Communication in the Service of Coordination *

Keith S. Decker and Victor R. Lesser
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
Email: decker@cs.umass.edu

May 4, 1994

Abstract

Agent communication can be in the service of many different possible goals, or even many simultaneous goals. One such goal is inter-agent coordination—managing the complex interdependencies between agent activities. Our model of coordination suggests that communicative acts in service of coordination are based on the detection and response to certain *coordination relationships* that exist between interrelated sets of tasks known by multiple agents. These relationships indicate both when results should be communicated ('technical' communication directly in the service of problem solving), as well as when meta-level ('decision-oriented') information should be communicated. We will illustrate this with an example in the hospital scheduling domain. This position paper will briefly discuss how agents may generate communicative goals for coordination, how they may rely on the recipient's inferences to achieve those goals, be sensitive to prior plans and communications, and carry out concurrent execution in time-sensitive domains using Generalized Partial Global Planning (GPGP). GPGP is a family of coordination mechanisms for multi-agent teams that generates communication actions in a domain-independent manner, while working in conjunction with an existing agent planner/scheduler.

1 Our model of multi-agent problem solving

Communicative actions can have technical, 'meta-level' decision-oriented (i.e. economic-rational), political, or symbolic origins [Adler and Borys, 1993]. Technical communicative actions are concerned directly with problem solving; meta-level decision-oriented communicative actions with rational decision making; political communicative actions with power and authority; symbolic communicative actions with expressions of (social) solidarity. Communicative actions may have multiple origins— in this paper we will confine ourselves to technical and decision-oriented meta-level communicative actions. Communication takes time, not only in the transmission of information but in the packaging, interpretation, and assimilation of messages. Such time costs cannot be ignored and must be taken into account when planning or scheduling communication actions in time-sensitive environments. Thus there is an explicit decision process associated with executing communication actions.

Our model of multi-agent problem solving is one of a set of tasks with deadlines that can be solved in multiple ways by multiple agents. The value or quality of a solution to a task or

*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.

related group of tasks may accumulate over time through the actions of the multiple agents. The tasks are related to one another in complex ways involving not only hard constraints (like precedence) but also soft constraints (for example, 'facilitation'—where completing one task before another is helpful but not necessary). In our formalism, such a constraint is called a *coordination relationship* and is defined whenever the execution of some task at one agent changes the duration or possible result quality of another task. For instance enables is a hard coordination relationship indicating that one task must be completed before the second can start. facilitates is a soft coordination relationship that indicates that the result of one task helps in the performance of a second task (making the second task faster, or increasing the quality, or both). hinders is a soft relationship that is the opposite of facilitate. We'll give a few more examples in Section 2.1. One important question is how the agents can individually plan their actions (including communication) to obey the hard task constraints and take advantage of positive soft constraints (and avoid any negative ones). Our solution involves detecting and responding to these coordination relationships.

Deciding when to detect coordination relationships and how to respond to them is crucially dependent on characteristics of both the task environment in which agents find themselves, and the particular problem-solving episode that they are facing. In this paper we will informally use the term *domain* to refer to a general class of more specific task *environments*—we may refer to the distributed sensor network domain, or the job-shop scheduling domain. An *environment* is more specific, and carries with it an implied distribution of potential problem-solving *episodes*—I might refer to the bridge of an AEGIS cruiser, or the Distributed Vehicle Monitoring Testbed (DVMT [Lesser and Corkill, 1983]) as different environments in the distributed sensor network domain; or a particular hospital's patient scheduling organization or a custom VLSI fabrication facility as job shop scheduling environments. An *episode* is a particular coordination problem instance:

- what are the tasks at hand, the methods for accomplishing those tasks, the constraints on those methods, and relationships between these things;
- the set of agents;
- a mapping from tasks, etc. to the knowledge and capabilities of the agents (who knows what? who can **do** what?);
- some set of performance criteria.

For example, the incident that occurred on July 3, 1988 on the bridge of the U.S.S. Vincennes is an episode, with certain tasks related in a certain way that were presented to a particular group of agents with particular knowledge and capabilities. An episode in the VLSI fabrication environment might describe certain equipment that was available, certain production goals and raw materials, and certain equipment and quality control failures that occurred in the episode.

Communication in the service of coordination, then, is all about balancing the costs of discovering, packaging, transmitting, and assimilating meta-level information versus both the *a priori* knowledge of environmental characteristics, and the effect of non-coordinated action in that environment. The need for meta-level communication between agents to detect coordination relationships, for example, is dependent on the environment. In a distributed sensor network interpretation task like the DVMT the relationship between sensing tasks at each agent is different in every episode and needs to be discovered anew each time if it is to be exploited. In a hospital patient scheduling environment certain treatments have consistent and well-known relationships, and the uncertainty lies more in which patients will have multiple interacting treatments. In general, meta-level communication about some coordination relationship depends on:

- The existence of a coordination relationship. As we have just mentioned, in some domains relationships need to be discovered at some cost to the agents, in others they are 'givens'. Nurses in a hospital do not need to rediscover the various necessary scheduling constraints between various standard blood tests, they just need to respond appropriately to those constraints. However, when a new test is developed (or a nurse is trained) these relationships need to be explicitly communicated. Secondary environmental uncertainty characteristics are also used by the agents (a relationship might almost always hold, or almost never hold, or be 'unpredictable').

- The character of the relationship itself. For example, is the facilitation strong or weak? Also, secondary characteristics such as the certainty in the characteristic in question! For example, in some environment, task T_A may always facilitate task T_B, but the strength may vary widely from episode-to-episode.

- The state of the agents themselves. For example, task T_A at Agent A may facilitate task T_B at Agent B, and Agent B would prefer to schedule T_B after receiving the result from A of T_A, but Agent B may be overloaded or need to respond to a deadline that will not allow that schedule to be followed in some particular episode. Secondary environmental characteristics also play a role (i.e., is it likely that Agent B is overloaded? What is the variance on B's load?)

- The impact of a coordination relationship on scheduling. Since coordination relationships are all about orderings, dependencies, negative interactions, and the like, it is not surprising that the communication surrounding them may also include information about the schedules, plans, or future activities of other agents. This becomes especially important when the performance of the system of agents is time-dependent (not just the presence of deadlines, but even performance criteria such as 'solve the problem as soon as possible'). If we continue the task T_A at Agent A and task T_B at Agent B example, then without a temporal performance component it is enough for Agent A and B to know that task T_A facilitates task T_B—Agent B can just wait for the result from Agent A. Under temporal constraints, such as a deadline on the result of task T_B, more information may be useful, such as the time that Agent A will finish task T_A, or from the other side, the latest time by which Agent B can start task T_B and still finish before the deadline. Again, secondary environmental uncertainty characteristics can be important, like the variance in the durations of tasks T_A and T_B.

1.1 Representing Task Environments

We have developed a task environment-oriented modeling framework called TÆMS (Task Analysis, Environment Modeling, and Simulation) that pays careful attention to the quantitative computational interrelationships between agent tasks, to what information is available (and when) to update an agent's mental state, and to the general structure of a task environment rather than single-instance examples. 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. We use TÆMS task structures in this paper to concretely represent domain examples; the notation and formal semantics are covered elsewhere [Decker and Lesser, 1993a, Decker and Lesser, 1993b].

The principle purpose of a TÆMS model is to represent the information available to agents in order 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.

The important features of TÆMS include:

- its layered description of environments (*objective* reality, *subjective* mapping to agent beliefs, *generative* description of the other levels across single problem-solving instances);
- its acceptance of any performance criteria (based on temporal location and *quality* of task executions);
- its non-agent-centered point of view that can be used by researchers working in either formal systems of mental-state-induced behavior or experimental methodologies.

The TÆMS objective subtask relationship indicates the relationship of tasks to system performance criteria; the subjective mapping indicates the what information is available to an agent's control decision structures; various coordination relationships such as enables and facilitates indicate how the execution of one task affects the duration or quality of another task.

2 From Structure to Communication

The task structure clearly delineates *potential* technical communication actions: If the result of task A enables task B, then it would clearly be advantageous to communicate the result of A to the agent performing B. However, how does the first agent know that the relationship exists, or that task B is intended by another agent? Is there some temporal constraint on the communication? If the relationship is a soft one, will it have an effect which is significant enough to warrant its exploitation? Often meta-level decision-oriented information needs to be communicated in order to answer these questions for an agent in uncertain and dynamic environments.

Next we present a brief example showing how the structure of the task affects technical and meta-level communications in a hospital scheduling problem. Then we will briefly discuss Generalized Partial Global Planning, a family of algorithms that provide a set of answers to the questions of what technical and meta-level communications an agent should make, given its current view of the task structure.

2.1 A Brief Example: Hospital Patient Scheduling

Let's look at a brief example of a TÆMS task structure model in terms of its ability to reason about communication actions. Note how the structure of the tasks and the information available about that structure—information that will be available at distinct locations and at

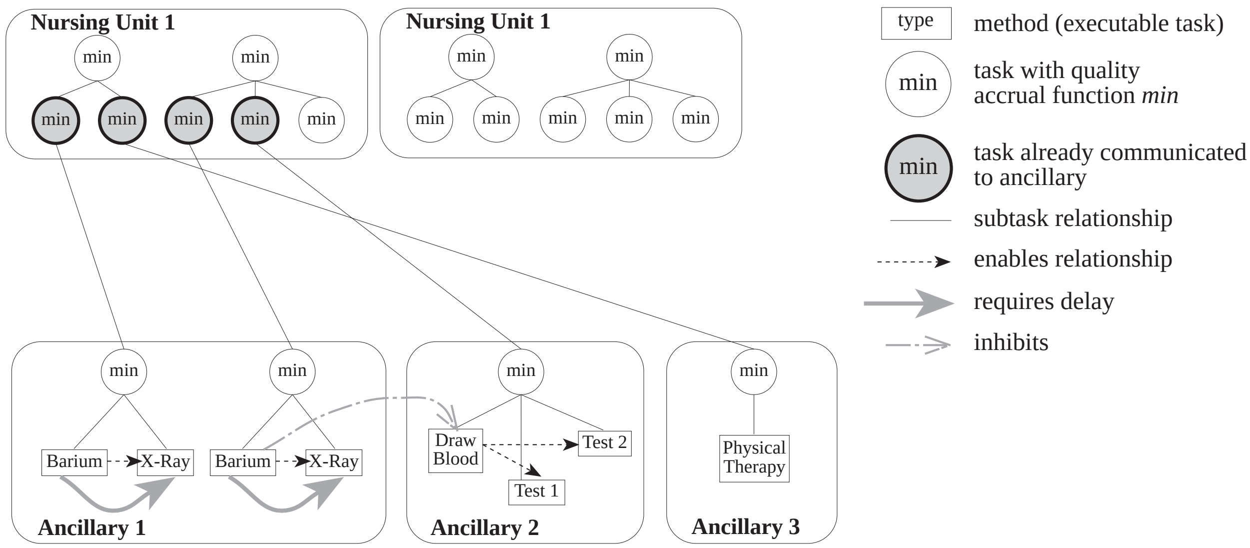

Figure 1: High-level, subjective task structure for a typical hospital patient scheduling episode. The top task in each ancillary is really the same objective entity as the unit task it is linked to in the diagram.

distinct times during problem solving [Stinchcombe, 1990]—may induce meta-level decision-oriented communication between agents with the goal of coordinating multi-agent activity to improve performance. 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.

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. 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. 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, 1993b].

In Ow's case study, he found that the nurses would communicate each patient's request for treatment to each ancillary. Each ancillary would then schedule the procedure immediately, but only notify the nursing unit a short time before the treatment was to occur. Conflicts that occurred were then resolved by the nursing units. This process, which involved only the 'minimal'[1] technical communication necessary could cause many problems, such as multiple ancillaries scheduling a patient for the same or overlapping times, and for ancillaries scheduling tests in violation of ordering constraints. Ow proposes a particular solution to this problem which adds two meta-level decision-oriented communication actions. The first has the ancillary notify the unit nurse as soon as a patient test request is scheduled. The technical request for patient delivery is then obviated, as the new communication serves the dual goals of providing the nurse with scheduling information early to avoid inter-test conflicts, *and* of a request to deliver the patient to the ancillary at the appropriate time. The second meta-level communication has each ancillary, upon scheduling a patient test, broadcast a patient's newly blocked times to the other ancillaries. In section 3 we will discuss a general framework for adding the appropriate meta-level communication actions to problem solving.

3 Generalized Partial Global Planning

When a relationship extends between parts of a task structure that are subjectively believed by different agents, we call it a *coordination relationship*. Remember that a coordination relationship (like facilitates) means that the execution of one task will change the duration and/or potential final quality of another task. The basic idea behind Generalized Partial Global Planning (GPGP) is to detect and respond appropriately to these coordination relationships.

The GPGP family of algorithms specifies three basic areas of the agent's communicated coordination behavior:

- how and when to communicate in order to construct non-local views of the current problem solving situation;
- how and when to communicate 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.[2]

The coordination mechanism supplies non-local views of problem solving to the local planner/scheduler, including non-local commitments about *what* non-local results will be available locally, and *when* they will be available.

The GPGP family is specified and implemented using a series of separate modules (or 'mechanisms', or 'reactions') that detect certain situations and respond with actions that include communication. Each agent comprises a local scheduler and a coordination component. The coordination component at each agent in turn comprises these GPGP modules and a shared decision-making substrate. The reason that separate, parameterizable modules (creating in effect a family of coordination algorithms, not a single fixed algorithm) are used is that no single coordination algorithm will always be the best (or even good) for all environments [Lawrence and Lorsch, 1967, Galbraith, 1977, Stinchcombe, 1990]. We have developed five modules so far, and more will be developed. The modules are not, and do not have to be, entirely independent in their actions because all communication is grounded in the agent's current schedule (they are intended to be used in any combination). If two modules attempt

[1] Minimal in the sense that *if there are no conflicts* only two communication actions are needed: a request to schedule and a request to deliver the patient.

[2] 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].

to make inconsistent commitments to action, only one will remain in the schedule (chosen by the decision-maker substrate, discussed below). We describe the formal specification of the modules, a method to decide if a module is useful in an environment, and an overview of the space of algorithms in [Decker and Lesser, 1994]. In the rest of this paper we will concentrate only on the types of communication each module produces.

The five modules described in [Decker and Lesser, 1994] form a basic set that provides similar functionality to the original partial global planning algorithm as explained in[Decker and Lesser, 1992]. All modules rest on a shared substrate that acts as the agent's central decision-maker, deciding on which plan/schedule to follow and when to terminate processing on a problem. The shared substrate also handles retractions of commitments to other agents when circumstances change and the agent is no longer able locally to keep its commitments. For example, if the agent is committed to completing task T_A by time 14, and now no schedule can be found that meets this commitment (perhaps some other task has taken far too long), then the decision-making substrate will communicate a retraction to all other agents who received the original commitment.

- Module 1 communicates useful private views of task structures: these communications are focused on uncovering previously unknown but important coordination relationships between multiple agent's tasks (important means either hard relationships or strong positive or negative relationships). For example, this module may suspect (or know) that task T_A is related to something (unknown) at agent B. In the vehicle monitoring domain task T_A might be a track extending into the physical area covered by both Agent A and Agent B's sensors. Agent A communicates information about task T_A to Agent B, which may allow the specific coordination relationship to be discovered (or trigger more detailed communication). A gross parameterization of this module allows it to communicate *all* of the local task structure to the other agents (in effect, trying to build a global view); to communicate *some* of the structure (specifically, whenever it suspects an important relationship exists, as in the previous example); or to communicate *none* of the local structure (the agent will still know about some coordination relationships that exist between parts of the task structure that are local to the agent).

- Module 2 communicates *results* in a natural way corresponding to commitments the agent has made to deliver certain results at a certain time. A gross parameterization allows agents to communicate *all* results, *none*, or only those results that have been *committed* to. For example, if Agent A commits to providing the result of task T_A by time 14 to Agent B, then this module can enforce that commitment. Where do commitments come from? See the next three modules...

- Module 3 communicates commitments about redundant methods. When more than one agent can potentially do the same task, any agent who plans to do so commits to it. If more than one agent commits to the same task, all but one agent retracts their commitment. For example, orderlies 1, 2, and 3 can all take patient A to x-ray, and in fact orderlies 1 and 2 plan to do so because it is convenient. Because they know the task is potentially redundant, they tell each other of this commitment. When orderlies 1 and 2 notice they are both committed to take the same patient to x-ray, they resolve the conflict (perhaps a coin toss chooses O1) and the 'loser' (O2) retracts the redundant commitment. The commitment that is communicated here is to *do* something (complete a task with some minimum quality), but not necessarily by a deadline (see Modules 4 and 5).

- Modules 4 and 5 communicate similar commitments about hard and soft coordination relationships. John and Mary both work in a college radio station, an environment notoriously known for a badly-kept music library. If John has scheduled a time to clean and re-alphabetize the music library, and he knows that Mary needs to pull some older library material for her show, he will commit to a certain time at which the library re-arrangement will be finished. Since this facilitates Mary's search, she schedules her time to this non-local commitment (if possible—maybe John can't finish until 5:45 and Mary's on the air at 6). The commitments produced by these modules is to *do* something by a *deadline*, as opposed to just doing it. The difference in the modules is that one responds to hard coordination relationships like enables, and the other to soft relationships like facilitates. Commitments to hard relationships are treated as harder to break by the decision-making substrate.

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. The 'family' of coordination algorithms is outlined by the presence or absence of each module, and what class of tasks or particular situations for which each module is active.

4 Conclusions

This position paper briefly discussed communication actions in service of coordination goals. We first discussed our model of multi-agent problem solving, where tasks at each agent are interrelated by both hard and soft coordination relationships. These relationships indicate potential technical communication actions for each agent. The communications are 'potential' because the information included might be too weak to matter, or might already be known by other agents. Furthermore, the task structure indicates a space of meta-level decision-oriented communications concerning trading non-local views of the task structure in order to uncover previously unknown coordination relationships, and timely commitments to certain actions (including future communications). The necessity of a particular communication action, and the meta-level information contained within it will depend on the existence of a coordination relationship, the quantitative character of the relationship itself (how strong?), the state of the agents (overloaded?), and the impact of the relationship on scheduling. The non-local aspect of communication decision-making in the service of coordination is quite different from that in [Durfee *et al.*, 1987], and more like the communication to resolve solution uncertainty used in DRESUN [Carver *et al.*, 1991]. We gave examples of such 'meta-level' decision-oriented communications in hospital scheduling system. Technical communication (such as patient delivery requests) at each agent was supplanted or extended by meta-level communication for coordination (such as notifying nurses about the patient's future delivery time). Finally, we briefly discussed a family of algorithms for making decisions about (planning) communicative actions in general domains where agents are cooperative and are members of a "team". This indicates that we may study the planning of communication in the service of coordinated behavior independently of a single domain or particular domain planning formalism.

References

[Adler and Borys, 1993] Paul Adler and Bryan Borys. Materialism and idealism in organizational research. *Organization Studies*, 14(5):657–669, 1993.

[Carver *et al.*, 1991] Norman Carver, Zarko Cvetanovic, and Victor Lesser. Sophisticated cooperation in FA/C distributed problem solving systems. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, Anaheim, July 1991.

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

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

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

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

[Decker and Lesser, 1993b] Keith S. Decker and Victor R. Lesser. Quantitative modeling of complex environments. *International Journal of Intelligent Systems in Accounting, Finance, and Management*, 2(4):215–234, December 1993. Special issue on "Mathematical and Computational Models of Organizations: Models and Characteristics of Agent Behavior".

[Decker and Lesser, 1994] Keith S. Decker and Victor R. Lesser. Designing a family of coordination algorithms. Computer Science Technical Report 84-14, University of Massachusetts, 1994.

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

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

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

[Lesser and Corkill, 1983] Victor R. Lesser and Daniel D. Corkill. The distributed vehicle monitoring testbed. *AI Magazine*, 4(3):63–109, Fall 1983.

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

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

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