs that are important for CSCW are: domain goals of agents, plans for these goals, preferences and priorities, roles in group activity, capabilities, expectations, and beliefs about other agents.

Agents in goal-directed cooperative activity develop plans for accomplishing their goals. Their actions are guided by their goals, plans, and the perceived worlds. Planning results in making others responsible for various parts of a plan, based on their capabilities, work load etc. These responsibilities define roles which constitute the structure of the group. Organizational structure may be viewed as a set of relatively static roles for establishing commitments and guiding the communication.

Preferences for a course of action are a result of evaluative processes in cooperative activity, while priorities aid an agent in deciding which action to perform next. Lastly, an agent has expectations about others' actions, and the changes in real world. Expectations arise from a variety of sources - sending a message and expecting a reply, commitments of others etc.

An agent may be aware of goals, plans, roles, capabilities and preferences of others (including the groups an agent is a member of). The exact nature of these shared beliefs depends on the course of the activity itself.

The system sees the agent and the world through the interface. The state of the system is this perceived agent model, real world, and the attribute values of system objects the agent manipulates (e.g. document). The state of the interface is the representation of system state that guides agent's actions.

The notion of state is important, because it represents the past activity and generates expectations about the activities that may occur in that state. These expectations can be used by the system to control its actions and respond appropriately.

2.3 Flow of Action

The structure of generic activities constrains the activity sequences occurring in cooperative environments. This structure enables systems to have expectations about the future activity based on the past (i.e. current state). A representation of the structure of generic activities must capture the causal and temporal dependencies between these activities.

Figure 3 shows the representation for the generic activity for exploring alternative ways of achieving a goal resulting in preferences among these alternatives. Circles represent the state of the activity for an agent and the edges between these circles are generic activities. States for different agents are distinguished. The state of goal directed activity as a whole is the set of states of the activity, one for each agent.

The figure shows that finding alternatives and establishing preferences between them are main components of this activity. An agent may request another (state 1) to find alternatives (the computer system may be viewed as one such agent) and can perform this activity cooperatively. Some activities such as decision making (in state 5 where an agent decides whether to accept a request) are omitted for clarity.

One important aspect of communication activities within these structures are the expectations that arise. In state 3, the request generates an expectation for the requesting agent (edge from state 3 to state 4), and the acceptance fulfills it. These expectation related activities are shown in thick lines and the causal relation between the request and these activities are indicated by an arc between these edges.

Activities in this structure are not atomic. For example, the activity for establishing preferences may be performed in a number of ways. The agents can meet and make this decision by making arguments, deciding evaluation criteria and ranking alternatives[Ste87] or a single

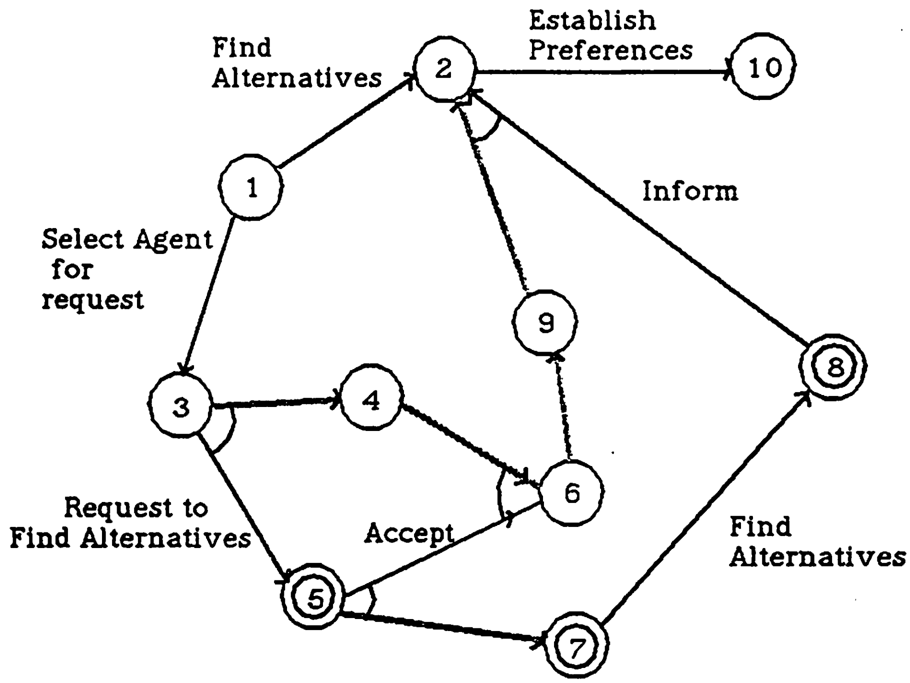

Figure 3: Exploring Alternatives

agent may do this based on his/her knowledge. It is also possible that these alternatives may change during the process of justifying them by arguments.

Using the representation, we now present the hierarchical activity structures for goal based cooperation and plan development. Other activities may be modeled in a similar manner.

Figure 4 shows the structure of a cooperative activity involving initialization, planning, execution and evaluation. Each of the activities have required preconditions and effects that result from its performance. For instance, initialization requires some context (e.g., resource conflict in state 5) and results in the formation of a goal. Similarly planning requires a goal and results in a plan, and may lead to initialization of new goals (such as subgoals of the plan).

The structure for the plan development activity is shown in figure 5. It consists of exploring alternatives, expanding goals into plans, establishing priorities, establishing commitments between agents for parts of the plan, and constraint satisfaction that involves allocation and scheduling of resources for the plan. Agents can begin to explore alternatives for the subgoals in a plan at any point during this process.

The activities in plan development are also structured. The structure for exploration of alternatives was presented in figure 3. Figure 6 shows the structure for establishing commitments.

In this structure, commitments may be established by selecting an agent for a role and requesting the agent to accept the role. This request results in an expectation for the requesting agent about the other agent's response. The arc connecting the edges for these activities in the figure indicates this causal relationship. An agent receiving this request can accept, reject or ask for more information before accepting it.

Any activity may be delegated to other agents or performed by a negotiation. For example, an agent may select another agent for a role by a bidding mechanism. In the generic activity framework, it is the exploration of alternatives for the selection of an agent, and negotiation that results in establishing preferences for the role assignment.

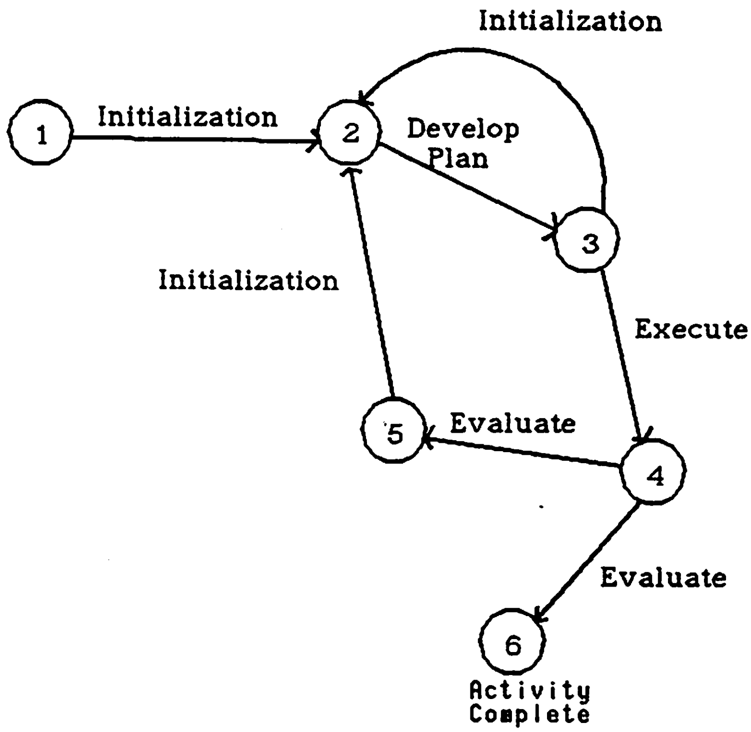

Figure 4: A Structure for Goal-Based Cooperative Activity

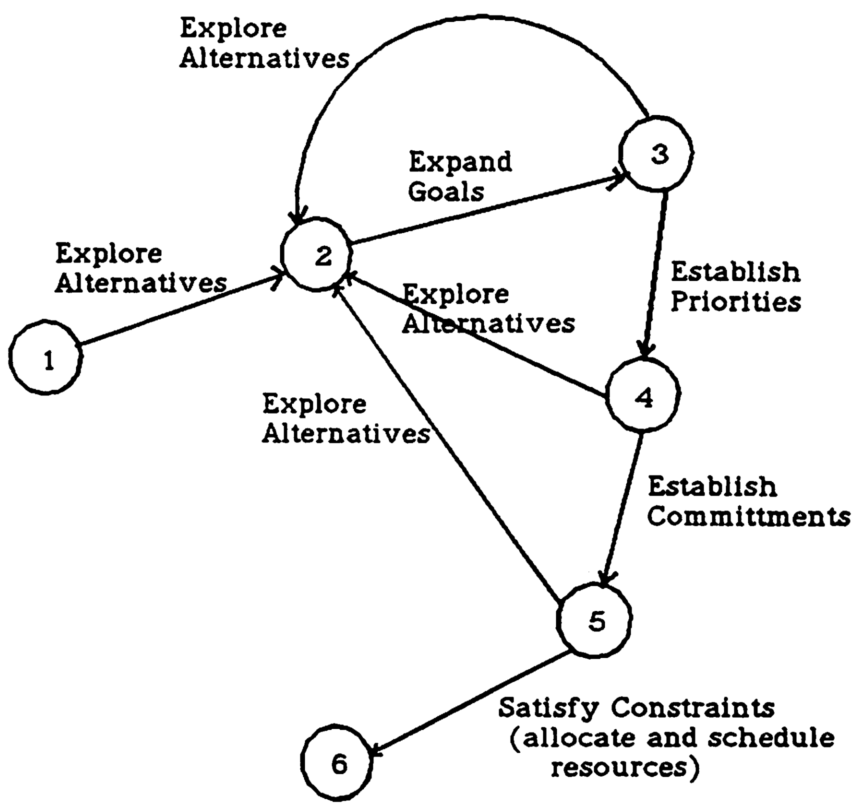

Figure 5: A Structure for Collaborative Plan Development

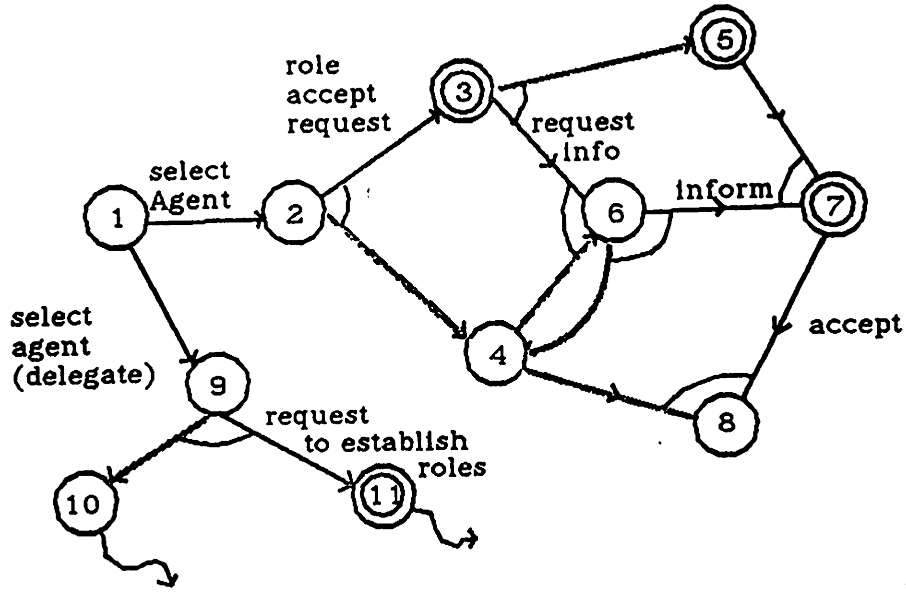

Figure 6: A Structure for Establishing commitments

3 An Example

An example of goal based cooperation between multiple agents (including the computer system) is presented here. We assume a hypothetical computer system whose functionality is based on our theory. The system has knowledge of generic activity structures, domain objects, typical domain activities, and maintains models of human agents (described in section 2). It distinguishes two kinds of objects: those shared among a group and those private to an individual. The structure of the organization in which the agents operate, containing the roles of agents in the organization and their capabilities, is shared. The system uses generic activity structures and an agenda of the current generic activities to coordinate its interaction with the agents.

Tom is a software development manager who wants to buy a bigger monitor for the MicroExplorer (a Lisp machine), as the current one is too small. Figure 7 is a picture of cooperation underlying this seemingly simple activity. The activities are numbered in sequence. System actions are shown in thin lines, while all others are shown in thick lines.

Tom specifies the goal to the system, along with constraints about its size, and the compatibility with the MicroExplorer. Buying is a routine activity in the system, although the evaluation processes within are not. The system places plan development activity for this goal on its agenda, and expands the goal into subgoals from its knowledge of domain activities. These subgoals are: deciding on the specifications for the monitor to be bought, informing the accounting, and placing a purchase order.

Since the first of these is not a routine activity, it places the activity for exploring alternatives for the monitor to be bought on its agenda using the activity structure for plan development (figure 5), and consequently the activity "Find Alternatives" on its agenda using the structure for exploring alternatives (figure 3).

The system object descriptions do not contain any big monitors which are compatible with the MicroExplorer. It then queries Tom whether he wishes to specify alternatives or would like to request someone else for this information. Since Tom does not know any alternatives, he selects the latter option. The system then responds with the organizational structure for Tom to browse (as part of selecting the agent in figure 3).

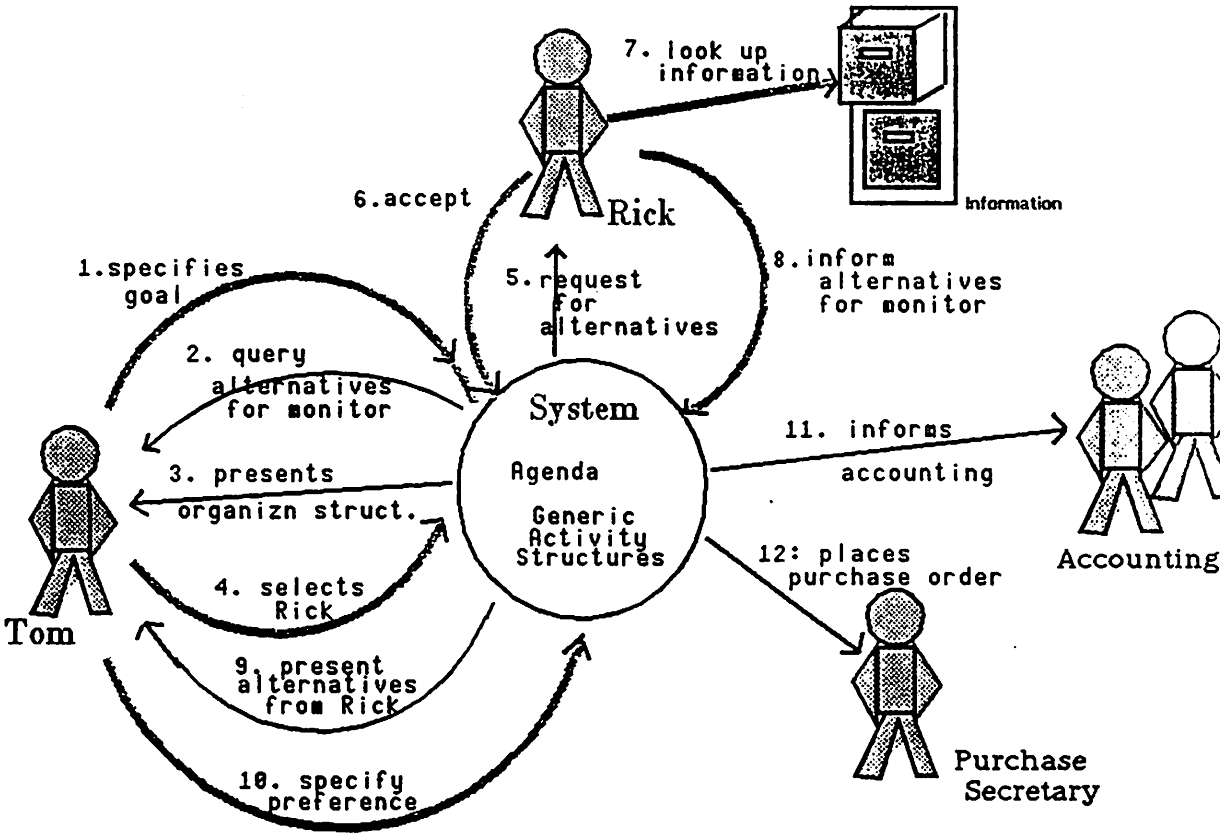

Figure 7: Example

Tom selects Rick from hardware maintenance department, and the system generates a request for Rick to find alternatives for this monitor. This request results in an expectation on the agenda of the system. Rick accepts the request and the system replaces its earlier expectation with another expectation, where it (and Tom) is awaiting alternatives from Rick. At this point the system also places the activity "Find Alternatives" on its agenda, with Rick as the agent for this activity.

Rick knows that there are two monitors "Radius" and "Apple" that satisfy the constraints for the monitor. However he needs to find information about their cost, dealer to contact etc. He then specifies these alternatives to the system which informs Tom of these alternatives (see figure 3). Tom then establishes preferences between these alternatives using some criteria, such as their cost, reliability of the manufacturer etc. Once the monitor specifications are known to the system, it can perform the rest of the activity automatically by sending messages to purchase secretary and accounting.

The above example serves to illustrate that our generic theory may serve as a basis for system design. The theory integrates the problem solving with communication in goal based cooperation and defines a role for the computer system in this activity.

4 Conclusions

The generic theory, presented here, relates the real world and the agents to their activities, taking into account the multiple, concurrent activities of these agents in cooperative environments. We described the structure of these activities to serve as a basis for CSCW system design, and how a system based on this theory could assist agents in coordinating their problem solving and communication. Just as the structure of cooperative work constrains the interactions between people, our theory constrains the functionality of a system and an interface designed to support this activity.

The generic activity structures model communication as part of a task context. This is unlike language/action formalism [FGHW88] which models communication independent of task context. Other theories of decision making have been as a basis for design, but they mainly address the interface issues in specific generic activities such as meetings [Ste87]. Our approach has been to focus on a larger context of which such activities are a part.

We are currently extending the theory to incorporate issues of human intentionality and rationality. A promising approach, combining aspects of information processing psychology [And83] and situated philosophies [Hei27], can be found in the area of action psychology [Ach05,FS85,VW85]. The central element of action psychology is the *Handlung* (engl. enactment), a wholistic view of activities, which sees the individual acting in task and social contexts.

Our generic theory guides the design of a framework for supporting a variety of application domains, into which traditional tools for domain activities may be integrated. In addition, knowledge-based approaches such as planning, may be used for generic activities such as exploration of alternatives for a goal. Such a framework can also integrate a variety of other tools and techniques. For example, meetings are a generic activity for solving some problem (goal). Using the task context, the framework can assist in tasks such as scheduling meetings. As another example, the framework can suggest agents for a role or finding out some information by using a set of heuristics (rather than simply presenting the organizational structure to the user as described in section 3).

Using this theory, we are currently developing a distributed CSCW system (D-POLYMER) [BC90] at our Collaborative Systems Laboratory. This system integrates knowledge-based planning, with the communication using the generic activity structures. It is based on POLYMER, a knowledge-based planning system [LC89] that supports planning and execution monitoring in underspecified domains. The POLYMER system is fully implemented.

The D-POLYMER framework is intended to support the functionality of a collaborative project management application. Traditional project management support provides a single user with tools for planning, scheduling, and resource allocation, but little support for the cooperation within the activity. A recent study [PS90] at our laboratory shows that actual project management transcends this narrow view, involving input from all the project members through the stages of planning the project to its performance and evaluation. Managers who were interviewed for this study expressed a strong need for reuse of project plans across projects, access and maintenance of organizational resource libraries, integration of plans developed by different people, and communication within the tool.

Although our theory aids in the design of systems for supporting goal-based cooperation, it can not predict the effects of a system based on this theory on the relations between people or organizational changes that come about by its introduction. However, since agents using the system make the goals for their interactions explicit, it can lead to better problem solving and decision making. We view this theory as a hypothesis that must be validated in social context, by evaluating systems based on this theory in a cooperative environment.

References

[Ach05] N. Ach. *Ueber die Willenstaetigkeit und das Denken.* Vandenhoeck, 1905.

[And83] J. R. Anderson. *Architecture of Cognition.* Harvard University Press, 1983.

[BC90] Nehru Bhandaru and W. Bruce Croft. An Architecture for Supporting Goal-Based Cooperative Work. To appear as COINS-CSL Technical Report, February 1990.

[Boe80] E. Boesch. *Kultur und Handlung.* Huber, Bern, 1980.

[FGHW88] F. Flores, M. Graves, B. Hartfield, and T. Winograd. Computer systems and the design of organizational interaction. *Transactions of Office Information Systems*, 6(2), 1988.

[Fik82] R. E. Fikes. A Committment-based Framework for Describing Informal Cooperative Work. *Cognitive Science*, 6:331–347, 1982.

[Fox81] Mark S. Fox. An Organizational View of Distributed Systems. *IEEE Transactions on Systems, Man and Cybernetics*, 11:70–80, 1981.

[FS85] M. Frese and J. Sabini, editors. *Goal Directed Behavior: The Concept of Action in Psychology.* Lawrence Erlbaum Associates, 1985.

[Gre88] I. Greiff, editor. *Computer Supported Cooperative Work: A Book of Readings.* Morgan Kaufmann, 1988.

[Hei27] M. Heidegger. Sein und Zeit. *Jahrbuch fuer Philosophie und Phaenomenologische Forschung*, 8, 1927.

[KGE88] R. Kraut, J. Galegher, and C. Edigo. Relationships and tasks in scientific research collaborations. *Human Computer Interaction*, 3(1), 1988.

[Lan87] T. K. Landauer. Relations between cognitive psychology and computer system design. In J. Carroll, editor, *Interfacing Thought.* MIT Press, 1987.

[LC89] L. S. Lefkowitz and W. B. Croft. Planning and execution of tasks in cooperative work environments. In *Proceedings of the Fifth IEEE Conference on Artificial Intelligence Applications.* IEEE, 1989.

[Mac69] R. Alec Mackenzie. The Management Process in 3-D. *Harvard Business Review*, November 1969.

[Mah90] D. E. Mahling. *A Visual Language for Knowledge Acquisition, Display and Animation by Domain Experts and Novices.* PhD thesis, University of Massachusetts, February 1990.

[MCC90] D.E. Mahling, B.G. Coury, and W.B. Croft. User models in cooperative task-oriented environments. In *Proceedings of the 23rd Hawaii International Conference on Systems Science.* IEEE, 1990.

[MGL+87] T. W. Malone, K. R. Grant, K-Y. Lay, R. Rao, and D. Rosenblitt. Semi-structured messages are suprisingly useful for computer-supported coordination. *ACM Transactions on Office Information Systems*, 5(2), 1987.

[ND86] D. Norman and S. Draper. *User Centered System Design.* Lawrence Erlbaum, 1986.

[PS90] C. Pietras and O. Sandvik. A review of project management tasks and tools. Technical Report CSL-90-1, University of Massachusetts, Collaborative Systems Laboratory, 1990.

[Ras86a] J. Rasmussen. A framework for the cognitive task analysis in systems design. In E. Hollnagel, G. Mancini, and D.D. Woods, editors, *Intelligent decision support in process environments*, pages 175–210. Springer Verlag, 1986.

[Ras86b] J. Rasmussen. *Information Processing and Human-Machine Interaction.* Systems Science and Engineering. North-Holland, 1986.

[SC88] B. Shneiderman and J. M. Carroll. Ecological studies of professional programmers: An overview. *Communications of the ACM*, 31(11):1256 – 1259, 1988.

[SG85] S. Sarin and I. Greiff. Computer-based real-time conferencing systems. *IEEE Computer*, 18(10), 1985.

[Ste87] Stefik, M., Foster, G., Bobrow, D., Kahn, K., Lanning, S., and Suchman, L. Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meetings. *CACM*, 30(1):32–47, 1987.

[Suc85] L. A. Suchman. Plans and situated actions: The problem of human-machine communiation. Technical Report isl-6, Xerox Corporation, 1985.

[VR84] C.R. Vick and C.V. Ramamoorthy, editors. *Handbook of Software Engineering.* Electrical/Computer Science and Engineering Series. Van Noostrand Reinhold, New York, 1984.

[VW85] R. R. Vallacher and D. M. Wegner, editors. *A theory of action identification.* Lawrence Erlbaum Associates, 1985.

[WF86] T. Winograd and F. Flores. *Understanding Computers and Cognition.* Ablex, 1986.

[WW87] J. Whiteside and D. Wixon. Discussion: Improving human computer interaction - a quest for cognitive science. In J. Carroll, editor, *Interfacing Thought.* MIT Press, 1987.