From: AAAI Technical Report SS-94-07. Compilation copyright © 1994, AAAI (www.aaai.org). All rights reserved.

Task Environment Centered Design of Organizations *

Keith Decker and Victor Lesser
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
Email: DECKER@CS.UMASS.EDU

Abstract

The design of organizations or other coordination mechanisms for groups of computational agents, either interacting with one another or with people, depends crucially on the task environment of which they are a part. Such dependencies include the structure of the environment (the particular kinds and patterns of interrelationships that occur between tasks) and the uncertainty in the environment (both in the *a priori* structure of any episode within an environment and in the outcomes of an agent's actions). Designing organizations also depends on properties of the agents themselves—but this has been studied more thoroughly by other researchers. The central idea is that the design of coordination mechanisms cannot rely on the principled construction of agents alone, but must rely on the structure and other characteristics of the task environment—for example, the presence of uncertainty and concomitant high variance in a structure. Furthermore, this structure can and should be used as the central guide to the design of coordination mechanisms, and thus must be a part of any comprehensive theory of coordination.

This working paper will briefly describe our modeling framework, TÆMS, for representing abstract task environments. We will also briefly describe a family of domain-independent, team-oriented coordination algorithms called Generalized partial Global Planning (GPGP). Having a family of algorithms allows us to tailor the algorithm to the environment (or even a specific situation). We will give an example of an analysis inspired by Burton and Obel's work on organizational structure and technology decomposability.

Introduction

We have developed a task environment-oriented modeling framework called TÆMS (Task Analysis, Environment Modeling, and Simulation) that features careful attention to the quantitative computational interrelationships between tasks, to what information is available (and when) to update an agent's mental state, and to the general structure of the task environment rather than single-instance examples [Decker and Lesser, 1993d; Decker and Lesser, 1994b]. Our task environment models can be used for both the analysis and simulation of coordination algorithms, and also to design organizational structures that are well-adapted to particular task environments.

The form of the TÆMS framework is more detailed in structure than many organizational-theoretic models of organizational environments, such as Thompson's notions of pooled, sequential, and reciprocal processes [Thompson, 1967], Burton and Obel's linear programs [Burton and Obel, 1984], or Malone's queueing models [Malone, 1987]. It is influenced by the importance of environmental uncertainty and dependency that appear in contingency-theoretic and open systems views of organizations [Lawrence and Lorsch, 1967; Galbraith, 1977; Scott, 1987; Stinchcombe, 1990]. TÆMS allows the quantitative operationalization of many organizational-theoretic environmental concepts, especially various dependencies and uncertainties (see the decomposability example, from [Burton and Obel, 1984], later in this paper) that are the basis of both contingency theoretic and transaction cost economic (i.e. [Williamson, 1975; Moe, 1984]) views of organizations. The observation that no single organization or coordination mechanism is 'the best' across environments, problem-solving instances, or even particular situations is also common in the study of cooperative distributed problem solving [Fox, 1981; Durfee *et al.*, 1987; Durfee and Montgomery, 1991; Decker and Lesser, 1993b]. Key features of task environments demonstrated in both these threads of work that lead to different coordination mechanisms include those related to the structure of the environment (what we will call **task interrelationships**) and environmental **uncertainty**.

Uncertainty. Less uncertainty in the environment means less uncertainty in the existence and extent of the task interdependencies, and less uncertainty in local scheduling—therefore the agents need less complex coordination behaviors (communication, negotiation, partial plans, etc) For example, one can have cooperation without communication [Genesereth *et al.*, 1986] if certain facts are known about the task structure by all agents. Sometimes agents will not have *a priori* information about the structure of an episode, but they will be able to get information about a subset of the structure after the start of problem solving—no single agent working alone will be able to construct a view of the entire

---

*This work was supported by DARPA contract N00014-92-J-1698, Office of Naval Research contract N00014-92-J-1450, and NSF contract CDA 8922572. The content of the information does not necessarily reflect the position or the policy of the Government and no official endorsement should be inferred.

problem facing the group. Some of this uncertainty may arise from disparities in the objective (real) task structure and the subjective (agent-believed) task structure, either initially or as the situation evolves over time.

Task interrelationships. Task interrelationships include the relationships of tasks to the performance criteria by which we will evaluate a system, to the control decision structures of the agents which make up a system, and to the performance of other tasks. The TÆMS objective `subtask` relationship indicates the relationship of tasks to system performance criteria; the subjective mapping indicates the initial beliefs available to an agent's control decision structures; various non-local effects such as `enables` and `facilitates` indicate how the execution of one task affects the duration or quality of another task. Each relationship is quantitatively defined. When a relationship extends between parts of a task structure that are subjectively believed by different agents, we call it a *coordination relationship.* The basic idea behind Generalized Partial Global Planning (GPGP) is to detect and respond appropriately to these quantitative coordination relationships.

The main thrust of this working paper is to present TÆMS as a framework that can be used for computational organization design; the remainder of this paper will give a brief overview of TÆMS, a simple architecture for agents working in a TÆMS environment, a brief overview of the GPGP family of team-oriented coordination algorithms, and an example of a simple analysis. Finally we will discuss how we can extend our analyses to more complex, hierarchical organizational structures.

Short Overview of TÆMS

The principle purpose of a TÆMS model is to analyze, explain, or predict the performance of a system or some component. While TÆMS does not establish a particular performance criteria, it focuses on providing two kinds of multi-criteria performance information: the temporal intervals of task executions, and the *quality* of the execution or its result. *Quality* is an intentionally vaguely-defined term that must be instantiated for a particular environment and performance criteria. Examples of *quality* measures (which can be vectors) include the precision, belief, or completeness of a task result. TÆMS models describe how several quantities—the quality produced by executing a task, the time taken to perform that task, the time when a task can be started, its deadline, and whether the task is necessary at all—are affected by the execution of other tasks.

A TÆMS model of environmental and task characteristics has three levels: *objective*, *subjective*, and *generative*. The *objective* level describes the essential, 'real' task structure of a particular problem-solving situation or instance over time. The *subjective* level describes the agents' view of the situation. A subjective level model is essential for evaluating coordination algorithms, because while individual behavior and system performance can be measured objectively, agents must make decisions with only subjective information.[1] Finally, the *generative* level describes the statistical characteristics of the objective and subjective situations in a domain. A generative level model consists of a description of the generative processes or distributions from which the range of alternative problem instances can be derived, and is used to study performance over a range of problems in an environment.

A problem instance (called an *episode* $\mathbf{E}$) is defined as a set of task groups, each with a deadline $\mathrm{D}(\mathcal{T})$, such as $\mathbf{E} = \langle \mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n \rangle$. The task groups (or subtasks within them) may arrive at different times. While task groups are independent of one another computationally[2], the tasks within a single task group are in general not independent. Figure 2 shows two objective task groups, and Figure 3 shows agent A's initial subjective view of that task group. A task group consists of a set of tasks related to one another by a `subtask` relationship that forms an acyclic graph (here, a tree). The circles higher up in the tree represent various subtasks involved in the task group, and indicate precisely how quality will accrue depending on what leaf tasks are executed and when. Tasks at the leaves of the tree (without subtasks) represent *methods*, which are the actual computations or actions the agent will execute (in the figure, these are shown as boxes). The arrows between tasks and/or methods indicate other task interrelationships where the execution of some method will have a positive or negative effect on the quality or duration of another method. The presence of these interrelationships make this an NP-hard scheduling problem; further complicating factors for an agent's local scheduler include the fact that multiple agents are executing related methods, that some methods are redundant (executable at more than one agent), and that the subjective task structure may differ from the real objective structure. This notation and associated semantics are formally defined in [Decker and Lesser, 1993d; Decker and Lesser, 1994b].

Hospital Patient Scheduling Example

Let's look at a brief example of a TÆMS task structure model in terms of its ability to reason about organizational decision making. The following description is from an actual case study [Ow *et al.*, 1989]:

> *Patients in General Hospital reside in* units *that are organized by branches of medicine, such as orthopedics or neurosurgery. Each day, physicians request certain tests and/or therapy to be performed as a part of the diagnosis and treatment of a patient. [...] Tests are performed by separate, independent, and distally located* ancillary departments *in the hospital. The radiology department, for example, provides X-ray services and may receive requests from a number of different units in the hospital.*

Furthermore, each test may interact with other tests in relationships such as `enables`, `requires-delay` (must be performed after), and `inhibits` (test A's performance invalidates test B's result if A is performed during specified time period relative to B). Note that the unit secretaries (as scheduling agents) try to minimize the patients' stays in the hospital, while the ancillary secretaries (as scheduling agents) try to maximize equipment use (throughput) and minimize setup times.

---

[1] In organizational theoretic terms, subjective *perception* can be used to predict agent actions or *outputs*, while unperceived, objective environmental characteristics affect performance (or *outcomes*) [Scott, 1987].

[2] Except for the use of the processor(s) or other physical resources [Decker and Lesser, 1994b].

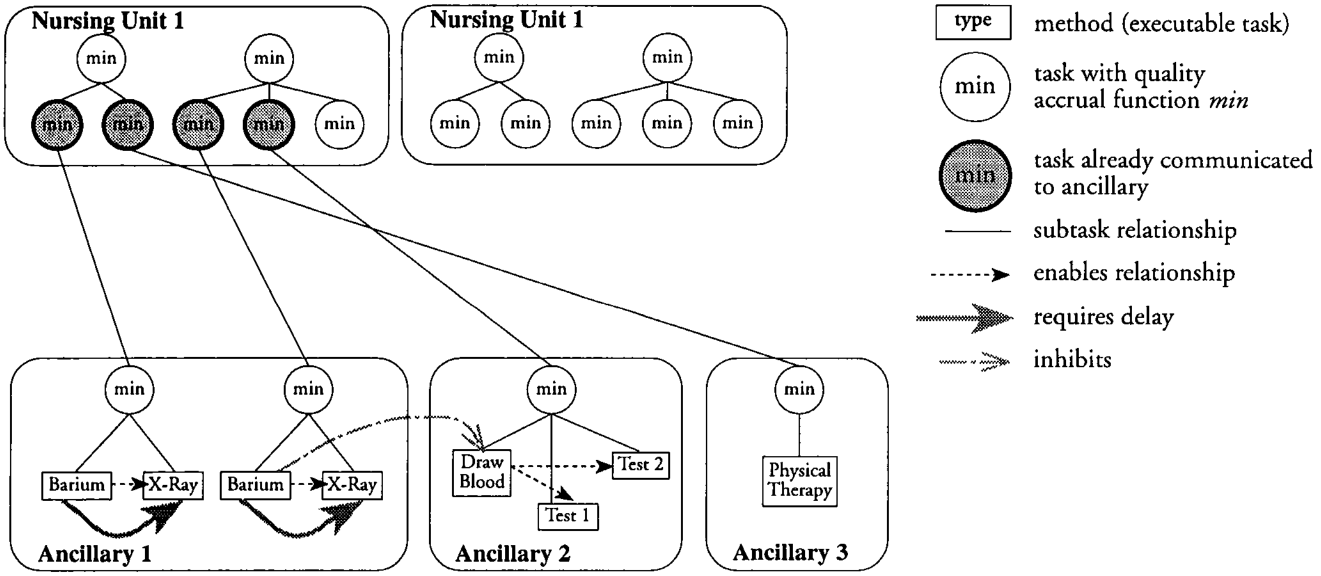


Figure 1: High-level, subjective task structure for a typical hospital patient scheduling episode. The top task in each ancillary is really the same objective entity as the unit task it is linked to in the diagram.

Figure 1 shows an subjective TÆMS task structure corresponding to an episode in this domain, and the subjective views of the unit and ancillary scheduling agents after four tests have been ordered. Note that quite a bit of detail can be captured in just the 'computational' aspects of the environment—in this case, the tasks use peoples' time, not a computer's. However, TÆMS can model in more detail the physical resources and job shop characteristics of the ancillaries if necessary [Decker and Lesser, 1994b]. Such detail is not necessary for us to analyze the protocols developed by [Ow *et al.*, 1989], who propose a primary unit-ancillary protocol and a secondary ancillary-ancillary protocol.

We use *min* (AND) to represent quality accrual because in general neither the nursing units nor ancillaries can change the doctor's orders—all tests must be done as prescribed. We have added two new non-local effects: requires – delay and inhibits. The first effect says that a certain amount δ of time must pass after executing one method before the second is enabled. The second relationship, A inhibits B, means that B will not produce any quality if executed in a certain window of time relative to the execution of A, and can be defined in a similar manner.

Analysis in TÆMS

The methodology we have been building uses the TÆMS framework and other DAI formalisms to build and chain together statistical models of coordination behavior that focus on the sources of uncertainty or variance in the environment and agents, and their effect on the (potentially multi-criteria) performance of the agents. For example, we have used this methodology to develop expressions for the expected value of, and confidence intervals on, the time of termination of a set of agents in any arbitrary simple distributed sensor network environment that has a static organizational structure and coordination algorithm [Decker and Lesser, 1993b]. We have also used this model to analyze a dynamic, one-shot reorganization algorithm (and have shown when the extra overhead is worthwhile versus the static algorithm) [Decker and Lesser, 1993c]. In each case we can predict the effects of adding more agents, changing the relative cost of communication and computation, and changing how the agents are organized (in this case, by changing the range and overlap of their capabilities). These results were achieved by direct mathematical analysis of the model and verified through simulation in TÆMS. We will omit the details here and refer the reader to our papers.

An Agent Architecture

The TÆMS framework makes very few assumptions about the architecture of agents—agents are loci of belief and action. Agents have some control mechanism that decides on actions given the agent's current beliefs. There are three classes of actions: method execution, communication, and information gathering. Method execution actions cause quality to accrue in a task group (as indicated by the task structure). Communication actions are used to send the results of method executions (which in turn may trigger the effects of various task interrelationships) or meta-level information. Information gathering actions add newly arriving task structures, or new communications, to an agent's set of beliefs. Formally, we write $B^t_A(x)$ to mean agent A subjectively believes x at time t [Shoham, 1991]. We will shorten this to $B(x)$ when we don't have a particular agent or time in mind.

The GPGP family of coordination algorithms makes stronger assumptions than TÆMS about the agent architecture. Most importantly, it assumes the presence of a local scheduling mechanism that can decide what method execution actions should take place and when, which will be described in the next section. It assumes that agents do not intentionally lie and that they believe what they are told.

The Local Scheduler

Each GPGP agent contains a local scheduler that takes as input the current, subjectively believed task structure and produces a schedule of what methods to execute and when. It chooses these methods in an attempt to maximize a pre-defined utility measure; in this paper the utility function is the sum of the task group qualities: $\mathrm{U}(\mathbf{E}) = \sum_{\mathcal{T} \in \mathbf{E}} Q(\mathcal{T}, \mathrm{D}(\mathcal{T}))$, where $Q(\mathcal{T}, t)$ denotes the quality of $\mathcal{T}$ at time t as defined in [Decker and Lesser, 1993d].[3]

Beside the subjective task structure, the local scheduler should accept a set of *commitments* **C** from the coordination component. These commitments are extra constraints on the schedules that are produced by the local scheduler. For example, if method 1 is executable by agent A and method 2 is executable by agent B, and the methods are redundant, then agent A's coordination mechanism may *commit* agent A to do method 1 both locally and socially (commitments are directed to particular agents in the sense of [Shoham, 1991; Castelfranchi, 1993]) by communicating this commitment to B (so that agent B's coordination mechanism records agent A's commitment, see the description of non-local commitments **NLC** next).

The final piece of information that is used by the local scheduler is the set of non-local commitments made by other agents **NLC**. This information can be used by the local scheduler to coordinate actions between agents. For example the local scheduler should have the property that, if method M_1 is executable by agent A and enables method M_2 at agent B (and agent B knows this), then if agent A makes a commitment to B to communicate the results of M_1 by time t, B will only schedule M_2 after time t.[4] Thus we may define the local scheduler as a function $\mathrm{LS}(\mathbf{E}, \mathbf{C}, \mathbf{NLC})$ returning a set of schedules $\mathbf{S} = \{S_1, S_2, \ldots, S_m\}$. More detailed information about this kind of interface between the local scheduler and the coordination component may be found in [Garvey *et al.*, 1994].

The GPGP Coordination Modules

The GPGP algorithm family specifies three basic areas of the agent's coordination behavior: how and when to communicate and construct non-local views of the current problem solving situation; how and when to exchange the partial results of problem solving; how and when to make and break *commitments* to other agents about what results will be available and when.[5] The coordination mechanism supplies non-local views of problem solving to the local scheduler, including non-local commitments about *what* non-local results will be available locally, and *when* they will be available.

The five modules described in [Decker and Lesser, 1994a] form a basic set that provides similar functionality to the original partial global planning algorithm as explained in [Decker and Lesser, 1992]. Module 1 exchanges useful private views of task structures; Module 2 communicates results; Module 3 handles redundant methods; Modules 4 and 5 handle hard and soft coordination relationships. More modules have been designed, such as a load-balancing module. The modules are independent in the sense that they can be used in any combination.

Simulation Example

Simulation is a useful tool for learning parameters to control algorithms, for quickly exploring the behavior space of a new control algorithm, and for conducting controlled, repeatable experiments when direct mathematical analysis is unwarranted or too complex. The simulation system we have built for the direct execution of models in the TÆMS framework supports, for example, the collection of paired response data, where different or ablated coordination or local scheduling algorithms can be compared on identical instances of a wide variety of situations (generated using the generative level of the model). We have used simulation to explore the effect of exploiting the presence of *facilitation* between tasks in a multi-agent real-time environment where no quality is accrued after a task's deadline [Decker and Lesser, 1993a]. The environmental generative characteristics here included the mean interarrival time for tasks, the likelihood of one task facilitating another, and the strength of the facilitation (ϕ_d).

In contrast to the previous analytical work, in this section we will consider the use of TÆMS as a simulator to explore hypotheses about the interactions between environmental and agent-structural characteristics. We use as an example a question explored by Burton and Obel: is there a significant difference in performance due to either the choice of organizational structure or the decomposability of technology[6]?

We equate a technology with a TÆMS task structure, instead of a linear program. Task structures allow us to use a clear interval measure for decomposability, namely the probability of a task interrelationship (in this example enables, facilitates, and overlaps). We define a nearly decomposable task structure to have a base probability of 0.2 for these three coordination relationships and a less decomposable task structure to have a base probability of 0.8 (see Figure 2). We will continue in this example to look at purely computational task structures, although the use of physical resources can also be represented (see [Decker and Lesser, 1994b]).

Burton and Obel were exploring the difference in M-form (multidivisional) and U-form (unitary—functional) hierarchical structures; we will analyze the GPGP family of team-oriented coordination algorithms. For our structural variable we will vary the communication of non-local views (GPGP Module 1). Informally, we will be contrasting the situation where each agent makes commitments and communicates results based only on local information (*no* non-local view) with one where the agents freely share task structure information with one another across coordination relationships (*partial* non-local view). Figure 3 shows an example—note

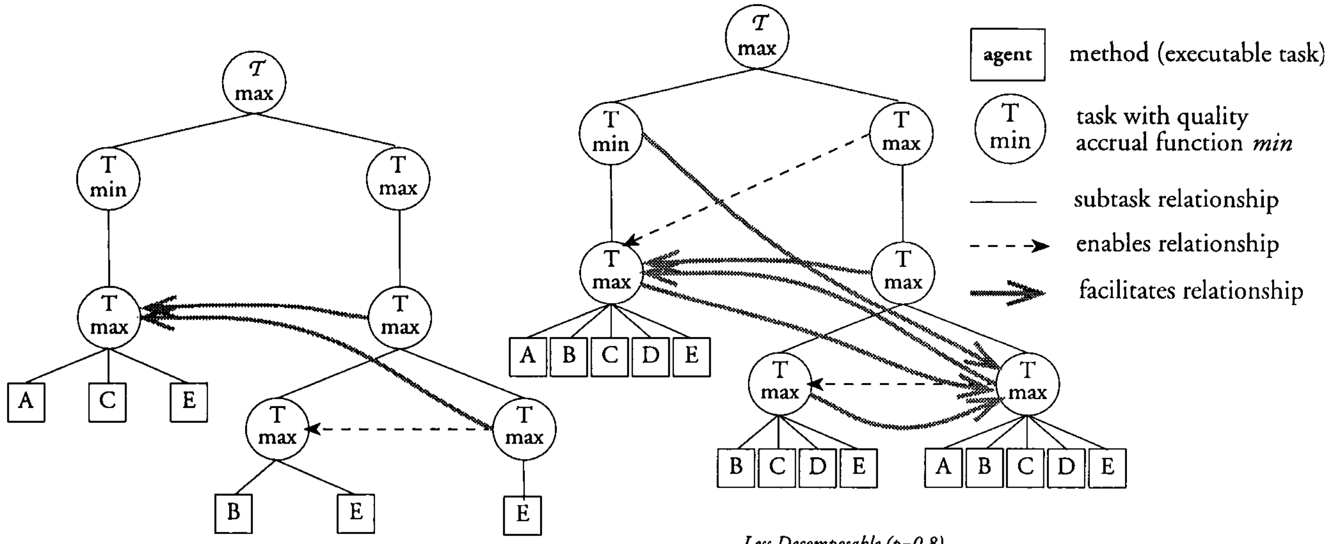
---

[3] Note that quality does not accrue after a task group's deadline.

[4] Of course in general a local scheduler cannot be optimal, and this holds for human agents as well. We can explore other models of local scheduling, prioritization, attention, style, etc. [March and Simon, 1958]. Some work is already being done in this area [Lin and Carley, 1993; J.C. Kunz, 1993].

[5] The use of commitments in the GPGP family of algorithms is based on the ideas of many researchers [Cohen and Levesque, 1990; Shoham, 1991; Castelfranchi, 1993; Jennings, 1993].

[6] *Technology* is used here in the management science sense of "the physical method by which resources are converted into products or services" or a "means for doing work" [Burton and Obel, 1984; Scott, 1987].



More Decomposable (p=0.2)

Less Decomposable (p=0.8)

Figure 2: Example of a randomly generated objective task structure, generated with the parameters in the previous table.

that in neither case does the agent have the *global* view of Figure 2.

Burton and Obel used a profitability index as their performance measure, derived from the percentage of optimal profit achieved. In general, the scheduling an arbitrary TÆMS task structure is an NP-hard problem and so we do not have access to optimal solutions. Instead we compare performance directly on four scales: the number of communication actions, the amount of time spent executing methods, the final quality achieved, and the termination time. Simulation runs for each of the four combinations of non-local view policy and level of task decomposability were done in matched sets—the randomly generated episode was the same for each combination with the exception of more coordination relationships (including more overlapping methods) being added in the less decomposable task structures. Following Burton and Obel, we used the non-parametric Friedman two-way analysis of variance by ranks test for our hypotheses. The assumptions of this test are that each block (in our case, randomly generated episode) is independent of the others and that each block contains matched observations that may be ranked with respect to one another. The null hypothesis is that the populations within each block are identical.

We generated 40 random episodes of a single task group, each episode was replicated for the four combinations in each block. We used teams consisting of 5 agents; the other parameters used in generating the task structures are summarized in Table 1 and a typical randomly generated structure is shown in Figure 2. Figure 3 shows the difference in the local view of one agent with and without creating partial non-local views. We first tested two major hypotheses:

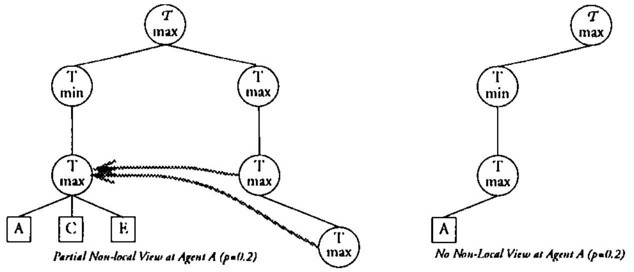
Hypothesis 1: *There is no difference in performance between agents with a partial non-local view and those without.* For the communication and method execution performance measures, we reject the null hypothesis at the 0.001 level. We cannot reject the null hypothesis that there is no difference in final quality and termination time. Teams of computational agents that exchange information about their private, local views consistently exchange more messages (in this experiment, a mean increase of 7 messages) but do less work (here, a mean decrease of 20 time units of work, probably due mostly to avoiding redundancy).

Parameter	Value
Mean Branching factor (Poisson)	1
Mean Depth (Poisson)	3
Mean Duration (exponential)	10
Redundant Method QAF	Max
Number of task groups	1
Task QAF distribution	(50% min) (50% max)
Decomposition parameter	p = 0.2 or 0.8
Hard CR distribution	(p enables) ((1-p) none)
Soft CR distribution	(p facilitates) (10% hinders) ((.9-p) none)
Chance of overlaps (binomial)	p

Table 1: Parameters used to generate the 40 random episodes

Hypothesis 2: *There is no difference in performance due to the level of decomposability of technology.* For the communication and method execution performance measures, we reject the null hypothesis at the 0.001 level. We cannot reject the null hypothesis that there is no difference in final quality and termination time. Teams of computational agents, regardless of their policy on the exchange of private, local information communicate more messages (in this experiment, a mean increase of 47 messages) and do more work (here, a mean increase of 24 time units) when faced with less decomposable computational task structures (technology).

Again following Burton and Obel, we next test for interaction effects between non-local view policy and level of technology decomposability by calculating the differences in performance at each level of decomposability, and then testing across non-local view policy. This test was significant at the 0.05 level for communication, meaning that the difference in the amount of communication under the two policies is itself different depending on whether task decomposability is high or low. This difference is small (two communication actions in this experiment) and was not verified in a second independent test.



Partial Non-local View at Agent A (p=0.2)

No Non-Local View at Agent A (p=0.2)

Figure 3: Example of the local view at Agent A when the team shares private information to create a partial non-local view and when it does not.

Conclusions & Future Directions

TÆMS is a framework for describing complex task environments. When combined with traditional DAI tools for describing coordination algorithms and agents, it provides a basis for analysis and/or simulation on either the TI Explorer Lisp machine or the DEC Alpha AXP. When analyzing an existing or proposed organizational design or coordination algorithm, we advocate an approach that focuses first on behavior due to coordination relationships in certain situations and then expanding this model to incorporate the sources of uncertainty that are present. Such a process may iteratively refine the organization or algorithm—with a parameterized algorithm one might approach this as a pure parameter optimization problem. In [Decker and Lesser, 1993b] we also discuss how the model of performance in a particular episode can be used with meta-level communication to choose a good organizational design dynamically when information about the current situation becomes available to the agents [Stinchcombe, 1990]. On the other hand, when trying to design an organization or coordination algorithm for a given environment, one can also start with both the coordination relationships and the uncertainties present, and add features to deal with each explicitly (our implementation of the GPGP algorithm, which features several independent 'plug-in' modules to deal with different classes of interrelationships, is a case in point). The two-level agent architecture that we use with the GPGP family (with its separate coordination and local scheduling functions) is not necessary for using TÆMS, which views agents only as consumers of subjective beliefs and producers of actions.

Representing Organizational Roles

Future work will include the analysis of more traditional hierarchical organization forms. The addition of the concept of 'organizational role' can be accomplished through the use of non-local commitments (and their accompanying expectations). In its simplest sense, agent A's *organizational role* is a set of continuing non-local commitments to certain classes of tasks. By making continuous commitments agents can avoid communicating about every episode anew—assuming the future structures are somewhat predictable. This reasoning leads naturally to learning organizational roles and to open systems concepts of temporarily settled questions (the current set of continuous commitments) and algorithms to re-open and re-settle these questions as the task environment changes. The development of such algorithms can then lead to exploring conceptions of agents as nothing more than convenient bundles of organizational roles [Gerson, 1976; Gasser *et al.*, 1989], and to when and why organizations do not act like 'big agents' [Allison, 1971]. We will also the expansion of commitments to meta-level roles (i.e., comitments to the coordination parameters in effect for certain classes of tasks).

The Tower of Babel

Up to this point, we have focused on the interrelationships and the uncertainty arising directly from the generative model of the environment, but we would also like to explore the uncertainty and variance arising from the difference in the objective and subjective models. Different relationships between these models will lead to different organizational structures. For example, in the case of uncertainty arising from the generative model of an environment, we showed that the wide variance in performance of a system of agents with static organizations in different episodes led to the use of meta-level communication to reorganize the agents to adapt to the particular episode at hand [Decker and Lesser, 1993c]. We are also looking to expand the TÆMS conception of environments to encompass more dynamic situations: another important source of environmental uncertainty. For example, in the *Tower of Babel* problem [Ishida, 1992], agents try to pile numbered blocks into a tower in numerical order. When many agents try to solve this problem without centralized coordination, they tend to bottleneck around the tower itself, bumping into one another. Ishida solves this problem by dividing the agents into two groups: one to collect blocks and bring them near (but not too near) the tower, and one group to stack the tower. From our point of view, the task structure at low levels is changing rapidly, and no agent can (or needs to) keep track of all the changes in relationships between agent and block positions. The rapid change in the environment means that agents' subjective views are always out of date. By dividing the agents into two groups, the bottleneck resource is controlled by organization—removing it from the uncertainties facing the agents. Stacking agents know there are not enough of them to saturate the bottleneck, and other agents no longer use the resource at all. The uncertainty is still there, but it no longer has an impact on the agents' decision-making process. A similar example is Gasser and Ishida's work on self-organizing production systems [Gasser and Ishida, 1991] where dynamic changes in the environment allowed the system to reorganize itself for more efficient performance. We could potentially analyze either of these systems.

References

Allison, Graham T. 1971. *Essence of Decision: Explaining the Cuban Missle Crisis.* Little, Brown and Company, Boston.

Burton, Richard M. and Obel, Børge 1984. *Designing Efficient Organizations: Modelling and Experimentation.* North Holland, Amsterdam.

Castelfranchi, C. 1993. Commitments:from individual intentions to groups and organizations. In Prietula, Michael, editor 1993, *AI and theories of groups & organizations: Conceptual and Empirical Research.* AAAI Workshop. Working Notes.

Cohen, Philip R. and Levesque, Hector J. 1990. Intention is choice with commitment. *Artificial Intelligence* 42(3).

Decker, Keith S. and Lesser, Victor R. 1992. Generalizing the partial global planning algorithm. *International Journal of Intelligent and Cooperative Information Systems* 1(2):319–346.

Decker, Keith S. and Lesser, Victor R. 1993a. Analyzing a quantitative coordination relationship. *Group Decision and Negotiation* 2(3):195–217.

Decker, Keith S. and Lesser, Victor R. 1993b. An approach to analyzing the need for meta-level communication. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence,* Chambéry, France.

Decker, Keith S. and Lesser, Victor R. 1993c. A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence,* Washington. 210–216.

Decker, Keith S. and Lesser, Victor R. 1993d. Quantitative modeling of complex computational task environments. In *Proceedings of the Eleventh National Conference on Artificial Intelligence,* Washington. 217–224.

Decker, Keith S. and Lesser, Victor R. 1994a. Designing a family of coordination algorithms. Submitted to AAAI-94.

Decker, Keith S. and Lesser, Victor R. 1994b. Quantitative modeling of complex environments. *International Journal of Intelligent Systems in Accounting, Finance, and Management.* To appear in special issue on "Mathematical and Computational Models of Organizations: Models and Characteristics of Agent Behavior".

Durfee, E. H. and Montgomery, T. A. 1991. Coordination as distributed search in a hierarchical behavior space. *IEEE Transactions on Systems, Man, and Cybernetics* 21(6):1363–1378.

Durfee, Edmund H.; Lesser, Victor R.; and Corkill, Daniel D. 1987. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers* 36(11):1275–1291.

Fox, Mark S. 1981. An organizational view of distributed systems. *IEEE Transactions on Systems, Man, and Cybernetics* 11(1):70–80.

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

Garvey, Alan J.; Decker, Keith S.; and Lesser, Victor R. 1994. A negotiation-based interface between a real-time scheduler and a decision-maker. Submitted to AAAI-94.

Gasser, Les and Ishida, Toru 1991. A dynamic organizational architecture for adaptive problem solving. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* Anaheim. 185–190.

Gasser, Les; Rouquette, N. F.; Hill, R. W.; and Lieb, J. 1989. Representing and using organizational knowledge in distributed AI systems. In Gasser, L. and Huhns, M. N., editors 1989, *Distributed Artificial Intelligence, Vol. II.* Pitman Publishing Ltd.

Genesereth, M. R.; Ginsberg, M. L.; and Rosenschein, J. S. 1986. Cooperation without communication. In *Proceedings of the Fifth National Conference on Artificial Intelligence,* Philadelphia, PA. 51–57.

Gerson, Elihu M. 1976. On 'quality of life'. *American Sociological Review* 41:793–806.

Ishida, Toru 1992. Tower of Babel: Towards organization-centered problem solving. In *Proceedings of the 11th Workshop on Distributed Artificial Intelligence,* The Homestead, Michigan. 141–153.

J.C. Kunz, R. E. Levitt G. P. Cohen 1993. Modeling effects of organizational structure and communication tools on design team productivity. In Prietula, Michael, editor 1993, *AI and theories of groups & organizations: Conceptual and Empirical Research.* AAAI Workshop. Working Notes.

Jennings, N. R. 1993. Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review* 8(3):223–250.

Lawrence, Paul and Lorsch, Jay 1967. *Organization and Environment.* Harvard University Press, Cambridge, MA.

Lin, Z. and Carley, K. 1993. Does agent style matter in organizational decision making? In Prietula, Michael, editor 1993, *AI and theories of groups & organizations: Conceptual and Empirical Research.* AAAI Workshop. Working Notes.

Malone, Thomas W. 1987. Modeling coordination in organizations and markets. *Management Science* 33:1317–1332.

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

Moe, Terry M. 1984. The new economics of organization. *American Journal of Political Science* 28(4):739–777.

Ow, P. S.; Prietula, M. J.; and Hsu, W. 1989. Configuring knowledge-based systems to organizational structures: Issues and examples in multiple agent support. In Pau, L. F.; Motiwalla, J.; Pao, Y. H.; and Teh, H. H., editors 1989, *Expert Systems in Economics, Banking, and Management.* North-Holland, Amsterdam. 309–318.

Scott, W. Richard 1987. *Organizations: Rational, Natural, and Open Systems.* Prentice-Hall, Inc., Englewood Cliffs, NJ.

Shoham, Yoav 1991. AGENT0: A simple agent language and its interpreter. In *Proceedings of the Ninth National Conference on Artificial Intelligence,* Anaheim. 704–709.

Stinchcombe, Arthur L. 1990. *Information and Organizations.* University of California Press, Berkeley, CA.

Thompson, James D. 1967. *Organizations in Action.* McGraw-Hill, New York.

Williamson, Oliver E. 1975. *Markets and Hierarchies: Analysis and Antitrust Implications.* The Free Press, New York.