Toward Generalized Organizationally Contexted Agent Control *†

Thomas Wagner and Victor Lesser
Computer Science Department
University of Massachusetts
wagner@cs.umass.edu

UMass Computer Science Technical Report 1999-18

April 20, 1999

Abstract

Generalized domain-independent approaches to agent control enable control components to be used for a wide variety of applications. This abstraction from the domain context implies that contextual behavior is not possible or that it requires violation of the domain-independent objective. We discuss how context is used in the generalized framework and our current focus on the addition of organizational context in agent control.

1 Introduction

From the vantage point of a long history of research in agents and agent control components for building distributed AI and multi-agent systems, we have focused our recent efforts on approaching agent control from a generalized domain-independent perspective. In implementation terms, the objective is to develop a set of agent control components that can be bundled with domain problem solvers or legacy applications to create agents that can meet real-time deadlines (and real-resource constraints) and coordinate activities with other agents. This paper has two objectives: 1) to describe our generalized approach to agent control and how it lends itself to situation specific conditioning or contextual problem solving, and 2) to describe our recent work in adding organizational knowledge or context to the agent knowledge base and reasoning process.

While generalization and domain independence might seem to be at odds with contextually dependent problem solving, it is in fact through this generalization that we achieve adaptable, contextually appropriate, agent control behaviors. The idea is to construct generalized control problem solvers than can *target* their problem solving behaviors on a particular class of solutions where the class of solutions is contextually dependent. In other words, the control problem solvers are conditioned by a view of the current context. In a typical application, a different component, e.g., domain problem solver or possibly even a contextual evaluation expert, reasons about the larger context of problem solving and abstracts or filters this information, translating it into a generalized set of input parameters to our control components. In terms of a *c-schema* type of approach [17], the context expert could associate particular environmental/contextual conditions with different control parameter settings.[1] The intellectual question that is still unresolved is whether or not this generalized, abstract, approach affords the necessary level of contextual control for all applications.

Our current focus is on the expansion of contextual information used to make control decisions within agents; whether or not this can be done in a generalized fashion remains to be seen. We believe that in order to scale-up agent technology for use in open application domains, e.g., electronic commerce on the web, agents must model their organizational relationships with other agents and reason about the value or utility of interacting and coordinating with particular agents. The

* A version of this paper appears in the *Proceedings of the 1999 AAAI Workshop on Reasoning in Context for AI Systems*.

† This material is based upon work supported by the National Science Foundation under Grant No. EEC-9209623, and by the National Science Foundation under Grant No.s IRI-9523419 and IRI-9634938, the Department of the Navy and Office of the Chief of Naval Research, under Grant No. N00014-95-1-1198. The content of the information does not necessarily reflect the position or the policy of the Government or the National Science Foundation and no official endorsement should be inferred.

[1] There are different graduations or couplings within this model, including *situation specific* selection of different agent coordination mechanisms [14] and contextually preconditioned views of agent activities [2, 7], however, for the purposes of this paper we will paint with a coarse brush and differentiate only when important.

foundation for this is twofold. First, in order to apply coordination technology to large multi-agent systems, it is necessary to control the combinatorial explosion of possibilities. Recall that the assumption of many multi-agent systems is that there is no global view of activities and no centralized agent that selects and schedules all tasks, i.e., coordination is, in a sense, a distributed search process. To control the combinatorics in large multi-agent systems, we look to *organizational structure* to specify which agents interact, which goals agents are likely to coordinate over, etc. The other motivating factor for this current work is that in large, open, multi-agent systems, the social situation in which agents reside is necessarily complex. An agent may belong to multiple different organizations and may have different alliances, power relationships, interaction styles, and so forth with other agents. By representing organizational knowledge, and modeling relationships between agents, we give agents the information necessary to reason about the relative importance of candidate tasks and actions. In other words, the objective is to add contextual knowledge so that agents can *decide* which tasks are more important in a given situation, or which tasks should be given preference.

One facet of this is determining the utility to an agent of different tasks being requested by other agents, and the utility of local problem solving actions, and choosing a balance between these. Another facet is relating self interested interaction with "disinterested 3rd party agents" to work being considered for more altruistic (cooperative) reasons. The issue of action selection and sequencing is obviously quite complex. To further muddy the waters, interaction effects between the candidate actions preclude independent evaluation. Agents situated in large, open environments are in fact embedded in a web of influences [6] that must be considered during decision making. To illustrate, Figure 1 shows an organized network of interacting information agents in the WARREN [4] style. There are three main agent types in the network:

Database Managers Agents that are experts in data maintenance and organization. These agents maintain repositories of information and act as the interface between a repository or digital library and the rest of the network. The repositories may be simple databases, collections of databases, or even entail lower-level database management agents with which the primary database manager interacts.

Information Gathering Specialists These agents are experts in particular domains. For example, one specialist might be an expert on automobiles whereas another might be an expert on software products or weather prediction. These experts know about databases (and database managers) pertaining to their area of expertise, or know how to locate such databases. Their task is to gather information, assimilate it, and produce a report, possibly accompanied by a recommendation to the client about a particular action to take based on the gathered information. These agents receive high level queries or requests for information and in response plan about which sites to query/search and handle the assimilation of the gathered data.

Personal Agents PAs are agents that interface directly with the human client, perhaps modeling the client's needs. These agents also decide with which information specialists to interact to solve a client's information need.

The agents in the network interact in different ways, reflecting their different relationships. Examples of different interactions include: agents from the same company performing services for free, agents belonging to allied companies coordinating by fully disclosing cost and profit information, and agents associated with highly competitive companies haggling furiously before agreeing to cooperate.

In the figure, relationships are denoted by edges between agent nodes; relationship types are expressed via *relationship specifiers* (integers associated with the edges, note the key) that intuitively illustrate the notion of quantified organizational relationships. Some relationships are influenced by corporate connections, e.g., the database manager agents for company X are mutually cooperative and they extend a slightly lesser degree of cooperative behavior to the agents belonging to company Y, a subsidiary of X. In contrast, the database manager for company Z will not service requests from the Microsoft information gathering (IG) specialist. A different type of relationship is that between the IG specialist for company B and the IG specialist for company A – they have a good professional relationship and will cooperate, doing tasks for free, with one another as long as the tasks are not too large or occur too frequently. The different relationships translate into a difficult issue at the control level, namely, how to evaluate different problem solving options. This issue is what motivates the need for organizational modeling and context in the local agent decision process. For example, in Figure 2, how would the IG specialist for company B decide between completing a request made by the company K personal information agent, and servicing requests from the two company B users? What if a Microsoft query arrives in the next timestep? The IG specialist may not be partial to Microsoft, but, perhaps Microsoft is paying a considerable amount for service. How does this financial value compare to the financial reward for servicing the request from the company K agent to the inter-corporate goodwill obtained by processing the inter-company requests? Our objective is to represent such situations and address the control issues that arise from the new contextual information.

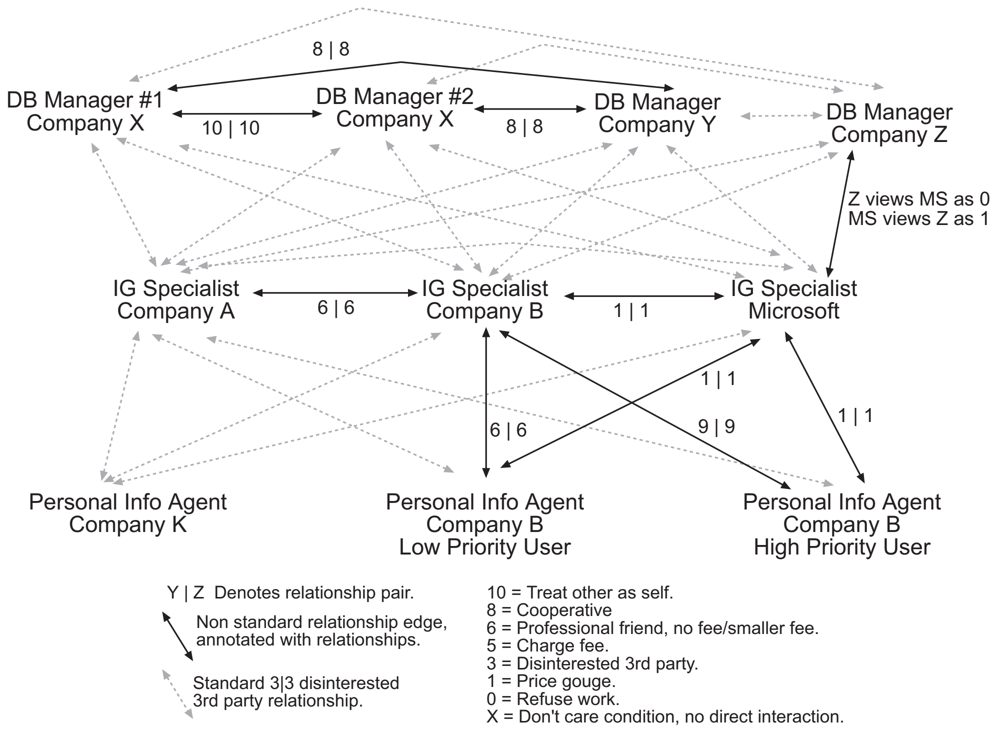

Figure 1: An Organized Network of Interacting Agents

This new emphasis on adding organizational context to agent control is related to recent research in social commitment [3] and obligations between agents [1], but differs in its quantification of agent actions and the reasoning about these actions in a dynamic context.

In the remainder of this paper, we briefly discuss our generalized approach to agent control, and how the control components can be adapted or modulated for different problem solving contexts. We also describe how the agent control components are typically used in an agent and then return to the issue of the integration of organizational context in the agent control process. Due to obvious space limitations the discussion is limited. Interested readers should consult [18, 13, 11] or the group web pages [9] for more information on the various technologies.

2 Generalized Agent Control

We approach the agent control problem from a domain independent perspective. Domain problem solvers, be they process program environments, sophisticated problem solvers, or planners, are coupled with a domain independent task modeling language, TÆMS [5], and modules for agent coordination (GPGP/*GPGP2*), agent scheduling (Design-to-Criteria), and possibly components for learning [16] and diagnosis [8]. The problem solvers translate their internal representations into TÆMS, possibly at some level of abstraction, and these structures are passed to the control components. The full prototypical agent architecture is shown in [13] though in this paper we will concentrate on the local agent scheduler, the multi-agent coordination module, and the TÆMS task description language.

2.1 TÆMS Task Models

TÆMS (Task Analysis, Environment Modeling, and Simulation) is a domain independent task modeling framework used to describe and reason about complex problem solving processes. TÆMS models are used in multi-agent coordination research [5] and are being used in many research projects, including information gathering [12], intelligent environments [10], and others. TÆMS models are hierarchical abstractions of problem solving processes that describe alternative ways of accomplishing a desired goal; they represent major tasks and major decision points, interactions between tasks, and resource constraints but they do not describe the intimate details of each primitive action. All primitive actions in TÆMS are

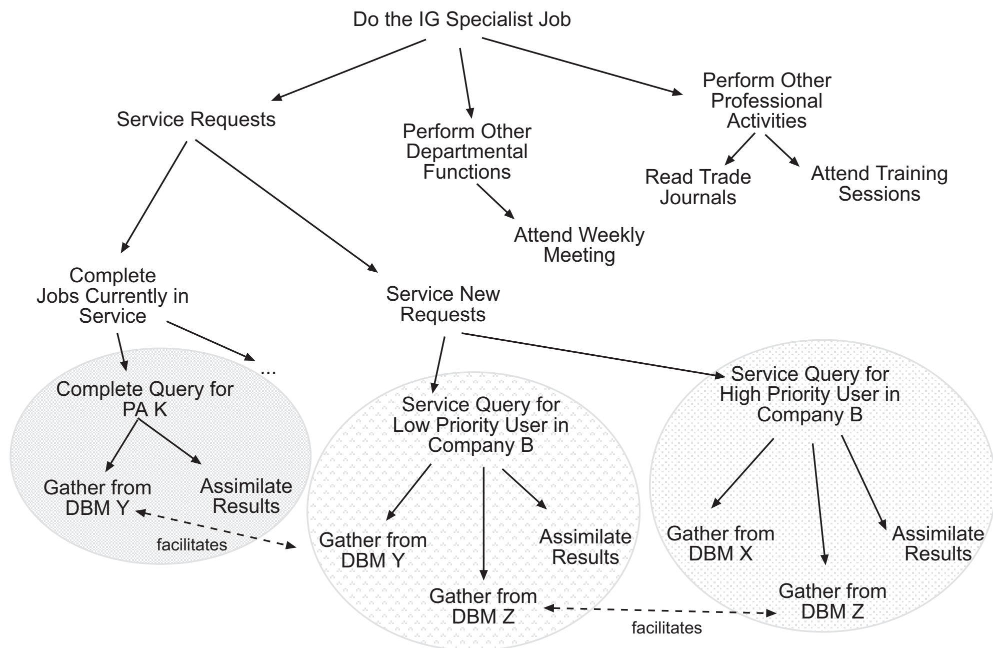

Figure 2: Relating Context to Action Selection

statistically characterized via discrete probability distributions in three dimensions: quality, cost and duration. Uncertainty in each of these dimensions is implicit in the performance characterization – thus agents can reason about the certainty of particular actions as well as their quality, cost, and duration trade-offs. The uncertainty representation is also applied to task interactions like enablement, facilitation and hindering effects.

Figure 3 shows a conceptual, simplified sub-graph of a task structure emitted by the BIG [12] information gathering agent; it describes a portion of the information gathering process. The top-level task is to construct product models of retail PC systems. Space precludes a detailed discussion, but the task structure represents a total of nine different ways to achieve the top level objective, and each different way has different statistical characteristics and represents different trade-offs. A notable feature of the task structure is the *enables* arc between *Get-Basic* and *Gather.* The arc represents a non-local-effect (nle) or task interaction; it models the fact that the review gathering methods need the names of products in order to gather reviews for them. Many different task interactions are modeled in TÆMS, e.g., facilitation and hindering effects, and these are of particular interest to multi-agent coordination as they identify interdependence and a form of joint goals.

2.2 Agent Scheduling and Coordination

In our work, each agent is comprised of multiple control problem solvers that are bundled with one or more domain experts. The primary agent control components are the Design-to-Criteria (DTC) agent scheduler and the GPGP coordination module. The Design-to-Criteria scheduler is the agent's local expert on making control decisions. The scheduler's role is to consider the possible domain actions enumerated by the domain problem solver (via TÆMS) and choose a course of action that best addresses: 1) the local agent's design/goal criteria (its preferences for certain types of solutions), 2) the local agent's resource constraints and environmental circumstances, and 3) the non-local considerations expressed by the GPGP coordination module. Note that since TÆMS models alternative different ways to perform tasks (possibly independent), the scheduling problem entails *choosing* which tasks to perform and *how* to perform them. The general idea is to evaluate the context in which the agent is operating and to custom tailor a schedule for the agent to meet the context. For example, if an agent is in a time critical situation, but has ample financial resources, the scheduler may produce a costly solution (e.g., buying intermediate results from an online database or another agent) that requires less time to execute than a lower-price solution.

GPGP (Generalized Partial Global Planning) [5, 11] is the agent's tool for interacting with other agents and coordinating joint activity. GPGP is a modularized, domain independent, approach to scheduling-centric coordination. The GPGP coordination module is responsible for communicating with other agents and making and breaking task related commitments

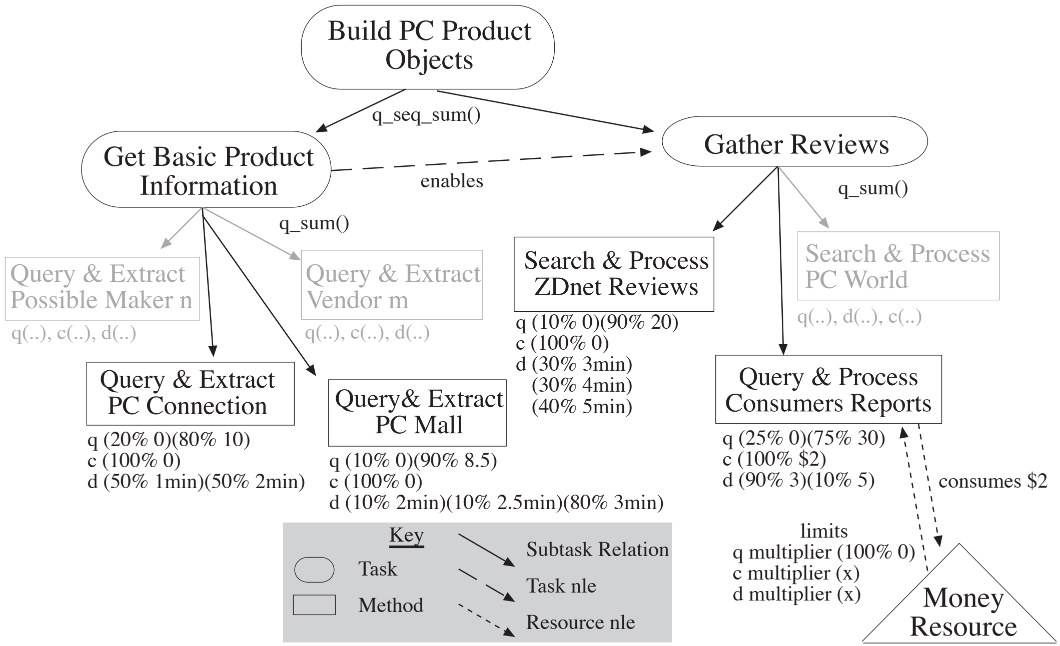

Figure 3: Simplified Information Gathering Task Structure

with other agents. GPGP *modulates* local control by placing constraints, the commitments, on the local DTC scheduler. Commitments represent either deals that GPGP has made with other agents, e.g., agreeing to perform method M by time t, or deals that GPGP is considering making with other agents. The commitments fall into four categories: *deadline*, *earliest start time*, *do*, and *don't*. GPGP consists of several coordination mechanisms, subsets of which may be applied during coordination depending on the degree of coordination desired or the class of interactions that are selected for coordination.

Both DTC and GPGP are designed to facilitate contextual control problem solving. In DTC, this takes the form of analyzing different quality, cost, duration, and uncertainty, trade-offs of different possible solution paths and selecting the solution path appropriate for the current situation. In GPGP, this takes the form of the coordination techniques being modular and parametrized so that the agent can coordinate over selected interactions, or selected classes of interactions, using different information exchange policies. The common denominator is that both control problem solvers are designed to be *targetable* in two ways: 1) for different desired solution classes or solutions that exhibit particular characteristics, and 2) for different allotments of computational effort, i.e., in resource bounded situations, both systems can reduce their search space via satisficing and approximation.

While both DTC and GPGP are control experts adept at reasoning about TÆMS task structures and multi-agent interactions, they are, by necessity, removed from the intimate details of the agents' problem solving activities. This is true of the other generic agent control components as well. By abstracting activities and modeling them in TÆMS, we have been able to design components that can be used in a wide variety of domains. However, state information as well as a detailed (process-level?) view of problem solving is required in order to properly evaluate the context in which the agent is situated. This is typically done by the domain expert, however, it could well well take the form of a context manager [17]. Such an expert could associate environmental/contextual conditions with actions that set the control parameters on the generalized components or actions that specify different goal or design criteria. This contextual or situation specific control modulation is important; intuitively no particular control setting is appropriate for all situations and experiments [14] support this concept. Another way in which a context expert can interact with the generic components is to condition [7, 2] the view that GPGP and DTC have of the agent's problem solving actions. Through this means, the external expert can specify which particular interactions are important for coordination in a particular context and which tasks are critical in the particular context. The organizational knowledge discussed in Sections 1 and 3 may well be integrated into the generic agent control framework using an organizational context expert through this conditioning interface.

Interestingly, the interface between the external context expert and the generic control components should also be two way. If the context expert could analyze the situation *a priori* to determine which classes of solutions were feasible, there would be no need for scheduling or coordination. Thus, though the context expert may specify particular preferences or highlight particular interactions for coordination, it may not actually be possible to implement them. In this case the context expert must change the way in which it conditions the TÆMS models for the other components. In a sense, the general agent control components can provide a more detailed contextual model for the expert by trying to implement a particular specified solution.

In this section, we have described the generalized agent control components and how their targetable control behaviors relate to customizing agent problem solving for a particular context. In our work, said context is normally tracked and monitored by a domain dependent component and then abstracted into control parameters for the generic control components, i.e., the components view the problem solving context through their control parameters. In the next section, we discuss our current work at adding organizational knowledge, and organizational context, to the agent control process.

3 Current and Future Directions: Incorporating Organizational Context

As discussed earlier, we believe modeling the organizational and social context in which the agents operate is critical to operating in an open environment. Some of the extensions we are considering include modeling organizational knowledge and organizational goals, as well as enhanced and quantified notions of commitment and joint goals. Space precludes a full description of the structures under consideration, however, the structures specify, or partially specify:

- The (multiple) organizations to which an agent belongs.
- The different organizational roles an agent is likely to perform for the organization(s). An organizational role is a description of the classes of tasks or duties that an agent is likely to carry out. Note that an agent may have multiple roles within a single organization.
- The other agents belonging to the organization.
- The organizational roles (or abstractions thereof) of the organization.
- The relationship between agents within the organization and between member agents and non member agents, i.e., internal to the organization and external.
- The relative importance of tasks carried out to address the needs of the organization. From another view, an objective or utility function for the organization so that the agent can evaluate how important particular tasks are to the organization.
- The importance of a given role is to the agent. (Even if a task is very important to a particular organization, if the agent is only marginally committed to the organization, the relative importance of said task may be very low.)
- The coordination protocols to use in different circumstances.
- The interaction styles for member and non-member agents, e.g., self-interested, altruistic, or ranges within these two behaviors.
- *A priori* joint goals or commitments, i.e., items that are somewhat static and typical for the organization – agents may generally perform certain activities with other agents and these can be specified in the organization role rather than discovered anew each time by the agents.

New contextual information such as organizational knowledge, organizationally centered joint goals or commitments, impacts the agent control equation in two primary ways:

Quantitative Decision Making The action - selection - sequencing problem is one of the central aspects of agent control and coordination. As we have discussed, this is typically handled in our work by the DTC scheduler. The proposed quantified, organizationally centered knowledge structures expand the context that an agent must consider when selecting and sequencing actions. Instead of focusing on quality, cost, time, and uncertainty trade-offs, the agent must evaluate the larger context to determine the *contextually dependent* value of different activities. This is different from actions simply having some inherent or intrinsic value (that is not contextually dependent). The new knowledge must influence the way in which the intrinsic characteristics of the actions are regarded or evaluated. In other words, the new information expands the context in which the value or utility of primitive actions is determined.

Distributed Computation Structure The new knowledge also specifies attributes that structure agent interactions. The types of structural elements contained in the organizational knowledge include: 1) which agents are likely to interact, 2) how they are likely to coordinate or which protocol to use, 3) the tasks over which they are likely to coordinate, 4) how agents of one group relate to agents of another group, etc.; all of which impose new structure on agent coordination. Information of this class can be incorporated into agent coordination protocols to reduce the amount of communication necessary to coordinate agents and to bias or predispose agents toward certain behaviors. This will enable MAS builders to construct intricate networks of agents without facing a combinatorial explosion of the coordination problem.

We make the distinction between the two different uses of the information because they pertain to different aspects of agent control in multi-agent systems. Information that influences the relative, contextually dependent, value of candidate actions (domain actions, coordination actions, communication actions) pertains mostly to the local agent decision process. This change does affect coordination activities, though indirectly, because it ultimately determines which tasks and actions an agent will perform. The other class of information is not directly relevant to the contextual evaluation of actions, though, it contributes to the inherent characteristics of the actions, e.g., a coordination action that involves one other agent will probably be less time consuming than one that involves several agents.

There are many ways that the new information can be leveraged once integrated into the local agent decision process (e.g., collective bargaining, coordination protocols for forming new organizations, etc.). However, the integration of this type of contextual knowledge with the generic agent control components is complicated issue. Due to the complexity of the scheduling and coordination actions, it is unappealing to attempt to further complicate the TÆMS models used in the scheduling/coordination process. In general, we have taken a view of the DTC/GPGP tools as being *feasibility experts* when coupled with other components that manage the domain view as well as a higher-level view of the tasks an agent my choose to pursue. For example, when the tools are coupled with a high-level problem solver that views the world from a software-process perspective, the process expert would translate only a portion of its domain information into TÆMS for scheduling and coordination. It then may use the feedback from our tools to determine if a change to the overall objective is required. In other words, our tools provide the detailed analysis and feasibility reasoning while the process controller manages the detailed domain view and determines high-level system objectives.

It is likely that the quantified, organizationally centered information will be incorporated into the local agent controller in much the same way that the agent components interface with domain experts, i.e., via a second, higher-level, decision process (possibly akin to an organization context expert). Just as the domain expert manages the domain view and only transfers a part of this knowledge to DTC/GPGP, this new higher-level process will be responsible for abstracting and translating part of the organizational context for use by GPGP/DTC. One design path for the new decision process is to view domain actions and control actions from a unified, but more abstract perspective, and to reason about actions in absence of the detailed constraints evaluated by DTC. The higher level process will then rely on DTC to perform detailed analysis and action selection/sequencing, possibly by translating portions of its unified abstract view of the actions into TÆMS for detailed analysis by the scheduler. If this approach is used it is likely that the two decision making components (that are operating at different levels of abstraction) will interface via a two-way question-and-answer mechanism. It is easy to envision the Design-to-Criteria scheduler discovering during scheduling that it needs more detailed information about other candidate actions, possibly because the candidate actions selected by the abstract view cannot be scheduled due to constraints or interactions not dealt with by the more abstract decision process. Similarly, one can also easily see the benefit of the more abstract organizationally-centered decision process querying the scheduler from time to time for detailed analysis as it determines the value of actions. This is akin to Simon's [15] notion of the organizational structure influencing the objective or utility function. We are currently in the process of developing the organizational structures and the accompanying reasoning process.

References

[1] Mihai Barbuceanu. Agents that work in harmony by knowing and fulfiling their obligations. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 89–96, 1998.

[2] Brett Benyo and Victor R. Lesser. Evolving Organizational Designs for Multi-Agent Systems. UMASS Department of Computer Science Technical Report TR-1999-00, March 1999.

[3] Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95)*, pages 41–48, 1995.

[4] K. Decker, A. Pannu, K. Sycara, and M. Williamson. Designing behaviors for information agents. In *Proceedings of the 1st Intl. Conf. on Autonomous Agents*, pages 404–413, Marina del Rey, February 1997.

[5] Keith S. Decker. *Environment Centered Analysis and Design of Coordination Mechanisms*. PhD thesis, University of Massachusetts, 1995.

[6] Les Gasser. Social conceptions of knowledge and action. *Artificial Intelligence*, 47(1):107–138, 1991.

[7] Bryan Horling and Victor Lesser. Using Diagnosis to Learn Contextual Coordination Rules. *Under review*, 1999.

[8] Bryan Horling, Victor Lesser, Regis Vincent, Ana Bazz an, and Ping Xuan. Diagnosis as an Integral Part of Multi-Agent Adaptability. Computer Science Technical Report TR-99-03, University of Massachusetts at Amherst, January 1999.

[9] The multi-agent systems laboratory at the university of massachusetts amherst. http://mas.cs.umass.edu.

[10] Victor Lesser, Michael Atighetchi, Bryan Horling, Brett Benyo, Anita Raja, Regis Vincent, Thomas Wagner, Ping Xuan, and Shelley XQ. Zhang. A Multi-Agent System for Intelligent Environment Control. In *Proceedings of the Third International Conference on Autonomous Agents (Agents99)*, 1999.

[11] Victor Lesser, Keith Decker, Norman Carver, Alan Garvey, Daniel Neiman, Nagendra Prasad, and Thomas Wagner. Evolution of the GPGP Domain-Independent Coordination Framework. Computer Science Technical Report TR-98-05, University of Massachusetts at Amherst, January 1998.

[12] Victor Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ. Zhang. BIG: A resource-bounded information gathering agent. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, July 1998. See also UMass CS Technical Reports 98-03 and 97-34.

[13] Victor R. Lesser. Reflections on the nature of multi-agent coordination and its implications for an agent architecture. *Autonomous Agents and Multi-Agent Systems*, 1(1):89–111, 1998.

[14] M.V. Nagendra Prasad and V.R. Lesser. Learning situation-specific coordination in generalized partial global planning. In *AAAI Spring Symposium on Adaptation, Co-evolution and Learning in Multiagent Systems*, Stanford, March 1996.

[15] Herbert Simon. Organizational objective functions. Personal Conversation, 1997.

[16] Toshi Sugawara and Victor R. Lesser. Learning to improve coordinated actions in cooperative distributed problem-solving environments. *Machine Learning*, 1998.

[17] Roy M. Turner. Context-Mediated Behavior for Intelligent Agents. *International Journal of Human-Computer Studies*, 48(3):307–330, March 1997.

[18] Thomas Wagner, Alan Garvey, and Victor Lesser. Criteria-Directed Heuristic Task Scheduling. *International Journal of Approximate Reasoning, Special Issue on Scheduling*, 19(1-2):91–118, 1998. A version also available as UMASS CS TR-97-59.