# i-AGENTS: Modeling Organizational Problem Solving in Multi-Agent Teams

Yan Jin

Department of Civil Engineering, Stanford University, Stanford, CA 94305-4020, USA

Raymond E. Levitt

Department of Civil Engineering and Center for Integrated Facility Engineering (CIFE), Stanford University, Stanford, CA 94305-4020, USA

**Abstract** **Organizations involving human and computer agents are constrained by a variety of factors including: task properties and arrangements; level of technology; knowledge held by, and distributed among, the agents; information and administrative structures; and organizational norms and policies. An important challenge to the scientific community is to develop, validate and apply theories and models to help managers re-engineer their organizations for higher levels of performance. Our research on organizational problem solving aims to develop a computational model of organizations to study interrelationships between agents' knowledge, task requirements, and organization structures and policies. This paper reports the first step of our research toward a computational organizational model—the i-AGENTS framework, a prototype computer system for modeling organizations of intelligent agents. i-AGENTS is composed of a number of high-level concepts: tasks, agents, organization and communication. A task is described in detail by task action, task object and task constraints; an agent is modeled to consist of cognitive attributes and expertise; role-based organizational structure is adopted for describing organizations. From an organization perspective, i-AGENTS extends traditional information processing models of organization (Galbraith 1977) by explicitly addressing the role of agents' knowledge of both the problem domain and the organization in problem solving. When viewed from an engineering perspective, our research is the first step toward an organizational problem-solving model that merges organization theory and distributed artificial intelligence and can be used to simulate and analyze organizational behavior of teams in engineering domains at a very specific level of detail.**

## INTRODUCTION

The performance of organizations in engineering domains can be affected by a variety of factors, including organizational task requirements, level of technology, agents' knowledge, administration and information structures, and organizational norms and policies. Although research to date in organization theory, distributed artificial intelligence and concurrent engineering has addressed some issues related to these factors, little work has been done to study systematically the relations between the factors, with respect to the organizational performance in the engineering domain, or to provide tools for organizational design. We believe that finding ways to design engineering organizations systematically has become critical

CCC 1055-615X/93/040247-24
© 1993 by John Wiley & Sons, Ltd.

*Received March 1993*
*Revised October 1993*

for the competitiveness of industrial firms, which operate in increasingly competitive global markets and rapidly changing technological environments.

Organization theory addresses issues in the design of human organizations qualitatively, and at a level of aggregation that abstracts most of the content from agents, tasks, and decision-making processes (March and Simon, 1958; Thompson, 1967; Galbraith, 1977; Cohen *et al.*, 1972; Hannan and Freeman, 1977; Pfeffer and Salancik, 1978; Malone, 1987). The kinds of organizational changes that are of interest to managers include the ways in which specific kinds of information and knowledge are shared by agents, the training of agents, alternative task decompositions, and the introduction of new technologies for information processing and communication (Tatum, 1984). However, the level of abstraction of agents and tasks in extant organization theory makes it impossible for this theory to predict the impact of such detailed organizational changes on organizational performance. We argue that the reason traditional organization theory remains highly abstract is that the sociologists who have dominated research in this field focus on performance at group, subunit or firm levels; and the reason it has little predictive power is that they have tended to use natural language to describe their theories, rather than more formal mathematical or computational models. Accordingly, the theory can only posit qualitative relationships among subunits or firms in terms of nominal—or, at most, ordinal—variables. There is a small but growing trend to exploit computational modeling to operationalize, quantify and simulate theories of the detailed behavior of subunits and even individual workers in organizations (Cyert and March, 1963; Cohen *et al.*, 1972; Masuch and La Potin, 1989; Carley *et al.*, 1992; Cohen, 1992). We believe that such computational approaches open the way to addressing organizational design at very specific and detailed levels.

Distributed artificial intelligence (DAI) and related research is concerned with building multiple and distributed intelligent systems to solve temporally, spatially and functionally distributed problems (Lesser and Corkhill 1981; Fox 1981; Davis, 1982; Bond and Gasser, 1988; Huhns, 1987; Gasser and Huhns, 1989; Durfee *et al.*, 1989). It emphasizes detailed descriptions of agents and tasks, and the knowledge and reasoning methods used by agents. However, DAI research rarely takes into account the organizational and social aspects of multi-agent systems and lacks a theory to link organizational design with the performance of multi-agent systems for given environmental situations. We argue that although ideas and methods emerging from DAI research may be effective to solve those problems that can be viewed as purely technical—e.g. distributed monitoring systems—their usefulness is limited when they are applied in a social context in which people are involved, computer systems are frequently updated and used for different purposes by different people in different organizational structures, and tasks and environment are changing.

Our research on organizational problem solving is concerned with how a group of intelligent agents, including human and computer systems, can be organized in ways that match the group with its task and environment to achieve effective and efficient organizational performance. The primary goal of our research is to understand and then model organizational mechanisms in analysis tools that can be used to engineer multi-disciplinary organizations and to co-ordinate intelligent computer systems. Real problems that suit this approach include project management, enterprise integration, CSCW (computer-supported co-operative work) and concurrent engineering. Our research investigates the area of overlap between organization theory and distributed AI, and adopts ideas and methodology from both. In comparison with organization theory that abstracts most of the content from agents, tasks and decision making, our research emphasizes the details of agents, tasks and specific problem-solving processes. Unlike DAI, which emphasizes the knowledge and reasoning methods for agents to interact with each other, and which is concerned with building functioning multi-agent systems, we put more emphasis on organizational mechanisms, and are concerned with how to organize multiple agents, including human and computer systems, to achieve better performance. Our research is particularly focused on the engineering domain. Our long-run objective is to develop a computational organization tool for modeling engineering organizations and investigating the impact of organizational knowledge, structure and policy on organizational performance.

In this paper we describe the first step of our research toward a computational model of engineering organizations—the i-AGENTS framework. The next section provides an overview of i-AGENTS. In the third to fifth sections we discuss in detail the concepts of **tasks**, **agents** and **organizations**, respectively, and describe how those concepts are elaborated and represented in i-AGENTS. The sixth section describes the implementation of i-AGENTS and presents results from an initial simulation. The seventh section compares our research with related work and the final section presents our conclusions and our plans for future research.

## i-AGENTS FRAMEWORK

i-AGENTS is a computerized framework for studying organizational problem solving in multi-agent teams. Before going into the details of its representation and reasoning, we define the problem and derive the conceptual requirements for the framework.

### Multi-agent Teams and Organizational Problem Solving

In our research **a multi-agent team** is defined as a project team in which there is at least one **task** and a number of decision makers called **agents** who can not only make decisions for their local problem solving but can also communicate with others using specific **protocols**, and collaborate with each other to solve relatively complex common problems. For example, in the facility engineering domain a multi-agent team for a building project is composed of the building project (i.e. the task) and a number of design agents, including architects, structural designers, heating, ventilation and air conditioning (HVAC) designers, and construction managers. The **goal** of the team is to build a building that satisfies various requirements and limitations on total duration and costs. The **tasks** of the team include feasibility study, conceptual design, structural design, project planning, construction, etc. Within a given team there can be sub-teams at lower levels. In the above example, the construction manager can be a representative of a multi-contractor team including carpenters, plumbers, painters and electricians.

An agent in a multi-agent team can be either a **computer decision-making system** (CDMS) working independently or a **computer decision-support system** (CDSS) coupled with a human decision maker. A CDSS can support a human decision maker by providing a communication channel to the team environment and by helping the human decision maker to make decisions. Since our concern in this research is focused on the co-ordination issue in organizational problem solving, we need a uniform agent description, emphasizing both technical and social aspects of the team agents.

In a multi-agent team, **organizational problem solving** is a **distributed** and **concurrent** decision-making process in which multiple agents that are required to solve a common problem work together. The distribution of decision making may be along different dimensions, such as temporal, spatial, functional, etc. For example, in the design team described above, the organizational problem solving is a concurrent design process in which architects, structural designers, HVAC designers and construction managers work in parallel and pass information to each other when necessary. In this case, the multi-disciplinary designers can be distributed in different locations while connected in a network. Concurrent design has the potential to save development time, and thus permit the consideration of more alternative design syntheses and save cost. It may, however, run the risk of deficient, redundant or inconsistent decision-making efforts if there is an inadequate **co-ordination** between design team participants (Levitt *et al.*, 1991). Thus how to achieve appropriate co-ordination for better organizational performance is a key problem.

From the above descriptions it is clear that in order to model organizational problem solving in multi-agent teams we need to incorporate and elaborate the following concepts:

- **Tasks**: An organizational task specifies the work to be carried out by a multi-agent team. A task can be a complex real task or a simple 'toy' task. The task description can be abstract or detailed. In i-AGENTS, tasks describe the workload for an agent organization and create a problem space in which agents solve their own problems, determine what information to exchange, identify potential conflicts and co-ordinate their activities. Task representation is important for organizational problem solving in multi-agent teams.

- **Agents and organizations**: Agents are decision makers that work together to identify tasks, solve problems and accomplish the project goals. Agents are **autonomous**, their activities do not require constant guidance or intervention from others. Agents are **intelligent** in the sense that they hold interests and beliefs, and can make rational choices in different problem-solving situations. Agents, however, are also **boundedly rational** (March and Simon, 1958; Simon, 1976), since their capability, capacity and cognitive processes are limited. Agents are **social** because they are always situated in a specific social position, formal and informal. In order to represent both formal and informal social relationships between agents we need a model of organization to describe groups of agents.

- **Communication protocol**: Although agents may interact with each other through shared memories, we assume that communication through message-passing is the basic means for agents to interact with each other. Because our concern is focused on how to organize agents for better performance, we will use a simplified version of KQML and KIF (Genesereth *et al.*, 1992) for agents to communicate.

- **Distributed problem solving and co-ordination**: As described above, organizational problem solving is distributed and requires co-ordination. Although developing mechanisms for distributed problem solving and co-ordination is important, we are more concerned with how to **organize** agents, given a set of tasks and agents. In order to investigate the impact of organizational mechanisms on organizational performance we need a computational framework that allows us to implement alternative organization and co-ordination mechanisms in different task and team situations. We expect that, as our research evolves, we will identify additional variables for distributed problem solving and co-ordination from an organizational perspective.

## Co-ordination

Co-ordination in general, is the act of managing interdependencies between activities (Malone and Crowston, 1991). In our research, co-ordination is viewed as the effort to manage interactions among multiple agents performing some collective task. Our hypothesis is that better co-ordination may help agents avoid **unnecessary** activities and consequently increase the efficiency of organizational performance.

In i-AGENTS we define the notion of **co-ordination scheme**[1] as a basic concept to represent various possible ways of co-ordination. A co-ordination scheme is characterized by a number of attributes, where different co-ordination schemes can be described using the same attributes but different attribute values. A co-ordination scheme represents a normative structure for the organization of agents; that is, it specifies or constrains organizational behavior. It also represents a mechanism that governs the behavior of agents. In this research we try to explore the relationships between various organizational mechanisms and the behavior of the organizational agents. The following is a brief description of the main attributes of a co-ordination scheme.

### *Knowledge Structure*

Knowledge structure specifies the distribution of knowledge among agents and may be described in terms of coverage, centrality and redundancy:

- **Coverage** specifies how much of the knowledge required for solving the problems

[1] The term **co-ordination scheme** is adopted from Carley *et al.* (1992) but we have a somewhat different definition.

comprising a given task is covered by the knowledge held by agents of the team. Generally, full coverage is required for a multi-agent team to perform a task well (Corkill and Lesser, 1983).

- **Centrality** describes how knowledge is distributed among multiple agents. In some cases, a team has only a few expert agents which hold knowledge over a wide range of the domain. In other cases a team may have many agents each of which can only solve a small part of the whole problem.

- **Redundancy** describes the degree to which knowledge is shared in multi-agent teams. In order to co-ordinate their interdependent activities agents need to share knowledge to a certain level. In most cases, higher knowledge redundancy implies robustness of an organization (Hewitt and Inman, 1991).

Understanding the interplay between knowledge structure and the organizational performance of multi-agent teams has important implications for design of a project team, as well as for training of its human workforce.

### *Information Structure*

Information structure specifies how information flows among information sources, which can be agents or databases. It is described by two substructures, the communication structure and the access structure:

- **Communication structure** specifies who can talk to whom in the team through message-passing. Two extreme cases are (1) there is no message-passing and agents communicate through shared memory (if there is any) versus (2) any agent can talk to any other agent.

- **Access structure** specifies who can directly access which information. High-level managers may have different information access privileges than lower-level designers. The information structure can be static or dynamic during organizational problem solving (Carley *et al.*, 1992).

### *Administrative Structure*

Administrative structure defines the authority or control relationships between agents in a team. Possible structures may range from completely flat to a strict hierarchy. Administrative structure has always been an important way for human organizations to co-ordinate their activities (Galbraith, 1977; Chandler and Dames, 1980). We believe that this insight applies to our multi-agent teams that involve both people and computer systems.

### *Social Regulations*

Social regulations specify values, norms and policies of the society of agents. **Values** are the criteria employed in selecting the goals of behavior; **norms** are the generalized rules governing behavior that specify, in particular, appropriate means for pursuing goals; and **policies** specify common tactics for agents to interact with each other, and, in particular, restrict the behavior of agents in negotiation. For example, a policy for negotiation between agents may be that whenever a dispute between two agents occurs, it is passed to the closest agent at a higher administrative level. Another policy can be for the agents to suggest solution alternatives themselves.

Figure 1 illustrates the concepts of the i-AGENTS framework discussed above. In the following sections we elaborate the concepts shown in this figure, and describe how they are represented and employed for studying organizational problem solving in multi-agent teams.

## TASKS FOR MULTI-AGENT TEAMS

A task in a real organization describes the work to be done. In a construction organization, for example, a task is defined by a contract and the associated design documents (drawings and specifications) of a project building. For research on organizational problem solving, either a simple task (e.g. blocks world) or a complex one (e.g. building design and construction) can be chosen, and the task can be represented at different levels of detail. The choice of task complexity and abstraction level will affect the observable organizational behavior that emerges from simulations. Thus,

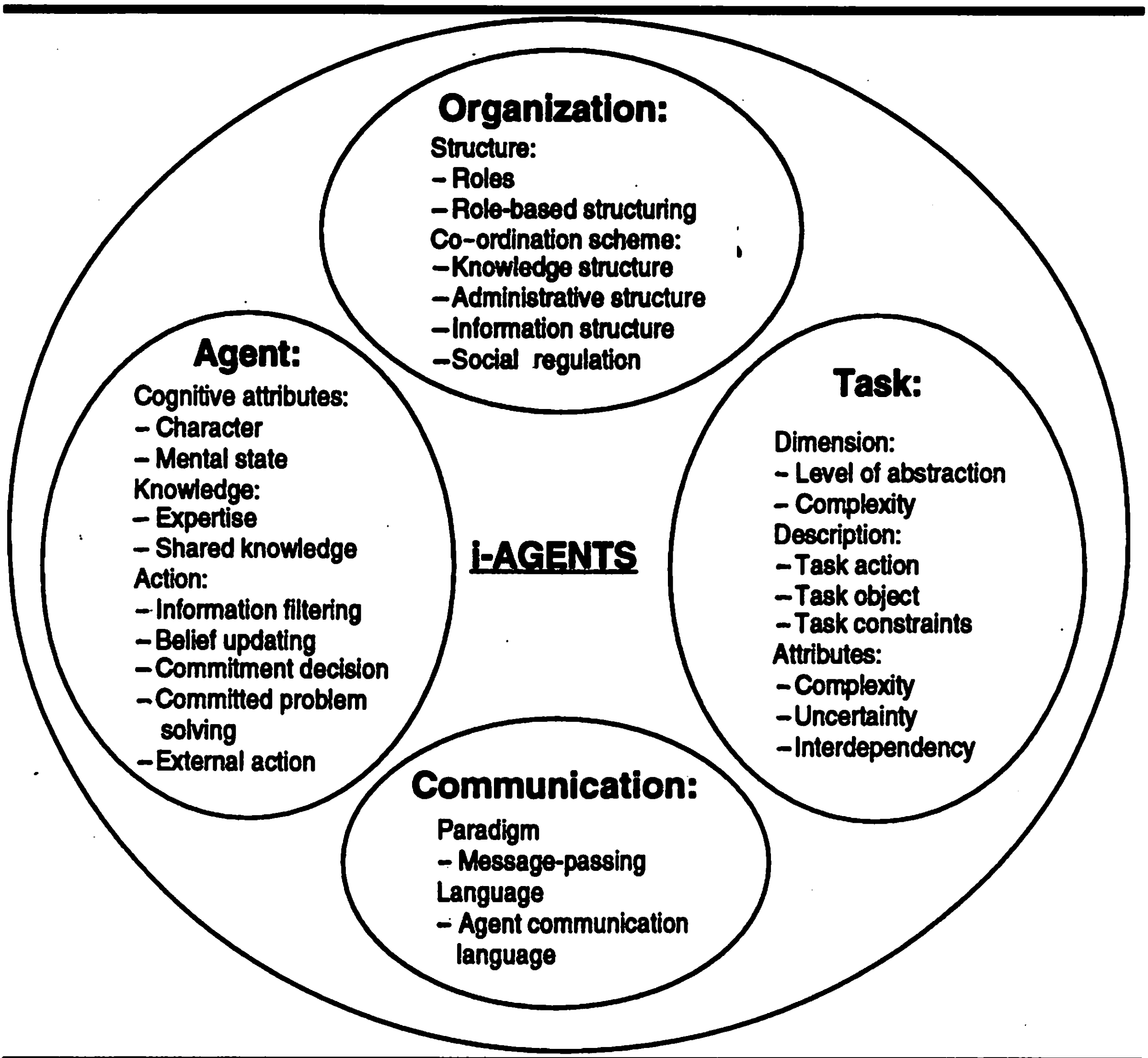


**Figure 1** High-level concepts used in the i-AGENTS framework

the task representation should be capable of describing tasks at different levels of detail and capturing both general and specific task properties.

### Task Abstraction and Complexity

Although early organization theory began with the systematic study of tasks (Taylor, 1911; Fayol, 1949), in much of current organization theory, the task is not explicitly articulated, or is overlooked (Perrow, 1967; Mackenzie, 1978, Carley *et al.*, 1992). Organization theorists, especially those who take a structural approach, have been looking at the general features of tasks and have tried to relate organizational design to general task properties such as complexity, uncertainty and interdependence. For example, in the contingency theory approach to organization structuring, task uncertainty has been recognized as a primary factor in choosing an appropriate organization (Galbraith, 1977). Transaction cost analysis also focuses on uncertainty in transactions as a determinant of organizational structure (Williamson, 1979).

DAI is concerned with the detailed problem-solving process of intelligent agents solving problems through interaction and co-ordination. How to describe, decompose and distribute an organizational task is one of the central issues in DAI research. It can be argued that most DAI research to date has focused on specific tasks, and different distributed problem-solving approaches are developed against those specific tasks. Thus, a task in a DAI system defines, in detail, the problem space in which agents solve problems collaboratively.

In summary, tasks in organization theory and DAI represent two extremes of the task-abstraction spectrum. On the one hand, organization theory abstracts most of the content from the task and retains only general properties for analyzing organizational performance. On the other, DAI tasks describe very detailed requirements, goals to be achieved, and associated constraints. The agents who work on the task have to identify problems and solutions that satisfy the requirements.

Complexity is another dimension of task description. By complexity of a task at a given level of detail, we mean the amount of knowledge that must be brought to bear for agents to carry out the tasks. From an engineering point of view, the more complex a task, the more realistic it is. For organizational problem-solving research, complex tasks reflect the real task and environment requirements, but constrain organizational performance and make it specific to the task as well. Choosing tasks of appropriate complexity and describing tasks at appropriate abstraction level is important. Table 1 compares abstract versus more detailed task descriptions from a computer-modeling perspective.

## Tasks in i-AGENTS

Since the objective of our research is to develop a computational organization model for investigating the impact of individual knowledge and organizational structure and policy on organizational performance and for designing engineering organizations, a **task** in i-AGENTS is described as a real engineering problem for which the goal needs to be achieved by multiple (and possibly multi-disciplinary) agents working together. We argue that detailed task specification is an important factor that contributes to our ability to model the organizational performance of multi-agent teams in engineering domains and that two conditions must be satisfied to explicate the contribution. First, the task description should be detailed enough that we can assess the impact of agents' knowledge and understand the interplay between task, organizational design, and levels of cognition of agents. Second, the task should be real, or complex, enough—rather than simplistic like a block stacking task—so that the engineering requirements can be reflected and the framework can be tested and validated in real engineering domains.

Based on the above considerations, our objective of task modeling is to identify elements and structures that are powerful enough to describe relatively complex tasks in detail and sufficiently robust to represent different types of tasks found in engineering domains. Our task model follows the insights gained from our previous work on project planning (Darwiche *et al.*, 1989; Jin and Levitt, 1993). The following paragraphs describe the elements of i-AGENTS's task model and general task properties.

- **Task**: A task in i-AGENTS is described in terms of **task action, task object,** and **task constraints**. For example, a construction task can be described as (Build, Smith-House, Within $500,000 and 6 months). It is obvious that the pair of **task action** and **task object**, i.e. Build Smith-House for the above example, represents the goal of the task. In our research we adopt the Set-based Recursive Decomposition (SRD) model of engineering design (Chen and Ward, 1991) for task description. That is, we view the task as a set of operations, decompose the overall task into several smaller sub-tasks, solve the decomposed sub-tasks, and recompose the local decisions into larger scale of task solutions. The top-level task, called **project**, is given to the representative of a multi-agent team by a client. Other tasks or sub-tasks are generated through a task-decomposition process.
- **Task Action**: Task action describes the operation required to accomplish the task. Engineering actions can be design, plan, install, paint, etc. For a given engineering domain, there exists a set of actions and their interrelationships. We call this action set an **action model** for that domain. From an organizational

**Table 1** Some properties of tasks descriptions at different level of details

| | More abstract | More detailed |
|---|---|---|
| Contents | **Process-oriented:** the task processing including task decomposition, distribution and task interaction must be included in the model. Solution of a task is not interesting. | **Product-oriented:** the goal, the associated requirements and constraints are given, but the task process is left to agents to resolve. Solution of tasks is important. |
| Randomness | **Random:** use probabilistic parameters to describe task properties and processes. | **More deterministic:** task description and process are deterministic, though arrival times of task may be random. |
| Agent description | **Simple and behavioral:** agent's capability and preference are described by high-level behavioral parameters. | **Sophisticated and cognitive:** agents are described in terms of knowledge and are capable of identifying (preference) and solving (capability) problems. Agents' behavior emerges from simulations. |
| Task scale | **Real and complex tasks:** abstract description makes it easy to describe large-scale realistic tasks based on a number of task process assumption. | **Simple tasks:** detailed description makes it difficult to represent real and complex tasks due to the limitation of current modeling technology. |
| Observable organizational performance | **Information processing features:** explicate the impact of organizational structuring, communication pattern and tool usage, etc. | **Social cognitive features:** explicate the impact of agent (organization) cognition and knowledge, organizational structure, norms, and policies, etc. |

problem-solving point of view, task action specifies the capability requirements of agents who work on the task. Explicitly representing task action provides an important dimension for task decomposition and distribution (Jin *et al.*, 1992).

- **Task Object**: The task object describes the focus of attention for an agent in executing the task. It may be a piece of hardware or software, or it can be a plan, as typified by a drawing, or an event such as a meeting. Whatever the specifics, the task object is to be 'engineered' within constraints, with resources, and by means of defined mechanisms to produce an 'optimal' system (i.e. task object) performance. From an organizational problem-solving point of view, the task object corresponds to the domain of interest of an organization, or an individual agent, and its explicit description makes it possible to address task decomposition and distribution along the object dimension. The interrelationships between task objects impose the task relations described below. All the task objects of an engineering domain can be collectively represented in an object model specific to that domain. (Jin *et al.*, 1992).

- **Task Constraints**: Task constraints describe temporal, spatial and process constraints which must be met when the task is carried out. **Temporal** constraints specify time requirements such as start and finish time of a task. **Spatial** constraints specify geometric location and space requirements. **Process** constraints are represented as task relations described below.

- **Task Relations**: Task relations are important task constraints and define, in part, the way in which tasks should be processed. In i-AGENTS, there are two types of task relations, i.e. **elaboration** relation and **interdependency** relations. If a task T is decomposed into t1, t2 and t3, we say that the T is **elaborated** by t1, t2 and t3. The dependency relations include **precedence** relations e.g. t1's successor is t2, and **information** relations, e.g. t2 needs information from t3. These interdependency relations fall into categories of **sequential** and **reciprocal interdependence** defined by Thompson (1967).

Given the representation elements of our task model, the general properties of a task can be described as follows:

- **Task complexity**: A task is more complex if it involves a complex task object which has more sub-components that are related with each other in a more complex way.

- **Task uncertainty**: A task is more uncertain if it provides less information (such as action model, object model and constraints) for task processsing.

- **Task interdependency**: A task is higher in interdependency if there are more inter-relationships between task objects and between task constraints of its sub-tasks.

A task example (Koo, 1987) described using the above task model is illustrated in Figure 2. In this example, an owner of a house wanted to have an extra bathroom for the guest room. An architect designed the bathroom, and they agreed on the layout as shown in Figure 2(a). Obviously, this project involves many task objects such as frames, walls and stalls etc. and task actions such as plan and install. We assume that the footings and the floor slab have been constructed already. The drainage pipe for the shower stall has also been installed. All the materials are available near the site. The existing wall in the guest room has been opened up and is ready for framing of the bathroom. The project then can be described as follows:

- Carpentry: install
  a frame, interior partition walls, exterior partition walls, floor tiles
- Plumbing: install
  pipes that run within the interior and exterior walls, a shower stall, plumbing fixtures
- Electricity: install
  wires that run within the walls, including a switch box, a light switch, a light
- Paint job: paint
  paint on the interior walls.

Figure 2(d) provides a full illustration of the task model represented using the concepts described above. Figures 2(b) and 2(c) are the object model and action model, respectively, used to support the task description.

## AGENTS

As described above, agents are decision makers that work together to identify and solve tasks. How to model agents is a key issue in modeling organizational problem solving in multi-agent teams. The research to date has resulted in many agent models that differ in the problems they try to solve and in the perspectives they take. Some consider agents as expert systems or tools coupled with communication capability (Genesereth, 1991), and others describe agents in terms of knowledge and mental states (Shoham, 1990). In our research on Virtual Design Team (VDT), an agent's capability and preference is defined by a number of behavioral parameters (Cohen, 1992; Levitt *et al.*, 1992; Christiansen, 1993; Jin *et al.*, 1994). Some decision-support system researchers are interested in the behavior of agents and model the agents in terms of stochastic processes rather than knowledge (Miao *et al.*, 1992).

Since our research is concerned with coordination of interdependence in multi-agent teams, and one of our objectives is to explore the relationships between agent knowledge and organizational performance, our agent model is based on knowledge. Following Shoham (1990), we describe agents in terms of **cognitive attributes** that constitute the agent's behavior basis, and **expertise** that provides the source of engineering knowledge of agents.

### A Case Example

To present the requirement of our agent model we first discuss a case example of building a construction plan for the bathroom project described above.

The owner of the house hired a general contractor GC to construct the bathroom. GC hired five subcontractors including: two carpenters CA1 and CA2, a plumber PL1, an electrician EL1, and a painter PA1. The contractor and subcontractors have their own value system for selecting goals, e.g. for

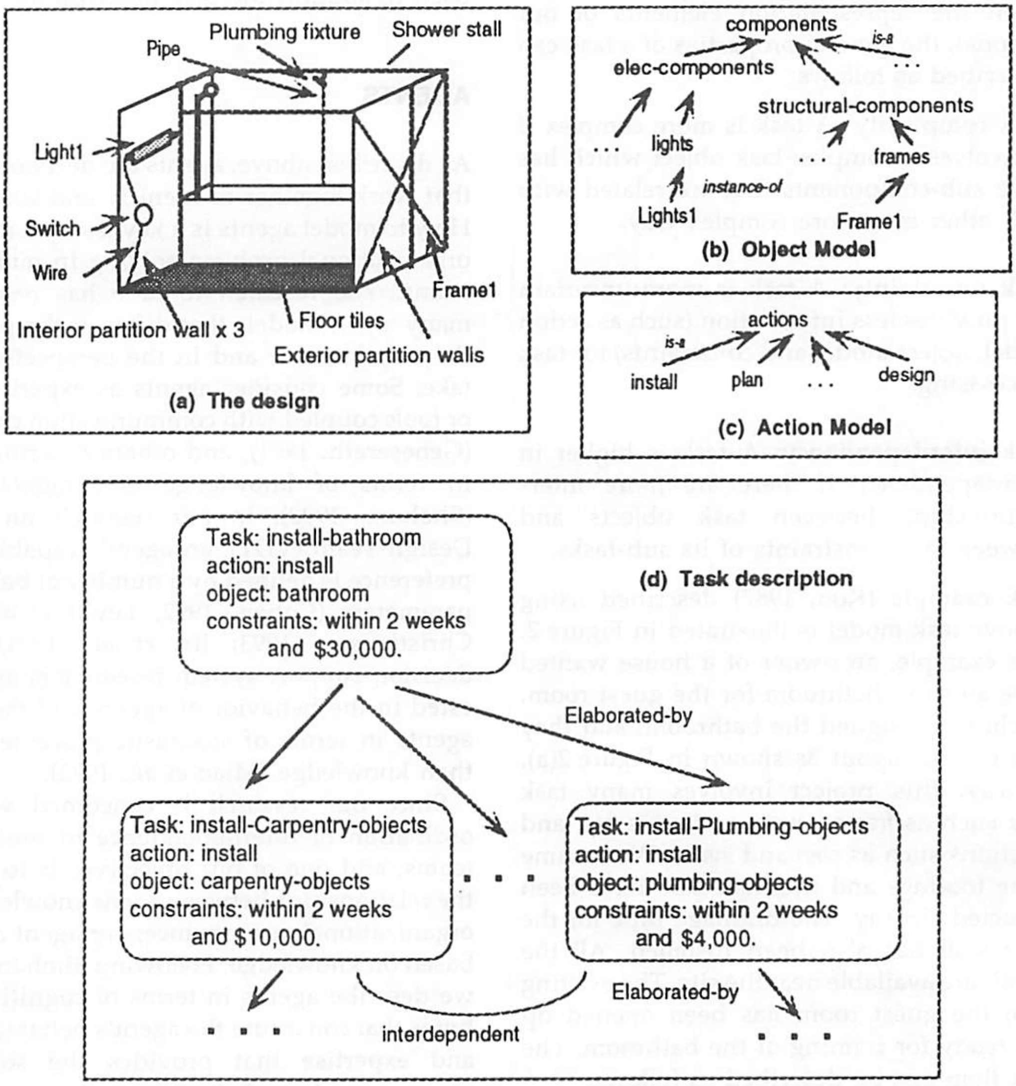


**Figure 2** Task example: a bathroom project

profit, or for establishing prestige, etc., and they have their own domains of interest, capabilities and expertise that support these capabilities. Also, their resources are limited (e.g. time and construction tools). We assume that the contractors can communicate with each other through a computer communication network. They may exchange information, requests and replies through message-passing over the network. The task for the team is (1) to create a construction plan and (2) to construct and install the related objects (components) in the order specified by the plan. In this example, we will examine only the planning process as an exercise in organizational co-ordination.

The goal of the team is to accomplish the task listed above within a given time. To do so, the contractors have to create action plans to achieve this goal. There are two kinds of co-ordination tasks involved in the concurrent planning: The first is task distribution— who should do what? The second is interdependency resolution—what activities should be finished before a certain activity can start? In the following, we summarize a scenario of both local problem solving of GC and subcontractors and interactions between them.[2]

- The house owner sends to GC a message requesting GC to accomplish the bathroom project within two weeks; GC evaluates the project and commits to do so.
- GC elaborates the top-level task (construct the bathroom) into lower-level sub-tasks including carpentry-task, electricity-task, plumbing-task, and painting-tasks, using his own knowledge of construction. Because GC knows of the subcontractors CA1, CA2, EL1, PL1, PA1 (e.g. their domain of interest, and their capability), he requests CA1, CA2, EL1, PL1, PA1 to perform carpentry-task, electricity-task, plumbing-task, and painting-task, respectively, rather than just broadcasting all the task requests.
- Upon receiving the request from GC, CA1 examines the task and finds that it matches her interests and that she is capable of carrying out the task herself. After checking the time schedule and the task constraint, CA1 decides to commit to the task and sends GC a message about her commitment. GC updates his belief by adding that CA1 has been commited to work on carpentry work for the project.

...

- GC updates his belief by adding that PA1 is committed to the painting-task. So far, GC believes that CA1, EL1, PL1 and PA1 have been committed to the requested sub-tasks.

...

- When establishing a plan for installation of lamps and switches, EL1 realizes that in order to set up the switch, the interior wall has to be erected. Because EL1 has no idea of who, besides himself and GC, is involved in the project, he requests GC to tell him who will handle the interior wall. GC tells EL1 that CA1 has been committed to install the wall.
- EL1 receives the name and address of CA1 from GC, and sends a message to CA1 requesting that CA1 finishes installing interior wall before EL1 can set up the switch. CA1 examines the plan herself and replies that she will definitely install the wall by tomorrow 3:00 pm, but cannot promise the exact time because its activity depends on a task of another project for which the schedule is not yet fixed. EL1 knows that if CA1, at the worst case, installs the wall tomorrow 3:00 pm, he will have to wait for this for two hours. So far, there is not another better choice, so EL1 decides to accept CA1's commitment and updates his belief base by adding that CA1 has been committed to install the interior wall by tomorrow 3:00 pm.

...

- Plumber PL1 knows that before he can fasten the pipe to the frame, someone has to erect the frame. After consulting with GC to get information about who is in charge of frames, PL1 sends a message to CA1 requesting that CA1 erects the frame before PL1 starts setting up pipes at 4:00 pm.

### Implications

Although the above example is commonplace in the construction field and is easily handled by human engineers, it is not a toy problem from a computer modeling point of view. Some important implications emerge from the above description.

- **Agents are capable of deciding what they want to do and what they can do.** In either human or machine organizations, usually an agent is expected to do many things by different agents. In fact, each agent has its own value for selecting goals, and its own interests and capabilities for selecting tasks, and it has capacities to determine whether it really can do what it thought it was capable of doing.
- **Agents hold some kind of expertise.** Carpenter CA1 is given the carpentry-task which includes installing frames, walls and floor

[2] There may be alternative scenarios for this case example. The one we describe here is only one scenario.

tiles. CA1 should know how (in what order) to install these objects. We call this knowledge agent expertise. From an individual agent's point of view, expertise may vary in complexity. Some agents have very simple expertise and consequently can solve only simple problems; others can solve complex problems such as structural design of buildings. From the multi-agent team point of view, part or all of the expertise of an agent can be shared by one or more other agents. In the above example, CA1 and CA2 share expertise of carpentry.

- **Agents hold beliefs about the world.** Agents have their model of the world which is composed of what they perceive (e.g. the drawing of the bathroom in the above example) and what they are informed. Agents believe their model of the world is true (from their perspective) and will update the model whenever they perceive or receive any new information. From an organizational point of view, agents' beliefs are partial and sometimes inconsistent. Co-ordination may help them reach a more consistent view of the world.
- **Co-ordination for task distribution and interdependence.** Co-ordination among agents is required when tasks are to be distributed between agents, and when interdependencies exist between the tasks of different agents. While task distribution is likely to be easy if the optimal mapping between tasks and agents is not required, solving the interdependency problem requires sophisticated co-ordination.
- **Commitments between agents facilitate the task relationships among them.** Commitments among agents play an important role for integrating multi-agent activities. Commitments are mutually agreed constraints on actions and beliefs and have associated resources (e.g. time) that are used for commitment execution. Agents make commitments for future situations and execute the commitments that they have made previously. The goal of co-ordination is to dynamically generate a commitment network that matches the situation.
- **Agents share knowledge.** Agents share knowledge in two ways. First, they share a communication language and some of each other's domain knowledge, e.g. both EL1 and CA1 know that a switch can be set up only after the wall is erected. Second, agents have knowledge about each other, ranging from knowing each other's communication address to knowing the domains of interests and capabilities of other agents. In the above example, GC knows the domain of interest of its subcontractors; and subcontractors get to know each other through communication with GC and with each other.
- **Knowledge and communication structure have an impact on team performance.** In this example, GC knows how to break down the original task, CA1 knows how to plan and install carpentry components, and so on. The knowledge held by all the agents covers the problem domain. If CA1 is also capable of taking care of electricity work and EL1 is not involved in the team, i.e. more knowledge is centralized in CA1, the problem can still be solved but in a different way. In this example, lateral communication between subcontractors is possible, so subcontractors can communicate with one another to resolve the interdependence between their activities. If this communication is impossible, then all subcontractors will have to talk to each other through GC. There may still be some solution but the performance of the team will most probably not be the same.

### Cognition of Agents

Intelligent agents, like people, are very complex, and it is often not easy to construct a model that can sufficiently, coherently and mechanistically describe their behavior. In i-AGENTS we assume that each agent has its own cognitive basis that specifies the internal mechanism for the agent to perform, and governs the agent's external behavior. We further assume that both CDMS and CDSS coupled with humans can be described in a uniform way using cognitive attributes. Our assumptions are motivated by research in philosophy (Polanyi, 1958; Bratman, 1990), cognitive science (Anderson, 1983; Nowakowska, 1986), and computer science (Cohen and Levesque, 1990a,b; Shoham, 1990). We will demonstrate that these assumptions are useful for investigating co-ordination issues in multi-agent teams.

**Table 2** Agent cognitive attributes and their values for a carpenter

| Category | Attribute | Definition | Example value |
|---|---|---|---|
| Character | Values | Criteria used in selecting goals. | Cost-first, quality-first |
| | Interest | Problem domains about which agent is interested. | Frames, walls |
| | Capability | Actions agent can carry out. | Plan, install |
| | Expertise | Rules/functions for solving domain problems. | Carpenter-knowledge |
| | Strategy | Rules/functions for interacting with others. | Self-interested |
| Mental state | Goal | Current goals to be achieved. | Plan-bathroom |
| | Capacity | Time and tools available to agent. | Free: 1–2 pm, Has: wrench |
| | Social role | Formal social positions. | Subcontractor |
| | Commitment | Committed tasks. | To GC: plan bathroom |
| | Belief | Knowledge of the facts true in the world. | (GC ..., EL 1 ..., ...) |

In i-AGENTS, the cognition of an agent is described in terms of **cognitive attributes** that fall into two categories: **character** and **mental state**. Table 2 provides a simple definition and example values of these attributes of a carpenter. Note that the values of the attributes tend to be more dynamic towards the bottom of the table.

*Character of Agents*

Character describes relatively static characteristics of an agent. In our current model, we assume that the character of agents does not change over time.[3]

- **Values**: The values of an agent specify the criteria used by the agent to select its goals. In our research, values are also used to establish group values and to distinguish the deviation of individuals from the group. In order for agents to work collaboratively together they need to share some values. In the above example, the subcontractors share the value cost-first.
- **Interest**. An agent's interest defines problem domains in which the agent is interested.[4] In our model we assume that all agents are interested in listening to incoming messages, but they may choose to ignore a message if they think the content of the message is not interesting. In i-AGENTS the problem domain is described in terms of task objects described above. Therefore, in the above example CA1's interested domain can be described as Frames, Walls, etc. Interest of an agents impacts how the agent chooses its goals. In i-AGENTS we assume that agents will not choose goals that have nothing to do with their local interest.
- **Capability**. The capability of an agent specifies the potential for the agent to **directly** perform some action.[5] In i-AGENTS we distinguish between **capable** of doing something and **can** do something. An agent being **capable** of doing something does not mean that the agent **can** do the thing under any circumstance. If the agent **can** do it, some conditions or constraints associated with the action must be satisfied. Among those conditions, the most important is resource availability. Carpenter CA1 is capable of installing the wall, but cannot do it today just because the schedule for today is already full.
- **Expertise**: In i-AGENTS we assume that each agent involved in the team is playing a

[3] High-level learning may change character attributes of agents. We plan to relax this strong assumption in future, when we address the impact of high-level learning on organizational problem solving.

[4] This is different from interests defined in DesignWorld (Genesereth, 1991), where the interests of an agent are used by a centralized facilitator for selectively forwarding messages.

[5] In our research we distinguish between direct and indirect capability. For example, the general contractor GC is not capable of installing frames and walls because he has no knowledge of that domain. He thus hires subcontractor CA1 and extends his capability through his control over CA1.

certain functional role to the extent that it solves some part of the overall design problem of the team. Expertise is the knowledge that agents hold for solving their part of various domain problems. A plumber knows in what order it should set up bathroom stalls. For an individual agent, expertise may vary in its depth and breadth. Agents with deeper expertise can solve their problem at both abstract and detailed levels; and those with broader expertise can solve problems in a wider range of domains. If we look at the expertise from the team point of view, expertise of the team may vary in coverage, centrality and redundancy as described above.

- **Strategy**: In i-AGENTS, we clearly distinguish between the knowledge for domain problem solving, i.e. expertise, and that for interaction or co-ordination with other agents. We call the latter 'strategy'. Strategy specifies the general plan for co-ordination behavior that constrains responses to the incoming message and balances the agent's local interests with global (team) interests. For example, a general strategy of design agents can be described as: (if the supervisor requests me to do a task, and the domain and requirement of the task match my general interests and capability, then I will commit to the task), and (if a peer agent requests me to do a task, and my domain interests, capability and capacity allow, commit to the task). The difference between these two rules is that if the request is from the supervisor, then the agent just commits to it without thinking about the availability of the resource (i.e. capacity). If the resource is not available, the agent will have to find ways to make it available. If the request is from peer agents, then the agent just does what it can, but not make further efforts. In i-AGENTS we treat strategy at different levels: some are common strategies shared by a large range of agents, some are less general and shared by agents in a specific category, while others are agent-specific.

The character of an agent described above specifies the basis of the cognitive behavior of the agent. From an organizational modeling perspective, the explicit representation of character of agents makes it possible for us to describe the **level** of knowledge of agents and the **attitudes** of agents in applying their knowledge under certain situations. Consequently, understanding the interplay between the characters of agents, including their distribution, and organizational performance may result in insights for hiring correct personnel, or for establishing better training systems.

### *Mental State of Agents*

The mental state of an agent represents the agent's cognitive model of the real world. Since the real world is volatile when agents carry out their decision making and actions, the mental states change whenever the agents perceive or are informed about new incidents in the real world. An agent may establish its cognitive model of the real world by incrementally acquiring new information and learning from its experience.

- **Goals**: Generally, a goal is a proposition that the agent tries to make true and the proposition can be anything. In i-AGENTS, however, we assume that a goal for an agent is a task to be accomplished by the agent directly by its own actions, or indirectly by trying to share the goals with other agents and letting others perform part or all of required actions. Agents' selection of their goals is triggered by perceiving or being informed about new information, and is based on their value, interests, capability, and possibly on social roles and capacities. If necessary, agents may elaborate their top-level goals into sub-goals using their expertise.
- **Social role**: As described above, our agent model is used in the context of multi-agent organizations and we are interested in explicating the impact of agent knowledge and organizational mechanism on organizational performance. Social role is the key attribute of agents that links individuals with their organizations. Social roles are expectations for, or evaluative standards employed in, assessing the behavior of occupants of specific social positions. An agent may be assigned a role by other agents or system designers or by itself depending on the situation. In the next section we

describe more details of the social role and its properties.

- **Capacity**: The capacity of an agent describes the availability of resources that are required for the agent to perform actions. In i-AGENTS, resources include time, space, tools, and materials. In order to erect the frame of the bathroom, a carpenter has to spend two hours on it. At a given time, the capacity of an agent may be overloaded, full, underloaded or even idle. The load changes over time. It is important to note that capacity is the dimension in which most inter-agent activity conflicts occur.
- **Commitments**: The commitments of an agent specify the constraints on future actions of the agent. They create mutual beliefs in a collective plan among agents, as well as responsibility which holds each agent to his or her part. Agents make commitments for future situations and execute commitments that they have made previously. Commitments play the role of co-ordinating agents' future actions. Once a future action is agreed upon, agents typically decide on other future actions to take, treating the commitment as a fixed constraint. In i-AGENTS, commitments are considered as an attribute of individual agents and the results of choice by the agents (Cohen, 1990; Gasser, 1991; Bond and Gasser, 1988). That is, we view commitments as individual commitments rather than social ones (Becker, 1960; Gerson, 1976).
- **Beliefs**: Beliefs of an agent construct a cognitive model of the real world which is composed of the physical world, representing the tasks, and the social world, representing all relevant agents and their social situations. Agents make their choices (decisions) based on beliefs about the world, though their cognitive model of the world may be incorrect, incomplete, and inconsistent with that of other agents. An agent may update its belief when it perceives or is told of any changes in the real world.

The cognition of agents, as described above, provides a basis that determines agents' cognitive behavior during their interactions and local problem solving. In order to act as a social agent, an agent must possess the knowledge that promotes and constrains cognitive and external behavior of the agent. We observed that shared knowledge (including knowledge about others) is one of the kinds of knowledge an agent should hold to participate in organizational problem solving.

### Shared Knowledge

In i-AGENTS we follow the hypothesis that a base of common knowledge is necessary for multiple agents to work together. By shared knowledge we mean the knowledge common to two or more agents in the organization. In our research we assume that all agents share a common communication language and ontology describing the domain of interest. Agents may share problem-solving expertise, and this knowledge sharing feature is captured in i-AGENTS by the knowledge structure of co-ordination schemes described above.

From an organizational problem-solving point of view, agents' knowledge about other agents plays an important role in organizational performance. Knowing about other agents and tracking their changes is important for agents to collaborate with each other efficiently. For example, a manager agent with a number of tasks to be distributed needs to know who will be capable of doing a certain task. An electrician who is to install the electrical system of a room needs to know whom to talk with to ensure that the structure (frames and walls) will be installed by a given time. Generally, the knowledge of others may be acquired through communication. It is possible for an agent to initiate a broadcast communication whenever it needs some information from other agents. However, it is desirable for an agent to have a memory that contains both the static and dynamic knowledge of related agents. Therefore, the agent model should provide means to address the following questions: In order to achieve more efficient collaboration: how much should an agent know about others? What aspect (e.g. attributes) should an agent know about which other agent? How can and should agents get to know each other? How may knowing others affect co-ordination?

In i-AGENTS knowledge about others is divided into several levels, from knowledge about the existence of others to knowledge of the character of others. Table 3 shows how the

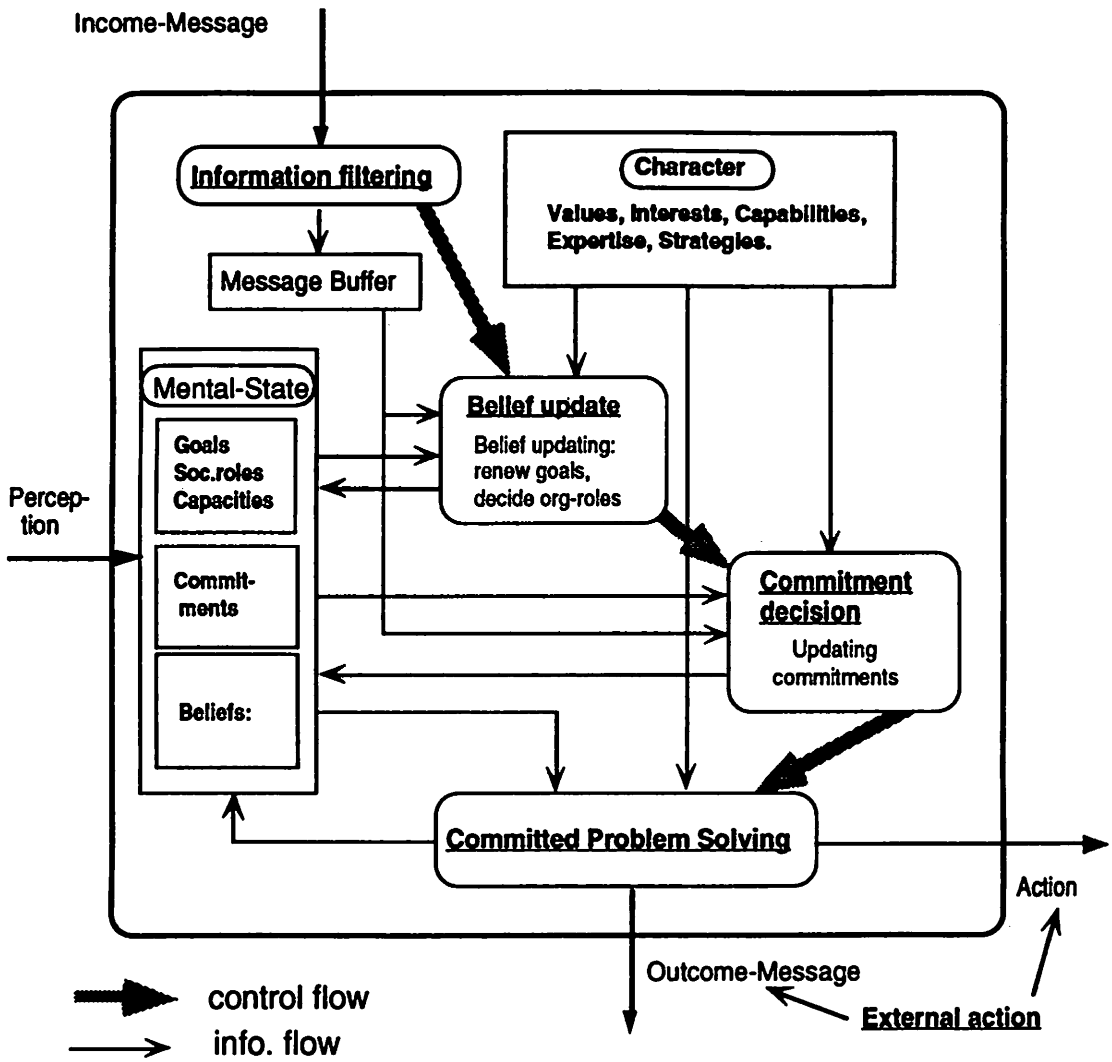


**Figure 3** How agents act

knowledge about others is explicitly represented in i-AGENTS using cognitive attributes of agents. In the current implementation an agents' knowledge about others is set at the initialization phase and will not change during organizational problem solving. We believe that changes in the structure of knowledge about others will impact organizational performance and our i-AGENTS model makes it possible for us to investigate the impacts.

It is important to note that knowledge about others can be recursive. General contractor GC may know what carpenter CA1 can do, and he may also know, or may not know, that CA1 knows that he knows what CA1 can do. In order to keep it simple, i-AGENTS does not explicitly represent the recursion of knowledge about others but assumes recursive knowledge about others.

### How Agents Act

Actions of agents in i-AGENTS is triggered by two types of external events: new incoming messages and perceivable changes to the world. As shown in Figure 3, an agent's action process includes the following phases:

- **Information filtering:** Agents live in a

**Table 3** Agents' knowledge about others represented in i-AGENTS

| Knowledge about others | Represented by |
|---|---|
| Existence of others | Name or reference to the others |
| Who needs what | Cognitive attribute: Interest |
| Who can do what | Cognitive attribute: Capability |
| Who knows what | Cognitive attribute: Beliefs |
| Relationship between others | Cognitive attribute: Social role |

dynamically changing environment. An agent receives new information when it receives a message from another agent and/or when it perceives any change in the modeled world. The incoming new information is differentiated into **interesting** and **non-interesting** information through a filtering process based on the agent's values, interests and capability. Interesting information is kept in memory for further processing and non-interesting information is thrown away.

- **Belief update**: Upon receiving new interesting information, an agent updates its belief-base. For the example in Figure 2, after receiving the commitment message from CA1, general contractor GC updates its belief base by adding 'CA1 commited to GC to plan carpentry work' to its belief-base.
- **Commitment decision**: After its belief-base is updated an agent then decides whether to make new commitments or to uncommit the old commitments and reschedule (reorder) all commitments. A new commitment may be requested by another agent in an incoming message or by the agent itself after perceiving a change in the world. The result of the commitment decision may change the agent's commitment, goal (current goal corresponds to current commitment to be carried out), capacity and social role.
- **Committed problem solving**: Committed problem solving is the process in which an agent solves its part of the organizational problem based on its expertise. Because tasks for different agents are usually interdependent on each other, this problem-solving process is also the place where interactions among agents are generated. Besides expertise, an agent's strategy and social role are the basis for the agent to make interaction decisions. The result of this problem-solving process may change the agent's goal, capacity and social role, and produce the agent's external actions.
- **External action**: By external action we mean the agent action that has an effect on the world external to the agent. There are two types of external actions: one is to send out a message to other agent(s) and the other is to perform some action that changes the state of the world. The external actions of an agent constitute its external behavior and contribute to the organizational problem solving either constructively or destructively.

## AGENT ORGANIZATIONS

Organizations are collectivities oriented to the pursuit of relatively specific goals and exhibiting relatively highly formalized social structures (Scott, 1987). From a computer modeling point of view, an organization can be implicitly modeled by defining a group of agents and setting up authority and communication relationships between the agents. In i-AGENTS, however, we explicitly represent organizations and organizational structure by introducing organizational roles (Jin and Koyoma, 1990).

### Organizational Roles

An organizational role defines expectations for, or evaluative standards employed in, assessing the behavior of occupants of specific social positions which represent locations in a system of social relationships (Scott, 1987). A role can be viewed as an abstract agent that defines the required behavior of agents who play the role, restricts the co-ordination levels, and facilitates representation of organizational structures. The organizational role description also represents the knowledge about the organization that can be shared by agents. For example, in a design project team each team member should know what the project manager—which is a role rather than a specific agent—is supposed to do, and its relationships with mechanical engineers and electrical engineers.

A role in i-AGENTS is defined to hold four attributes, i.e. Name, Qualification, Responsibility, Relation-with-other-roles:

- **Name**: Specify the functional name of the role. In the engineering domain a multi-agent team is usually composed of several functional entities which perform different functions (e.g. architectural design, structural design, and project management) during the problem-solving process. Roles in i-AGENTS correspond to the descriptions of these entities.
- **Eligibility**: Specify who can play the role. In many cases a role requires the agent that is to play the role must hold resources and rationality above a certain level which is the minimum requirement for solving the tasks of the role.
- **Requirements**: Specify what tasks the role should accomplish under a certain time constraint. The value of this attribute depends on the task definition. In most cases responsibility also specifies the level of performance, e.g. timeliness, and accuracy.
- **Relation-with-other-roles**: Specify what relationships the role should have with other roles. Two types of relations can be specified by this attribute, i.e. authority relations in which higher level roles control the lower ones, and communication relations.

Roles, as defined above, have the following properties. First, they specify and constrain the behavior of agents, so they may be used to guide individual activities of agents. Second, roles are compositional, and they can be combined to form a simple and flat or a complex and hierarchical structure. This attractive property of the role led us to defining organizational structure based on roles. Third, the definition of a role may change through interaction between agents, though predefining roles in a static way is simpler, and still effective. Finally, when viewed by an agent, a role has feasibility and desirability. A role is feasible for an agent if the agent is qualified to play the role. When there are alternative feasible roles, an agent prefers more desirable roles.

## Organizational Structuring

An organization's structure defines the ways in which it divides its agents into distinct tasks and achieves co-ordination among them (Mintzberg, 1979). In i-AGENTS an organizational structure is explicitly represented as a set of organizational roles and the relationships between the roles. A role is instantiated when it is assumed by an agent. When all the roles in an organizational structure are instantiated, we say that an organization is created. We allow multiple agents to play one role or an agent to play multiple roles, depending on the organizational design. Once an organization is created, agents may behave on their own according to the specification of their roles. In i-AGENTS we assume that an organizational structure is static during organizational problem solving, but the instantiation of the organizational structure can change over time. This means that we can assign different agents (people, computers) to a formal position, i.e. a role, dynamically during organizational problem solving.

Though we may define an organizational structure based on roles, identifying the roles that are required for a certain organizational task remains an organization design problem. Using role-based organizational structuring representation, we can design different organizational structures and use the computer to simulate organizational behavior. Figure 4 illustrates an example of using role-based organizational structuring to create an organiza-

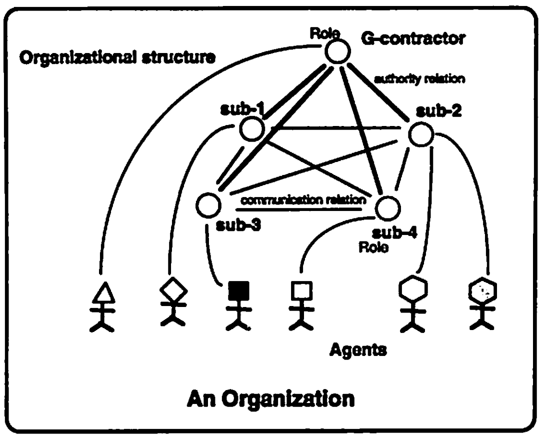


**Figure 4** Role-based organizational structuring

tion for the example project described earlier. It is worth noting that the role based organizational structure appears to be the most important common knowledge shared by agents modeled in i-AGENTS.

## IMPLEMENTATION

The i-AGENTS framework is implemented as a symbolic model using object-oriented programming techniques. It has a set of objects with attributes and methods to define the tasks, the agents, the organizations, and the co-ordination schemes. i-AGENTS is currently implemented on Sun workstations in KEE, a Lisp-based object-oriented knowledge engineering environment. It is being moved into Prokappa, a C-based successor to KEE.

The agent model is implemented to consist of an interactive agent description, representing the general interactive agent behavior, and an expert library providing knowledge for solving engineering problems. An instance of an agent is created by inheriting descriptions from both the interactive agent and the expert library, as shown in Figure 5. Each expert description specifies default values of interest and capability, but users can over-write the values by setting specific default values. An organizational structure is defined by a set of organizational roles and the relationships between the roles. An organization emerges after agents assume their roles. Tasks in i-AGENTS are composed of **task action** model and a **task object** model which are defined corresponding to the expertise library. The organizational problem solving simulation starts when a user sends a message to the team to accomplish a top-level task, e.g. create the bathroom construction plan.

The outputs of the system include statistics describing the status of agents, organization, tasks and organizational performance. There are two categories of performance evaluation. One is the **quality of organizational problem solving**, including successes/failures, reasoning steps, conflict-resolution ratio, and the number of backtrackings. The other is **communication overhead**, including total amount of communication, vertical/lateral communication ratio, and communication overhead on each agent.

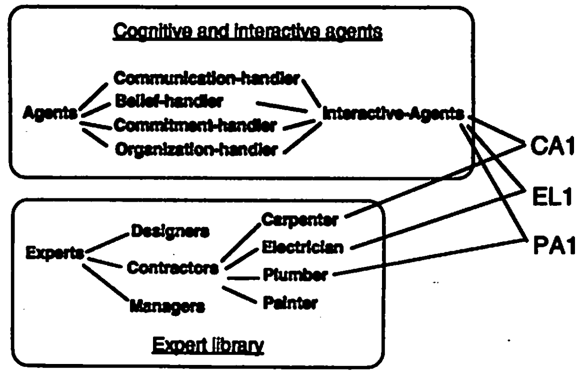


**Figure 5** Multiple inheritance for agent definition

We are presently in the process of validating the i-AGENTS model, and the results will be reported separately. Figure 6 is a screen dump of the i-AGENTS system that presents the solution—a multi-agent construction plan—to the bathroom project described above. The agents team is composed of the general contractor GC, and subcontractors CA1, EL1, PL1, and PA1. The interdependencies between agent activities are resolved by agents through interactions. Different administrative and communication structures may result in different communication overhead and a different number of reasoning steps, but will arrive at almost the same solution. Our primary test has confirmed the effectiveness and the internal consistency (Masuch and LaPotin, 1989) of the i-AGENTS model.

## RELATED WORKS

Research in the DAI field (Bond and Gasser, 1988; Gasser and Huhns, 1989; Durfee *et al.*, 1989) has been focused on understanding the knowledge and reasoning requirements for agents to co-ordinate with each other in a multi-agent environment. Although some DAI researchers treat organizational structuring as a strategy to co-ordinate multi-agent computer systems (Fox 1981, Corkhill and Lesser, 1983; Bond and Ansser, 1988), few address the organizational issues in a **social** context. Our research on organizational problem solving focuses on organizational design issues for multi-agent teams where both human and

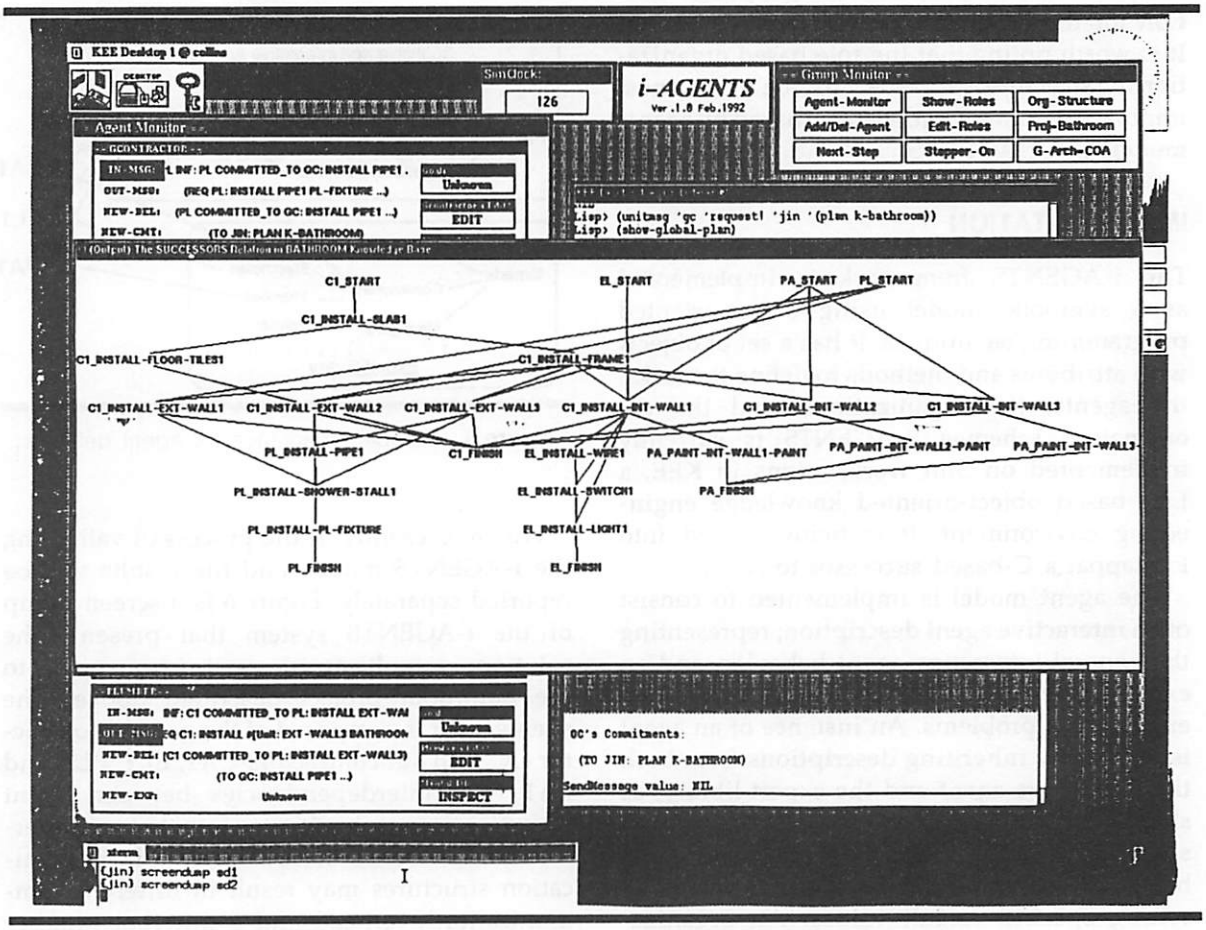


**Figure 6** The multi-agent plan developed by the agent-based organization for the bathroom project

computer agents are involved, and we model human agents using character and mental state.

From a point of view of intelligent system modeling, character is very similar to the higher-level control of the blackboard system described by Hayes-Roth (1985). The difference is that while Hayes-Roth's higher-level control mechanism is intended to define sophisticated control for an intelligent system to deal with its environment where the existence of other agents is not assumed, the character defined in our research is used to facilitate the interactive behavior of the agents who exist in an organization and try to make good use of their collective **expertise** and **capability** and to balance their own **values** and **interests** with the group ones through using their own **strategies**. The mental state of i-AGENTS follows that of Shoham's Agent-0; in particular, the concepts of commitments and beliefs are the same. The difference between our research and his is that while agents in Agent-0 use predefined commitment-rules for commitment decision making, agents in i-AGENTS make commitments based on their higher-level characters. Thus our agents are more specific and customized.

Musuch and LaPotin's (1989) DoubleAISS is a pioneering attempt to apply a symbolic modeling approach to model agents as decision makers, communication among agents, and organizational structures. Our research is different from DoubleAISS in the level of abstraction. It aims at developing a model of organizational problem solving in the engineering domain, and needs to model both tasks and agents in considerable detail. DoubleAISS, however, uses simplified tasks and agents to explore the extent to which a symbolic model can be used for organization research.

Carley *et al*'s (1992) Plural-Soar is another computerized multi-agent system used for organization study. In Plural-Soar, the organizational task is detailed enough to address interplay between job requirements, agents' skills, and overall schemes for co-ordination among agents. Although Plural-Soar is quite similar to i-AGENTS, in the sense that they both model tasks and agents in detail, they have different interests. While Carley *et al*'s objective is to develop a general and unified organization theory, ours is more modest. We aim to develop models and methods to guide organizational design in engineering. This difference in goals leads to a difference in complexity of tasks and agents. While Plural-Soar uses simple tasks and generally intelligent agents, i-AGENTS emphasizes complex and real engineering tasks, and cognitive and intelligent agents with specialized expertise.

In a parallel research project, called The Virtual Design Team (VDT), we have taken the first step toward developing analysis tools for systematically designing organization structures (Cohen, 1992; Levitt *et al.*, 1993; Christiansen, 1993; Jin *et al.*, 1994). VDT is concerned with the organizational performance of design teams performing relatively routine tasks, and attempts to explicate the interplay among team performance, communication tools used by the team, and team's organizational structure. VDT takes an information-processing approach to model organizations. It describes tasks based on engineering principles, and models capability, preference and capacity of agents in terms of behavioral parameters. VDT can predict project duration based on the organizational structure and communication tools used by the team. However, the superficial notions of agent cognition in VDT do not permit it to capture the impact of knowledge on organizational performance, nor to relate important phenomena such as negotiation, agent learning and task scheduling to organizational performance. This is the main concern of i-AGENTS research.

## SUMMARY AND FUTURE WORK

This paper has described i-AGENTS, a computational model of organizational problem solving in multi-agent teams. i-AGENTS is composed of several high-level concepts including tasks, agents, organizations, and co-ordination scheme.

The task model in i-AGENTS consists of a number of elements such as task, task action, task object, task constraints, and task relations. We argue that in order to investigate the impact of knowledge and policy on organizational performance in the engineering domain, the task should be described in sufficient detail to mirror the cognitive features of agents and their organizations, and should have enough complexity to reflect the engineering domain.

Agents in i-AGENTS are described in terms of character and mental state. The character of an agent determines its values, interests, capability, expertise and co-ordination strategy. It governs the cognitive behavior of the agent. The mental state of an agent represents its cognitive model of the real world. The action of an agent is triggered by external event perceived and received by the agents. Agent action is carried out through five phases, i.e. information filtering, belief updating, commitment decision making, committed problem solving and external action. Agents may share knowledge, and may hold knowledge of the organization and knowledge about other agents.

An organization is explicitly represented to include organizational roles, tasks and a number of agents. The role-based organizational structure allows us to represent shared organizational knowledge, and to define various organizational structures in a natural and powerful way. The concept of co-ordination scheme allows us to impose various organizational strategies and to observe their effectiveness and efficiency through computer-based simulation.

From an organizational perspective our research on i-AGENTS extends traditional information processing models of organizations—which, as Carley *et al.* (1992) point out, tend to leave concepts such as knowledge ill-defined with respect to cognition—by addressing the impact of knowledge that governs agent's capabilities and cognitive behavior such as preference. It has been pointed out that without a detailed task, communication becomes fearless, actions become uniform and agents become skill-less (Carley and Prietula, 1992). Through explicitly representing agent cognition and knowledge-based organizational problem solving, i-

AGENTS allows us to explore not only the impact of organizational structuring but also task- and agent-constrained organizational phenomena such as the role of agent knowledge, negotiation, task distribution and scheduling etc. By increasing the complexity of tasks and introducing the expertise associated with tasks, i-AGENTS may be used to explicate organizational issues in specific task domains.

When viewed from an engineering perspective, i-AGENTS is the first step toward an organizational problem-solving model that merges organization theory and DAI and can be used to simulate or analyze organizational behavior of teams in particular engineering domains at a very specific level of detail. We anticipate that organization-engineering tools will be required for firms to match their organizational structure—including information processing and communication tools—to dynamic and changing environments. This kind of tool may help them to achieve the full potential advantages of computer technology by taking into account both the **social** and the **technical** aspects of CIM, CE and CIE.

We plan to validate the i-AGENTS framework by testing a number of typical engineering case examples and will choose two typical organizations from construction and semi-conductor design domains, respectively, and formulate test cases using the i-AGENTS model. We will attempt to further validate our model and system through a process of extensive testing by deliver the i-AGENTS system to users in real engineering organizations. We expect that extensive testing will enrich the i-AGENTS model and eventually make it a computer tool for designing engineering organizations.

## Acknowledgments

This research has been supported by CIFE, the Center for Integrated Facility Engineering, at Stanford University. We are grateful to Tore Christiansen and Gaye Oralkan for their helpful comments on an early draft of this paper and to three anonymous reviewers who provided us with excellent critiques and detailed suggestions for sharpening the paper's focus.

## References

Anderson, J. *The Architecture of Cognition*, Harvard University Press, Cambridge, MA, 1983.

Becker, H.S. 'Notes of the concept of commitment', *American Journal of Sociology*, **66**, 1960, pp. 32–40.

Bond, A.H., and Gasser, L. (eds), *Readings in Distributed Artificial Intelligence*, Morgan Kaufman, San Mateo, CA, 1988.

Bratman, M.E., 'What is intention?' in Cohen, P.R., Morgan, J. and Pollock, M.E. (eds), *Intentions in Communication*, MIT Press, Cambridge, MA, 1990, pp. 15–31.

Carley, K.M.C., Kjaer-Hansen, J., Newell, A. and Prietula, M., 'Plural-Soar: a prolegomenon to artificial agents and organizational behavior', In *Artificial Intelligence in Organization and Management Theory*, Masuch, M. and Warglien, M. (eds), North-Holland, Amsterdam, 1992, pp. 87–118.

Carley, K.M.C. and Prietula, M., 'ACTS THEORY: Extending the model of bounded rationality', Technical Report, Carnegie-Mellon University, 1992.

Chandler, A.D. and Dames, H. (eds) *Managerial Hierarchies*, Harvard University Press, Cambridge, MA, 1980.

Chen, M. and Nunamaker, J. F., 'Integration of organization and information systems modeling: an object-oriented approach', *Proceedings of the Twenty-Second Annual Hawaii International Conference on System Sciences*, Vol. III, 1989, pp. 70–79.

Chen, A.T. and Ward, A.C., 'Unifying QFD, Pugh concept selection, function structures and morphological analysis', University of Michigan, Dept of Mech. Eng. and App. Mech., Design Lab. 1991.

Christiansen, R.T., *Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams*, PhD thesis, Stanford University, September 1993.

Cohen M.D., March, J.G. and Olsen, J.P., 'A garbage can model of organizational choice', *Administrative Science Quarterley*, **17**, No. 1, 1972, pp. 1–25.

Cohen, M.D., 'Artificial intelligence and the dynamic performance of organizational designs', in March, J.G. and Weissinger-Baylon, R. (eds), *Ambiguity and Command: Organizational Perspectives on Military Decision-Making*, Pitman, Boston, MA, 1986, pp. 53–71.

Cohen G.P., *The Virtual Design Team: An Information Processing Model of the Design Team Management*, PhD thesis, Stanford University, December 1992.

Cohen P.R., and Levesque, H.J., 'Persistence, intention and commitments', in Cohen, P.R., Morgan J. and Pollock, M.E. (eds), *Intentions in Communication*, MIT Press, Cambridge, MA, 1990, pp. 33–69.

Corkill D.D. and Lesser, V.R., 'The use of meta-level control for coordination in a distributed problem solving network', in *Proceedings of the 1983 International Joint Conference on Artificial Intelligence*, pp. 748–756, August 1983.

Cyert, R.M. and March, J.G., *A Behavioral Theory of the Firm*, Prentice-Hall, Englewood Cliffs, NJ, 1963.

Darwiche, A., Levitt, R.E. and Hayes-Roth, B., 'OARPLAN: Generating project plans in a blackboard system by reasoning about objects, actions and resources', *The Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, **2**(3), 1989.

Davis, R., 'Report on the Second Workshop on Distributed Artificial Intelligence', *SIGART Newsletter*, **80**, 1982, pp. 13–23.

Durfee, E.H., Lesser, V.R. and Corkill, D.D., 'Trends in cooperative distributed problem solving', in *IEEE Transactions on Knowledge and Data Engineering*, **KDE-1** (1), 1989, pp. 63–83.

Fayol, H. *General and Industrial Management*, Pitman Press, London, 1949 (first published 1919).

Fox, M. 'An Organizational View of Distributed Systems' in *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-11**(1), 1981, pp. 70–80.

Galbraith, J.R., *Organization Design*, Addison-Wesley, Reading, MA, 1977.

Gasser, L. and Huhns, M.M. (eds) *Distributed Artificial Intelligence II*, Pitman, London, 1989.

Gasser L., 'Social conceptions of knowledge and action: DAI Foundations and open systems semantics', *Artificial Intelligence Journal*, **47**, 1991, pp. 107–38.

Genesereth, M., 'Design World', in *Proceedings of IEEE Conference on Robotics and Automation*, 1991.

Genesereth, M., Fikes, R.E. *et al.*, 'Knowledge interchange format, Version 3.0 Reference Manual', *Technical Report Logic-92-1*, Computer Science Department, Stanford University, 1992.

Gerson, E.M., 'On quality of life', *American Journal of Sociology*, **41**, 1976, pp. 793–806.

Hannan, M.T. and Freeman, J. 'The population ecology of organizations', *American Journal of Sociology*, 1982, pp. 929–964.

Hayes-Roth, B., 'A blackboard architecture for control', *Artificial Intelligence Journal*, No. 26, 1985, pp. 251–321.

Hewitt, C. and J. Inman., 'DAI betwixt and between: from 'Intelligent Agents' to 'Open System Science', *IEEE Transactions on Systems Man and Cybernetics*, **SMC-21**(6), November/December 1991, pp. 1409–1419.

Huhns, M.M. (eds), *Distributed Artificial Intelligence*, Pitman, London, 1987.

Jin, Y. and Koyama, T., 'Multiagent planning through expectation based negotiation', in *Proceedings of AAAI 10th International Workshop on Distributed Artificial Intelligence*, Bandera, TX, USA, October 1990.

Jin, Y., Kunz, J.C., Levitt, R.E. and Winstanley, G., 'Design of project plans from fundamental knowledge of engineered systems', in *Proceedings of AAAI 92 Fall Symposium — Design From Physical Principles*, Cambridge, MA, October, 1992.

Jin, Y. and Levitt, R.E., 'Model-based and interactive planning approach to predictive maintenance management', in *Proceedings of Fifth International Conference on Computing in Civil Engineering*, Anaheim, CA, June 1993.

Jin, Y., Levitt, R. E., and Christiansen, T., 'The Virtual Design Team: A Computer Simulation Framework for Studying Organizational Aspects of Concurrent Design', to appear in *Proceedings of 1994 International Conference on Simulation in Engineering Education*, January 1994, Tempe, Arizona.

Koo, C., *A Distributed Model for Performance Systems — Synchronizing Plans among Intelligent Agents Via Communications*, PhD thesis, Stanford University, 1987.

Lesser, V. R. and Corkill, D. D., 'Functionally Accurate, Cooperative Distributed Systems', in *IEEE Transactions on Systems, Man and Cybernetics*, **SMC-11**(1), 1981, pp. 81–96.

Levitt, E., Jin, Y. and Dym, C.L., 'Knowledge-based support for management of concurrent, multidisciplinary design', *Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, (5) 2, 1991, pp. 77–95.

Levitt, R.E., Cohen, P.G., Kunz, J.C., Nass, C., Christiansen, T. and Jin, Y., 'The virtual design team: simulating how organization structure and communication tools affect team performance', to be published in Carley, K.M. and Prietula, M.J. (eds), *Computational Organization Theory*, 1993.

Mackenzie, K.D., *Organizational Structures*, AHM Publishing Corporation, Arlington Heights, IL, 1978.

Malone, T.W., 'Modeling coordination in organization and markets', *Management Science*, **33**, 1987.

Malone, T.W. and Crowston, K., 'Towards an interdisciplinary theory of coordination', MIT Sloan School Working Paper 3294-91-MSA, April 1991.

March, J.G. and Simon, H.A., *Organizations*, John Wiley, New York, 1958.

Masuch, M. and LaPotin, P., 'Beyond garbage cans: an A1 model of organizational choice', *Administrative Science Quarterly*, 34, 1989, pp. 38–67.

Miao, X., Luh, P.B. and Kleiman, D.L. 'A normative-descriptive approach to hierarchical team resource allocation', *IEEE Transactions on Systems, Man and Cybernetics*, **22**, No. 3, 1992.

Mintzberg, H., *The Structure of Organizations*, Prentice-Hall Inc. NJ, 1979.

Nowakowska, M., *Cognitive Science—Basic Problems, New Perspectives, and Implications for Artificial Intelligence*, Academic Press, Orlando, FLA, 1986.

Perrow, C., 'A framework for the comparative analysis of organizations', *American Sociological Review*, **32**, No. 2, April 1967, pp. 194–208.

Pfeffer, J. and Salancik, G.R., *The External Control of Organizations*, New York: Harper & Row, 1978.

Polanyi, M., *Personal Knowledge: Towards a Post-critical Philosophy*, The University of Chicago Press, Chicago, 1958.

Scott, W.R., *Organizations: Rational, Natural and Open Systems* (first edition), Prentice-Hall, Englewood Cliffs, NJ, 1987.

Shoham, Y., 'Agent oriented programming', *Technical Report STAN-CS-1335-90*, Stanford University, 1990.

Simon, H. A., *Administrative Behavior: A Study of*

*Decision-Making Processes in Administrative Organization*, Free Press, New York, 1976.

Simon, H.A., *The Sciences of the Artificial*, second edition, MIT Press, Cambridge, MA, 1981.

Tatum, C.B., 'Organizing large projects: how managers decide', *Journal of Construction Engineering and Management, ASCE*, **110**, No. 3, September 1984, pp. 346–358.

Taylor, F., *The Principles of Scientific Management*, Harper, New York, 1911.

Thompson, J.D., *Organizations in Action: Social Science Bases in Administrative Theory*, McGraw-Hill, New York, 1967.

Williamson, O.E., 'Transaction costs economics: the government of contractural relations', *Journal of Law and Economics*, **22**, 1979, pp. 233–61.