CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Modeling Organizational Problem Solving in Multiagent Teams

by

Yan Jin and Raymond E. Levitt

CIFE Working Paper
Number 27

February, 1994

Stanford University

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Contents

Modeling Organizational Problem Solving in Multiagent Teams[1]

Yan Jin and **Raymond E. Levitt**

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

Abstract[2]

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 individual 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 organizational perspective, i-AGENTS extends traditional information processing models of organization (Galbraith 77) 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.

1. INTRODUCTION

The performance of organizations in engineering domains can be affected by a variety of factors including organizational task requirements, level of technology, individual 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 for the competitiveness of industrial firms, which operate in increasingly competitive global markets, and rapidly changing technological environments.

[1] This reseach was supported CIFE and in part by NSF Grant #IRI-9122541. The opinions expressed in this paper are solely those of the authors.

[2] A similar version of this paper has been accepted for publication by *International Journal of Intelligent Systems in Accounting, Finance and Management* for the special issue on "Mathematical and Computational Models of Organizations: Models and Characteristics of Agent Behavior".

of i-AGENTS. In Section 3, 4, and 5, we discuss in detail the concepts of *tasks, agents,* and *organizations,* respectively, and describe how those concepts are elaborated and represented in i-AGENTS. Section 6 describes the implementation of i-AGENTS and presents results from an initial simulation. Section 7 compares our research with related work, and Section 8 presents our conclusions and our plans for future work.

2. I-AGENTS FRAMEWORK

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

2.1. MULTIAGENT TEAMS AND ORGANIZATIONAL PROBLEM SOLVING

In our research, a *multiagent 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 multiagent 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, hearting, 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, and construction, etc. Within a given team, there can be subteams 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 multiagent 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 coordination issue in organizational problem solving, we need a uniform agent description, emphasizing both technical and social aspects of the team agents.

In a multiagent 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 multidisciplinary 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 *coordination* between design team participants (Levitt 91). Thus how to achieve appropriate coordination for better organizational performance is a key problem.

From the above descriptions, it is clear that in order to model organizational problem solving in multiagent teams, we need to incorporate and elaborate the following concepts.

• *Tasks*: An organizational task specifies the work to be carried out by a multiagent team. A task can be a complex real task, or a simple "toy" task. The task description can be abstract or

Coverage specifies how much of the knowledge required for solving the problems comprising a given task is covered by the knowledge held by the agents of the team. Generally speaking, full coverage is required for a multiagent team to perform a task well (Corkill 83).

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 multiagent teams. In order to coordinate their interdependent activities, agents need to share knowledge to a certain level. In most cases, higher knowledge redundancy implies robustness of an organization (Hewitt 1991).

Understanding the interplay between knowledge structure and the organizational performance of multiagent teams has important implications for design of a project team, as well as for training of its human work force.

• *Information Structure*: Information structure specifies how information flows among information sources, which can be agents or databases. An information structure 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, vs. (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 92).

• *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 coordinate their activities (Galbraith 77; Chandler 80). We believe that this insight applies to our multiagent 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 come up with solution alternatives themselves.

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

==

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

==

3.2. 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 multidisciplinary, agents working together. We argue that detailed task specification is an important factor that contributes to our ability to model the organizational performance of multiagent teams in engineering domains and that two conditions must be satisfied to explicate the contribution. First, the task description should be detailed enough so that we can assess the impact of individual 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 identifying elements and structures that are powerful enough to describe relatively complex tasks in detail and robust enough 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 89; Jin 92, 93). 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 91) for task description. That is, we view the task as a set of operations, decompose the overall task into several smaller subtasks, solve the decomposed subtasks, and recompose the local decisions into larger scale of task solutions. The top-level task, called *project,* is given to the representative of a multiagent team by a client. Other tasks or subtasks 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 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 92).

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 92).

==

4. 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 multiagent 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 91), and others describe agents in terms of knowledge and mental states (Shoham 90). In our research on Virtual Design Team (VDT), an agent's capability and preference is defined by a number of behavioral parameters (CohenG 92; Levitt 92, Christiansen 93). 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 92).

Since our research is concerned with coordination of interdependence in multiagent 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 90), 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.

4.1. 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 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 coordination.

The goal of the team is to accomplish the tasks 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 coordination 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[4].

- 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 subtasks including carpentry-task, electricity-task, plumbing-task, and painting-tasks, using his own

---

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

Agents hold some kind of expertise. Carpenter CA1 is given the carpentry-task which includes installing frames, walls, and floor 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 multiagent 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. Coordination may help them reach a more consistent view of the world.

Coordination for task distribution and interdependence. Coordination 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 coordination.

Commitments between agents facilitate the task relationships among agents. Commitments among agents play an important role for integrating multiagent activities. Commitments are mutually agreed constraints on actions and beliefs. Commitments 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 coordination 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 EL and CA know that a switch can be set up only after the wall is erected. Secondly 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.

4.3. 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

Expertise: In i-AGENTS, we assume that each agent involved in the team is playing a 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 in Section 2.4.

Strategy: In i-AGENTS, we clearly distinguish between the knowledge for domain problem solving, i.e., expertise, and that for interaction or coordination with other agents. We call the latter strategy. Strategy specifies the general plan for coordination 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 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 bahavior 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.

4.3.2 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 let 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 multiagent 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 which other agent? How can and should agents get to know each other? How may knowing others affect coordination?

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 knowledge about others is explicitly represented in i-AGENTS using cognitive attributes of agents. In the current implementation, an agent's 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.

==

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

==

4.5. 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:

==

Figure 3: How agents act

==

Information filtering: Agents live in a 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 committed 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.

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, roles 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. Lastly, 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.

5.2. ORGANIZATIONAL STRUCTURING

An organization's structure defines the ways in which it divides its agents into distinct tasks and achieves coordination among them (Mintzberg 79). 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 a 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, 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 organization for the example project described in Section 3.2. 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.

==

Figure 4: Role-based organizational structuring

==

6. 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 behaviors to define the tasks, the agents, the organizations, and the coordination 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 *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 notions 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.

Masuch and LaPotin's DoubleAISS (Masuch 89) 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. Our research 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 and her colleagues' Plural-Soar is another computerized multiagent system used for organization study (Carley 92). In Plural-Soar, the organizational task is detailed enough to address interplay between job requirements, agents' skills, and overall schemes for coordination 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 '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 (CohenG 92, Levitt 92, Christiansen 93). 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.

8. CONCLUSIONS AND FUTURE WORK

This paper has described i-AGENTS, a computational model of organizational problem solving in multiagent teams. i-AGENTS is composed of several high-level concepts including tasks, agents, organizations, and coordination 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 coordination strategy. It governs the cognitive behavior of the agent. The mental state of an agent represents its cognitive model

REFERENCES

(Anderson 83) Anderson, J. *The Architecture of Cognition*, Harvard University Press, Cambridge, MA, 1983.

(Becker 60) Becker, H.S, Notes of the Concept of Commitment, in *American Journal of Sociology*, 66, pp.32-40, 1960.

(Bond 88) Bond, A. H., and Gasser, L., (Joint Editors), *Readings in Distributed Artificial Intelligence*, Morgan Kaufman, San Mateo, California, 1988.

(Bratman 90) 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.

(Carley 92) 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* (Eds.) Masuch M. and Warglien M., North-Holland, Amsterdam, 1992, pp.87-118.

(Carley 92b) Carley, K. M. C. and Prietula, M., "ACTS THEORY: Extending the Model of Bounded Rationality", Technical Report, Carnegie Mellon University, 1992.

(Chandler 80) Chandler, A.D. and Dames, H. (Eds.), *Managerial Hierarchies*. Cambridge, MA: Harvard University Press, 1980.

(Chen 89) 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 91) Chen, A.T. and Ward, A.C., "Unifying QFD, Pugh Concept Selection, Function Structures & Morphological Analysis", Univ. of Michigan, Dept. of Mech. Eng. and App. Mech., Design Lab. 1991.

(Christiansen 93) Christainsen, R. T., *Modeling Efficiency and Effectiveness of Coordination in Engineering Design Teams.* Ph.D Thesis, Stanford University, September 1993.

(CohenM 72) Cohen, M. D., March, J. G., and Olsen, J. P., "A Garbage Can Model of Organizational Choice," *Administrative Science Quarterly*, Vol. 17, No. 1, 1972, pp. 1-25.

(CohenM86) Cohen, M. D., "Artificial Intelligence and the Dynamic Performance of Organizational Designs," in *Ambiguity and Command: Organizational Perspectives on Military Decision-Making*, (Eds.) March, J. G. and Weissinger-Baylon, R., Pittman, Boston, Massachusetts, 1986, pp. 53-71.

(CohenG 92) Cohen, G. P., *The Virtual Design Team: An Information Processing Model of the Design Team Management*, Ph.D Thesis, Stanford University, December 1992.

(Hannan 77) Hannan, M.T. & Freeman, J. "The population ecology of organizations", *American Jounal of Sociology,* 1982: 929-964.

(Hayes-Roth 85) Hayes-Roth, B., A Blackboard Architecture for Control. *Artificial Intelligence Journal,* Number 26, 1985, PP. 251-321.

(Hewitt 91) Hewitt, G. and Jeff Inman, "DAI Betwixt and Between: From 'Intelligent Agents' to Open System Science," in *IEEE Transactions on Systems, Man and Cybernetics*, SMC-21(6), pp.1409-1419, November/December 1991.

(Huhns 87) Huhns, M. N., (Eds) *Distributed Artificial Intelligence,* Pitman, London, 1987.

(Jin 90) 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 92) Jin, Y., Kunz, J.C., Levitt, R.E., and Winstanley, G., "Design of Project Plans from Foundamental Knowledge of Engineered Systems," in *Proceedings of AAAI 92 Fall Symposium -- Design From Physical Principles,* Cambridge, MA, October, 1992.

(Jin 93) Jin, Y. and Levitt, R.E., "Model-Based and Interactive Planning Approach to Predictive Maintenance Management," to appear in *Proceedings of Fifth International Conference on Computing in Civil Engineering*, Anaheim, CA., June 1993.

(Koo 87) Koo C., *A Distributed Model For Performance Systems - Synchronizing Plans Among Intelligent Agents Via Communications*, Ph.D. Thesis, Stanford University, 1987.

(Levitt 91) Levitt, Raymond E., Yan Jin and Clive L. Dym, "Knowledge-Based Support for Management of Concurrent, Multidisciplinary Design," *Journal of Artificial Intelligence in Engineerng Design, Analysis and Manufacturing, (5) 2,* 1991, pp. 77-95.

(Levitt 92) Levit, R. E., Cohen, P. G., Kunz, J. C., Nass, C. , Christiansen, T. and Jin, T., "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.*

(Mackenzie 78) Mackenzie, K.D., *Organizational Structures*, AHM Publishing Corporation, Arlington Heights, Illinois, 1978.

(Malone 87) Malone, T.W., "Modeling Coordination in Organization and Markets", in *Management Science*, 33, 1987.

(Malone 91) Malone, T. W. & Crowston, K. "Towards an Interdisciplinary Theory of Coordination", *MIT Sloan School Working Paper* # 3294-91-MSA, April 1991

(March 58) March, J. G., and Simon, H. A., *Organizations*, John Wiley, New York, 1958.

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 tasks 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 assumptions.	*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.

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

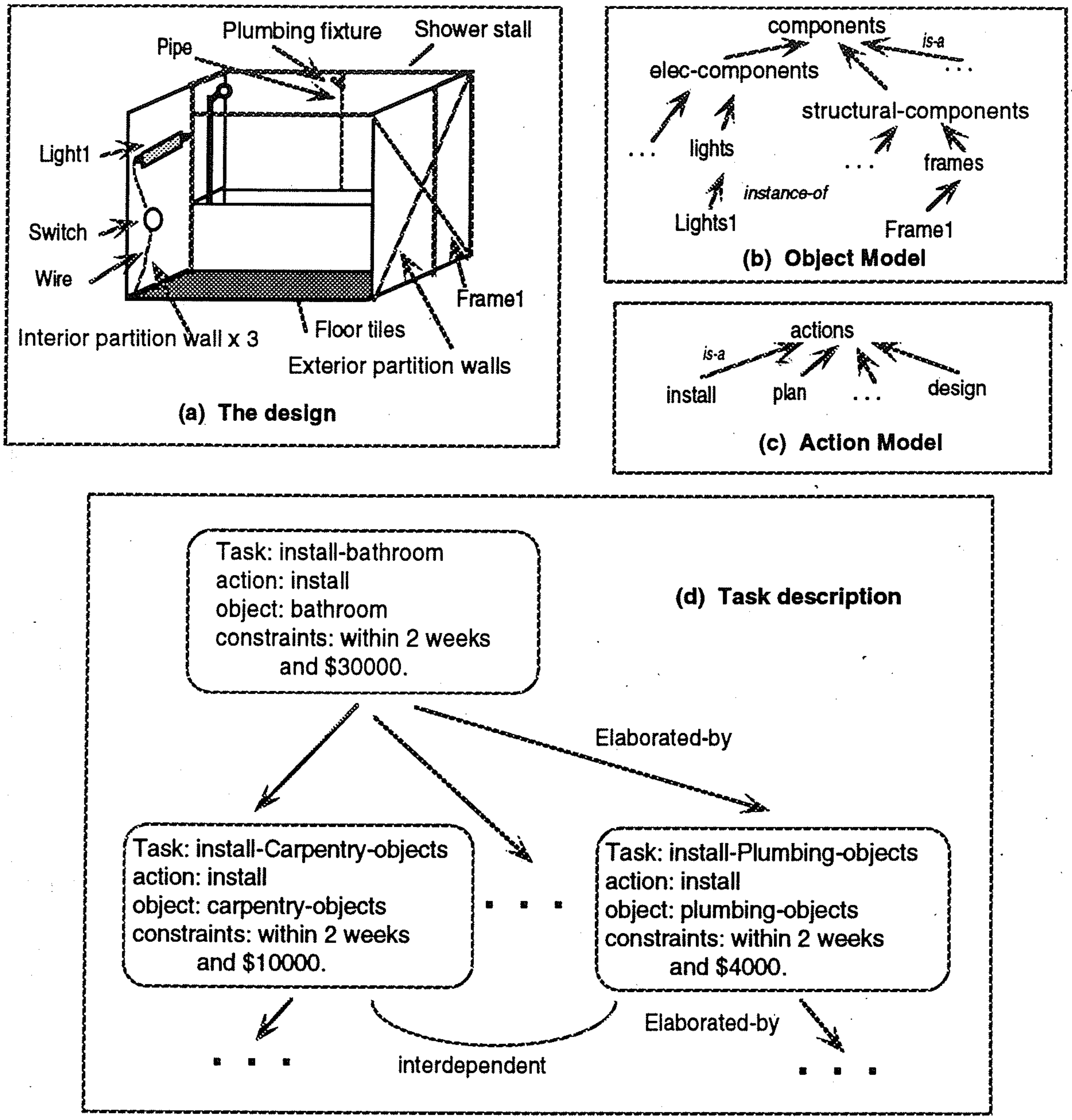

Figure 2: Task example: A bathroom project

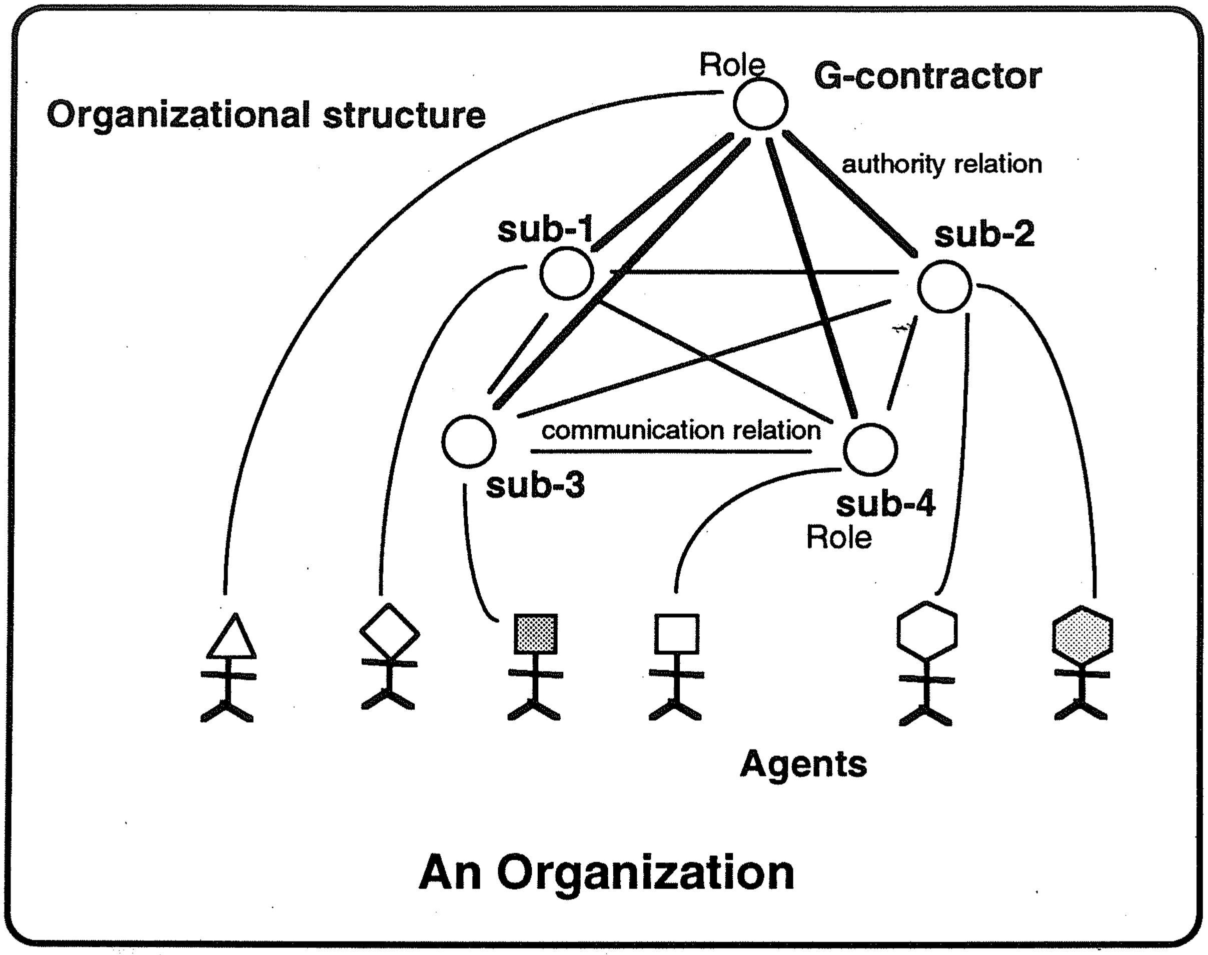

Figure 4: Role-based organizational structuring

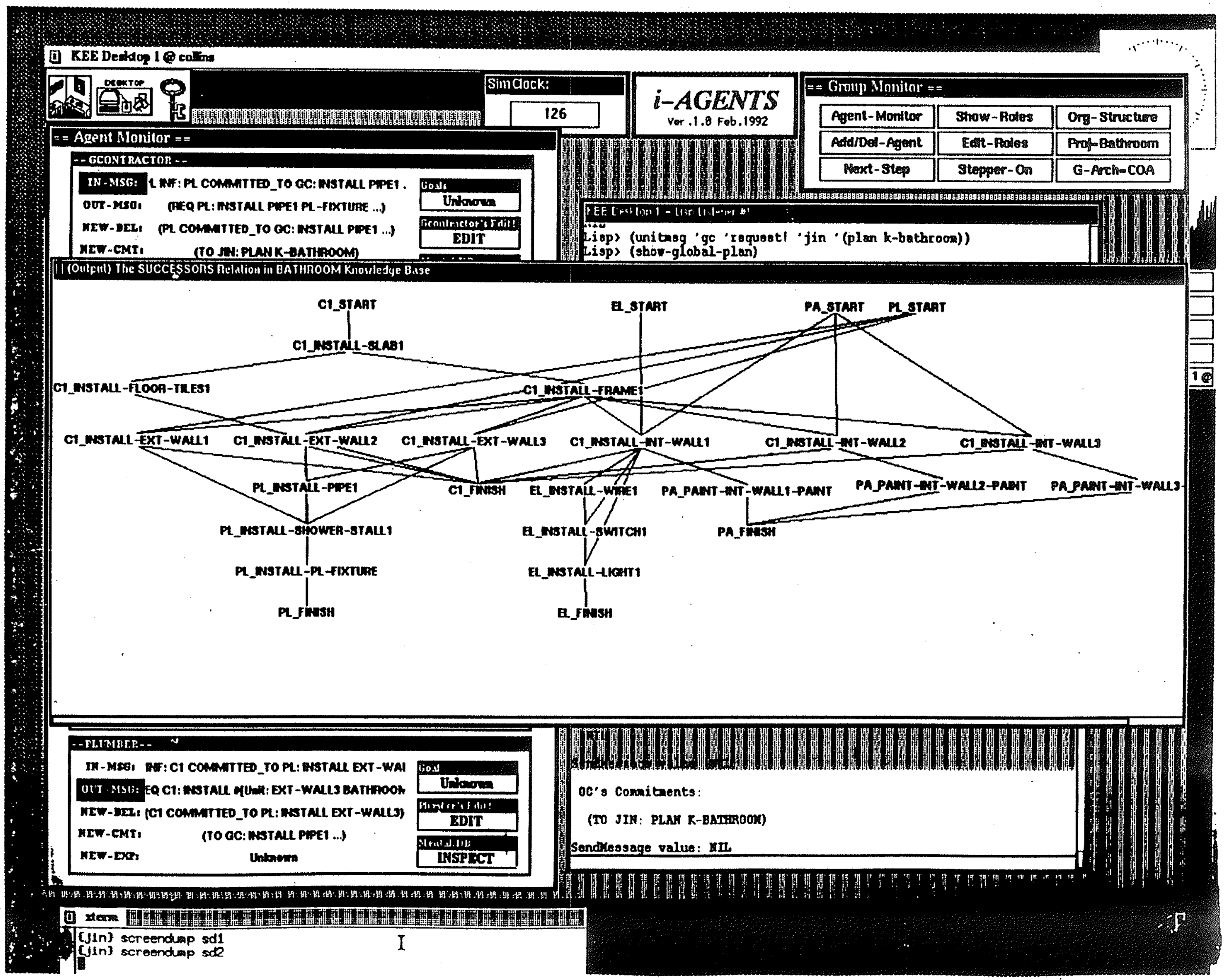

Figure 6: The multiagent plan developed by the agent-based organization for the bathroom project