Multi-Level Conflict in Multi-Agent Systems *†

Thomas Wagner, Jonathan Shapiro, Ping Xuan, and Victor Lesser

Department of Computer Science
University of Massachusetts at Amherst
wagner@cs.umass.edu

UMass Computer Science Technical Report 1999-17

April 20, 1999

Abstract

Conflict in multi-agent systems is ubiquitous. Research often focuses on the process of resolving conflicts between different agents. We call this the *inter-agent* conflict resolution process. However, in complex problem solving agents the process of resolving conflicts with other agents impacts local problem solving as well as deals made with other agents. This leads to the need for an *intra-agent* conflict resolution process between the agent's coordination mechanism and its local controller. In situations in which conflicts cannot be resolved to produce satisfactory solution paths for the agent, or a set of agents, it may be necessary to move the conflict resolution process to a higher level, a meta level, in which the agents negotiate to possibly revise their (individual and/or joint) high-level objectives. In this paper, we explore these different levels and their interdependence in the context of our research in multi-agent control and coordination.

1 Introduction

Definitions of agency [10, 12, 2, 22, 15, 9, 5, 25] differ, but, from a high-level, agents can be regarded as having multiple goals or tasks, as being rationally bounded, situated in an environment, and being autonomous, that is having a choice of which activities to perform, and when. Autonomy or choice, in conjunction with bounded rationality alone is enough to ensure that conflict in multi-agent systems is ubiquitous. When agents have different goal sets, or are affiliated with different organizational entities (e.g., different corporations, different users), the issue of agent conflict becomes even more pressing. In some sense, all agent communication and interaction is motivated by the need to resolve conflict, by the need to deal with interdependence.

One class of conflicts in multi-agent systems arises from resource scarcity or task interdependencies (task or resource interaction). Conflict resolution typically entails a dialogue or conversation in which agents negotiate over the interaction, generally making an agreement to either avoid the interaction through a change in the planned activities

* A version of this paper appears in the *Proceedings of the 1999 AAAI Workshop on Negotiation in Multi-Agent Systems.*

† This material is based upon work supported by the National Science Foundation under Grant No. IIS-9812755, the Department of the Navy and Office of the Chief of Naval Research, under Grant No. N00014-95-1-1198, and by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory Air Force Materiel Command, USAF, under agreement number F30602-97-1-0249. The content of the information does not necessarily reflect the position or the policy of the Government, the National Science Foundation, ONR, or DARPA/USAF, and no official endorsement should be inferred.

or by temporally sequencing their activities. Conflict resolution may also be “non-verbal,” taking the form of social convention [20] or default knowledge, e.g., the agent to the right has the right-of-way or the least committed agent should replan.[1]

In our research, this process typically takes place through an agent’s local coordination module (GPGP [4] or *GPGP2* [23]) and entails an exchange of local views, a detection of interactions, and negotiation over the interactions, culminating in the formation of a commitment between the involved agents. There several types of commitments, or deals between agents, in *GPGP2* that are used to resolve the conflict: 1) *earliest-start-time*, in which an agent agrees not to perform a task before a specified time, t; 2) *deadline*, in which the committed agent agrees to perform a task before t; 3) *do*, in which the committed agent agrees to perform the task in question; 4) *don’t*, where the committed agent agrees not to perform the specified task during a specified interval. The conversation held by the agents can simply entail the proactive offering of a commitment or a complex dialogue in which constraints are exchanged and proposals explored. For example, agent α may need a result from β by time t, however, α may be holding a resource needed by β during the interval from 0 to t. In this case, the agents must exchange their local information, detect the interaction, and resolve the conflict by α agreeing (via don’t commitment) not to use the resource during the specified interval and β agreeing to provide a result (via deadline commitment) to α by time t. We return to the issue of negotiation at this level in Section 2. It is important to note that through these commitments, the committed agent potentially agrees to change its selected set of tasks or actions, and/or to change when it performs the actions. These changes bring us to another issue in dealing with conflict in MAS, an issue that is in some sense lower-level, but, interdependent with the dialogue held between the agents.

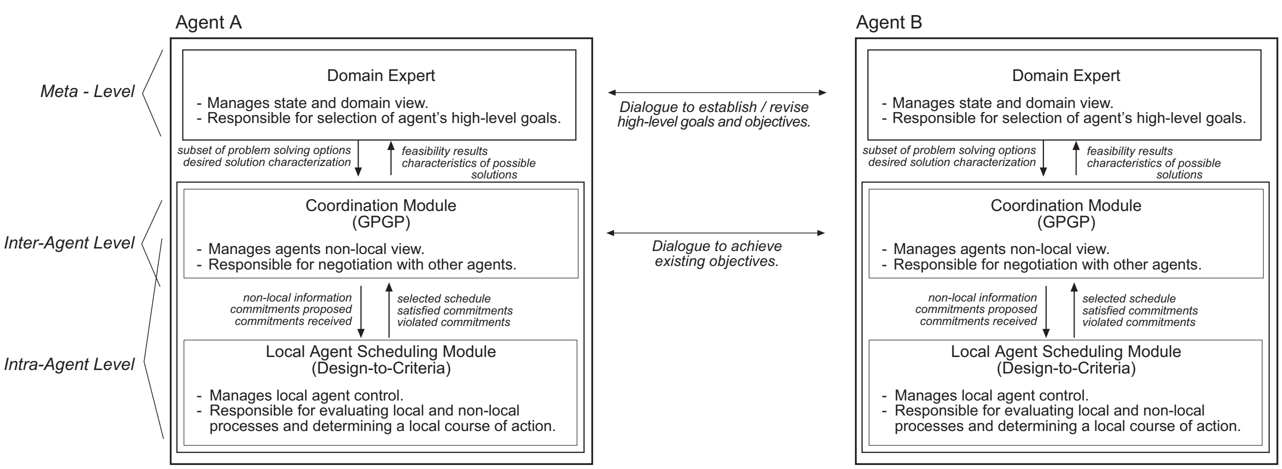

Figure 1: Multiple Interacting Levels of Conflict Resolution

Research often focuses on conflict resolution at the inter-agent level, at the level of the conversation held between one or more agents to handle these task interactions. However, in addition to conflict resolution between agents, the existence of conflicts also results in a need for resolution and negotiation within the agent [8]. If an agent has multiple tasks to perform, or multiple goals to achieve, and it is autonomous (flexible, making its own choices, etc.), the conflict resolution process in which the agent engages with others impacts its pre-existing decisions and its other objectives. In other words, the process of forming a commitment may change the agent’s selected tasks or actions, and it may change the scheduling of the agent’s activities. Unless tasks are entirely independent (including via deadlines or with

[1] Though it is unclear whether certain types of social laws or conventions lend themselves to application with computational processes, i.e., where the world state is less observable or less obvious.

respect to temporal flow)[2], these changes will impact other aspects of the agent's problem solving and possibly affect other commitments made with other agents. For example, if agent α has a local deadline of t for task T_j, and it negotiates with agent β to provide a result for task T_k by time $t+i$, and there is not sufficient time to achieve both, the agent must fail in one of its objectives. In fact, forming a new commitment may impact future commitments as well as existing ones. Stability in this complex situation is generally achieved by tasks having different degrees of importance, or different utilities, and by their associated commitments being similarly quantified. Quantification of commitment in conjunction with costs or penalties for decommitment [18] can ensure a certain level of equitability in the MAS. However, the interplay between the agent's different goals and its interactions with other agents requires a negotiation or dialogue between between the inter-agent negotiation process and the local agent controller, [3] in our work this is the Design-to-Criteria (DTC) scheduler and the GPGP coordination module. There are several facets to this intra-agent negotiation process, and several interaction models for intra-agent conflict resolution. We return to this issue in Section 3.

Hereto we have identified two different levels in the process of conflict resolution in MAS, the intra-agent level and the inter-agent level. To clarify, the intra-agent level entails the process of conflict resolution between the agent coordination mechanism and the local controller. From another view, this is the interaction between local-control and non-local modulation. The inter-agent level, on the other hand, pertains to the dialogue held between one or more agents, the exchange of information, constraints, and the formation of commitments. In our research, we have come to view these two levels as pertaining to feasibility and implementation processes, as the overall goals and objectives of the system are generated elsewhere. In some sense, it is the task of DTC and GPGP to explore the various constraints for a selected set of tasks or goals and to attempt to implement a desirable solution. On the other hand, it is the task of some higher-level controller (e.g., process expert [11] or an information gathering expert [14]) to propose candidate tasks and to perform the domain problem solving activities. In this view, GPGP/DTC are the control problem solving experts while the domain expertise is localized within the domain expert.

This suggests still a third level of conflict resolution, a meta-level, in which the domain experts negotiate to select shared and individual high-level goals and associated objective functions [13]. Recall that agents have multiple goals or tasks; it is entirely possible that through the inter and intra agent negotiation processes (feasibility analysis) that it is not possible to resolve the conflicts satisfactorily to the local agent or some set of agents (depending on whether the model is self-interested or cooperative). In economic terms, it might be impossible for an agent to find a course of action in which its costs outweigh its gains, and this might be the case for all agents in the system[4]. Regardless of the criteria for solution dissatisfaction, in the event of widespread solution dissatisfaction, it may be desirable for the involved agents to *change* the set of tasks or goals that they are pursuing. In other words, feasibility analysis (scheduling and coordination) might not yield any appealing solutions, in which case, the agents may need to move the negotiation to a new level, a meta-level, and change the tasks or goals over which they are negotiating. This may also entail changing the objective function(s) [21] as often the conflict may be resolved by changing the way in which the goals or tasks are evaluated. The different levels, their roles, and their interactions are shown in Figure 1.

In this paper we explore these ideas. Section 2 discusses our recent research in inter-agent negotiation to resolve conflict and Section 3 explores the different negotiation models to resolve conflicts between the coordination module and the local agent controller. Section 4 returns us to the issue of the meta-level and Section 5 discusses future directions in our research.

[2] Note, however, that if tasks are independent there is no need to coordinate and no conflict. We assume a certain level of complexity of agent activities and that the activities have interdependencies.

[3] This also assumes a certain level of complexity in the agent's activities. The general scheduling/coordination problem is exponential and thus agents cannot simply produce the optimal local schedule for a given set of constraints. If that were the case, the coordination module could simply collect constraints, reschedule, and communicate its committed action. There would be no need for negotiation.

[4] Assuming a model in which problem solving is progressive and not governed by zero-sum gains.

2 Inter-Agent Conflict Resolution

In order for agents to do team work, agents need to exchange necessary commitments to each other. To the agent offering the commitment, the commitment imposes constraints on its local activities; while to the receiving agent, the commitment is intended to satisfy constraints so that agent local activities can proceed successfully. Clearly, an agent is in a consistent state if all the constraints (including the constraints due to local commitments) not satisfied through local planning are satisfied by the commitments it received (i.e., non-local commitments). Hence, deciding a set of commitments that leads to all agents in consistent states, is the key to agent coordination.

Conceptually, a centralized planner can search for a set of appropriate commitments if it knows the capabilities, constraints, and goals of each agent, therefore preventing conflict from occurring. However, in a multi-agent system, it may be undesirable, if not impossible, to use a centralized approach. Therefore, agents often need to make decisions locally and exchange commitments in a decentralized manner, with only partial knowledge of each other's belief, desire, and intentions. Due to the inter-dependency of agent tasks and resources (and hence the inter-dependency of commitments), and also the uncertainties associated with agent activities, conflicts in the commitments can often occur, which results in unsatisfied constraints.

To resolve the conflicts, agents need to have a common mechanism to decide what actions to take and how to temporally sequence activities over interactions[5]. A predefined set of social rules may be used for this purpose. However, in our research, we focus on the more general approach which can support a range of approaches from socially defined commitments to dynamic construction of commitments via explicit agent conversation and negotiation. We view the protocols of the conversations as a family of *coordination mechanisms* in the GPGP/GPGP2 framework. In GPGP/GPGP2, a family of coordination mechanisms is defined, covering the exchange of non-local viewpoints and results, the coordination over hard/soft interrelationships, and resource usage. Typically, when a set of agents needs to coordinate over a task or resource interrelationship, they instantiate a conversation process that tries to form commitments that address this interrelationship. Of course, the proposed commitment may depend on another task or resource interrelationship, therefore triggers another conversation process. Also, it is possible that the agents may have conflicting opinions on the proposed commitment(s), thus lead to objections or counter-proposals. In this case, further iterations of the negotiation process may be needed. Our commitment negotiation [19] framework [26] provides a set of message types for the iterative conversation to continue. These message types (conveyed via [7]) allow an agent to specify its intentions [3] regarding the commitment in question: *request, propose, accept, decline, counter-propose, change, no-change*, and *decommit*. There are also other messages that allow dynamic update of the state of the commitment, such as *update*, *fulfilled*, and *failure*. Each coordination mechanism can utilize a subset of these message to construct a conversation protocol that suits its purpose.

The agents have the choice of which set of coordination mechanisms to use, and how much effort they are willing to spend on negotiation, depending on the specific problem solving situation [6]. For example, when two agents are coordinating over an *enables* relationship (meaning task T_α in agent α cannot start until task T_β in some other agent β has finished with non-zero quality), the dialogue can be initiated either by α or β. In one of the coordination mechanisms, agent α can proactively offer a commitment to β because it knows that it is highly likely that β may need this enablement. On the other hand, the agents can also use a reactive mechanism. In this case, β initiates the dialogue by explicitly requesting a deadline commitment from α for task T_α to complete before time t. Agent α then reacts to the request and search for a possible proposal. Depending on how constrained agent α is, it may propose a commitment that promises to finish T_α at a later time t' instead of the requested time t. Agent β then needs to study the received offer and may issue a counter-propose, and in turn agent α may change its proposal. Thus, a new round of negotiation is initiated. The choice may depend on how much time is allowed, or quality and cost requirements, or a mix of these, as well as how much reasoning the agent can do due to the temporal and resource constraints on the problem solving.

[5]This is true for both self-interested interaction and cooperative interaction.

The conversation terminates when the commitment is accepted to all involved agents, when the agents realize that the commitment is unattainable given the constraints of the agents (over-constrained), or when the agents choose to abort the conversation because of the cost of coordination (such as in a time-sensitive environment). As mentioned before, failure to resolve conflict in this level may indicate the need to conflict resolution in other levels.

While the protocols define the ways agents exchange information, undergo negotiation, and reach consensus in a domain-independent way, agents need to have their own domain strategies for negotiation in order for the negotiation process to be effective, efficient, and productive. For example, agents need to know which agents need to make compromises when a conflict occurs? How to realize that a stalemate has occurred? To answer these questions, agents need to have a model of utility in order to reach social decisions. For self-interested agents, agents try to maximize local utilities, but in order to do so the agent may have to negotiate for tasks it cannot do locally and pay the cost of having other agents solving a subproblem and the cost of remote resource accesses. For cooperative agents, the overall goal is to maximize overall utility of the system, which depends on the local utilities of all agents. This means that agents may need to not pursue its locally optimal goal in return for increase of global utility. In either case, negotiation is directed through the exchange of utilities among the agents. In our work, the focus of negotiation is not only on locating agents to satisfy a task/resource constraint but also on the improvement of overall utility produced. We associate a commitment with *marginal costs* [17], namely the utility difference between having and not having the commitment [26]. This way, the commitments can have direct influences over the agent's local decision, therefore drive the negotiation towards a social conclusion.

Complex inter-agent negotiation often involves multi-linked negotiation. The issue of multi-linked negotiation arises when multiple resources have to be acquired in order to solve the problem, and/or when an agent needs other agents to solve subproblems, and they in turn have subproblem and resource interdependency as well as temporal requirements. One example of multi-linked negotiation is a logistic supply chain. To solve the problem in an efficient and flexible manner may require agents to develop organization knowledge and/or share meta-level knowledge about the agent workload and resource usage profile.[6] Also, agents need to be able to perform negotiation in both reactive and proactive manner. For example, if agent α needs a result of task T_β from agent β by time t, but T_β in turn needs resource R which α possesses. If α has the knowledge of the linkage between T_β and the resource, it may proactively offer β the resource at the same time it requests a deadline commitment from β. This would then allow the agents to reduce the amount of iteration needed to complete the negotiation, thus reduces the complexity in negotiation.

3 Intra-Agent Conflict Resolution

One source of conflict between the coordination mechanism and local controller within the agent is the fact that each component is capable of taking a driving role in feasibility analysis, but it is often unclear *a priori* which component is in the better position to move first. The effectiveness of feasibility analysis in complex environments is determined by the agent's ability to balance its need to acquire additional scheduling constraints, such as commitments and non-local information from other agents with its ability to search the space of possible schedules. In our work, we have at various time considered three different patterns of interaction between these two components.

In a common schedule-driven model, a set of goals (task structure) is presented to the local scheduler by the domain expert. In our research, a goal is represented in the TÆMS modeling language as a hierarchical structure of tasks. The scheduler then generates a schedule (and possibly several alternates) based on some prespecified satisficing criteria. This schedule is then passed to the coordination component, which attempts to "implement" the schedule by negotiation. This coordination may take the form of acquiring commitments for resources controlled by other agents, or don't commitments that ensure the availability of some decentralized resource at the scheduled time, or contracting

[6] In general, meta-level information about the larger context in which a particular coordination episode is taking place is beneficial – it can serve to focus coordination activities and supplement communication-based information exchange.

of tasks to other agents. Whatever mechanisms are used, one of two results obtains from the coordination process: either the schedule is implemented or it is not. If the negotiator is unable to implement the schedule, then additional constraints are placed on the scheduling component and it is re-invoked to produce a new schedule. In this model, of course, while the agent is attempting to implement its schedule, other agents are trying to implement their own schedules. Commitments made to other agents by the coordination component are also reflected as constraints on local control. Coordination in this model is driven by the output of the local scheduler and incoming requests from other agents; the local coordination component is reactive.

In a more complex model, the negotiation component may take a more proactive role by attempting to secure critical commitments in advance of generating a complete schedule. Here, the idea is to increase the likelihood that a generated schedule will, in fact, be implemented by, for example, securing highly contested resources early in the process. From a distributed search perspective, we can think of this approach as a form of backtracking avoidance. It can be particularly effective for self-interested agents operating in resource-constrained environments, where the incentive is to negotiate as early as possible. One challenge under this model is to balance the aggressiveness of agents in meeting their individual needs with social concerns such as fairness and global utility. Protocols that allow agents to decommit at a cost are useful here to give agents the freedom to make commitments that they may not actually need while providing incentive to decommit as early as possible [18].

A third model, inspired by recent work in contingency scheduling [24], allows coordination to be tightly integrated with the executing schedule in the form of actions triggered by an execution monitor or even as first class tasks to be scheduled for future execution along with domain-level tasks. For example, a contingency schedule may contain failure recovery actions with certain resource or non-local task requirements. The coordination component must make sure that the appropriate commitments are in place well in advance of a schedule checkpoint at which it will be determined if failure has, in fact, occurred. The agent may choose, however, to trigger a decommitment in the branch of the schedule that does not include the failure recovery option. Similarly, the failure recovery branch may trigger decommitment actions. There are incentives for both cooperative and self-interested to release commitments at the earliest opportunity. For the individual agent, delayed decommitment is a missed opportunity to recover cost. There is also a social cost associated with decommitment if if comes to late for the released resources to be of use to other agents. This balance of cost and time is further complicated by the fact that just as the agent has incentives (both social and individual) to decommit early, there may also be reasons to retain unused commitments when rescheduling of activities becomes necessary. As in the second model above, the scheduler can make use of existing commitments to reduce uncertainty in the schedule, while the overall stability of the system may benefit when agents are conservative about releasing commitments. We intend to investigate this complex set of tradeoffs in future research.

Notice that these models are not mutually exclusive. Each model is enabled by a progressively more complex bidirectional interaction between local scheduling and coordination components. We can see these more complex intra-agent mechanisms as reasonable responses to complexity in the inter-agent environment. In taking this view, we observe an interesting tradeoff in which attempts to implement socially desirable outcomes via protocol design call for a more complex intra-agent dialog on the part of the individual agent.

As an example, we consider a distributed scheduling application in a resource-constrained environment [1] in which each agent receives a set of goals with resource requirements that should be accomplished according to certain deadlines and quality criteria. In this environment, agents are limited to resource negotiation (i.e. agents perform all tasks locally) The authors further assume that there are no external resources; all resources are controlled by other agents whose primary goal is to maximize utilization of the controlled resources. It is assumed that self-interested agents are interested in implementing their individual schedules at the lowest possible cost, while still meeting deadlines and quality criteria. It is also assumed that cooperative agents are interested in minimizing cost across all agents, while still meeting all deadlines and quality criteria.

We first consider the case in which agents all use the schedule-driven model. Since task arrival, schedule generation, and coordination happens asynchronously across agents, agents attempt to acquire resources when schedules are generated locally. Pathologies such as poaching described in [1] can easily arise under this model in highly resource-constrained environments. Agents who get their schedules sooner have a better chance of acquiring resources. This means that underloaded agents, whose schedules are significantly less complex to compute are at an advantage.

For the designer, there are several possible responses to this scenario. In a cooperative setting, agents may adopt mechanisms such as those proposed in [1] which tend to reduce the parallelism but do have the effect of synchronizing resource acquisition to achieve globally desirable allocations. In the absence of any incentive to synchronize, self-interested agents who are starved for resources may find it advantageous to adopt a more aggressive strategy for resource acquisition, namely one of acquiring resources in advance of scheduling for their actual use. These agents will monitor resource demand and attempt to acquire resources that are likely to be critical to their (still ungenerated) schedules when demand rises above a certain threshold. Once resources are acquired, these agents place constraints on the scheduler to induce a bias toward schedules that use the resources already acquired. Some resources will not be used due the scheduler's inability to create schedules that make use of them. However, if the cost of unused resources is less than the overall improvement in schedule utility, then this strategy has paid off for the self-interested agent.

All system stakeholders are now confronted with a situation that is to no one's satisfaction. The resource-controlling agents and system designers will observe that real resource utilization has decreased by the total amount of resources that are acquired by agents but not actually used. Agents who have successfully implemented their schedules may have done so at a higher cost than should be required because of the need to over-acquire and because of the increase in prices that comes with high demand for resources. One response is to introduce more sophisticated protocols that allow agents to decommit with some penalty less than the total cost of the reservation when they find that they are holding reserved resources that they will not actually use [18]. Agents are thus provided an incentive to release unneeded resources as early as possible to minimize this penalty, but the precise nature of this incentive remains an interesting research topic.

4 Meta-Level Conflict Resolution

As discussed in Section 1, when inter and intra agent negotiation cannot produce acceptable results, it may be necessary to move the negotiation process to a higher level where the overall objectives of the agent are examined and potentially modified. These "objectives" may take the form of the high-level tasks or goals (or organizational roles) that the agent is pursuing, or they may take the form of the objective function(s) that the agent uses to evaluate different possible (candidate) solutions. The notion of a changing objective function assumes a *quantified* model of agent activities in which there are generally multiple solutions and different solutions have different statistical characteristics, e.g., some solutions take more time but produce a higher quality result, some solutions cost more, some solutions require fewer resources, etc. In other words, agents *satisfice* and evaluate the "goodness" of particular solutions using goal criteria or an objective function. Thus, agents can redirect problem solving to compensate for poor solutions, to explore a new portion of the solution space, by changing their goals or tasks, or, changing their evaluation criteria or the function that defines which goals or tasks and which solution characteristics (e.g., completeness) are important.

This relates to the notion of a lattice of potential compromises in DENEGOT [16]. However, while DENEGOT also uses a satisficing model of computation, satisficing in DENEGOT entails relaxing hard constraints; the lattice of potential compromises expresses preferences for the relaxation of particular constraints or sets of particular constraints. In our current research, GPGP/DTC/TÆMS, this is analogous to potentially changing hard deadlines, hard cost constraints, or hard quality requirements on particular tasks or sets of tasks. Our current view of the meta-level of conflict resolution is more general – rather than just relaxing hard requirements, it may be desirable for agents to propose a new set of tasks or goals to pursue, or to propose a new set of goal criteria for use by GPGP/DTC when evaluating candidate solutions. This view would map back into DENEGOT as there being a two dimensional array of lattices in which each point is a lattice for a particular set of goal criteria and for a particular set of tasks or goals for the agent.

If we view the overall solution space as surface, the idea is that there may be different high-points or peaks within the solution space, any one of which is approximately acceptable *if* the solution can be scheduled and coordinated. If we are unable, via feasibility analysis at the lower levels, to schedule a solution for the set of selected tasks and goals, it may be desirable to "jump" to a different part of the solution space and try again to implement the solution via scheduling and coordination.

The process of selecting new tasks and goals, and possibly changing objective functions, implies communication between the involved agents, i.e., it appears to pertain mostly to meta-level inter-agent conflict resolution. However, it also has a meta-level intra-agent component as the selection of high-level objectives is interdependent with the process of performing the low-level scheduling and coordination. Just as there is a mutual, two-way, interaction between scheduling and coordination, and coordination between agents, there is also a mutual two-way interaction between the process of changing the high-level objectives and the detailed feasibility analysis. Earlier thoughts on this meta-level process [13] did not identify or address the potential for intra-agent interaction.

5 Conclusion and Future Directions

Interaction and conflict appear at all levels of agent control. In a very real sense, most aspects of agent control problem solving are interdependent, and most aspects of agent control and domain problem solving are also interdependent. This interdependence is often avoided through simplified agent control models or assumptions of independence. However, as we as a discipline push agent technology and apply it to wider problems (and build open systems), the issue if interdependence moves to the foreground. Unfortunately, this paper provides few answers, and attempts only to identify and describe some of the issues and concerns that we have encountered. In the future we will continue to explore the issue of interdependence, and conflict, within and between agents.

References

[1] Michael Chia, Daniel Neiman, and Victor Lesser. Coordinating asynchronous agent activities in a distributed scheduling system. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, 1998.

[2] Phillip R. Cohen, Adam Cheyer, Michelle Wang, and Soon Cheol Baeg. An open agent architecture. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents*, pages 197–204. Morgan Kaufmann, 1998.

[3] P.R. Cohen and H.J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(3):213–261, 1990.

[4] Keith Decker and Jinjiang Li. Coordinated hospital patient scheduling. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS98)*, pages 104–111, 1998.

[5] Robert Doorenbos, Oren Etzioni, and Daniel Weld. A scalable comparision-shopping agent for the world-wide-web. In *Proceedings of the First International Conference on Autonomous Agents*, pages 39–48, Marina del Rey, California, February 1997.

[6] P. Faratin, C. Sierra, and N. Jennings. Negotiation Decision Functions for Autonomous Agents. *International Journal of Robotics and Autonomous Systems*, 24(3-4):159–182, 1997.

[7] T. Finin, R. Fritzson, D. McKay, and R. McEntire. KQML as an agent communication language. In *Proceedings of the Third International Conference on Information and Knowledge Management CIKM'94*. ACM Press, November 1994.

[8] Alan Garvey, Keith Decker, and Victor Lesser. A negotiation-based interface between a real-time scheduler and a decision-maker. In *AAAI Workshop on Models of Conflict Management*, Seattle, 1994.

[9] Michael N. Huhns and Munindar P. Singh. Agents and multiagent systems: Themes, approaches, and challenges. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents*, pages 1–23. Morgan Kaufmann, 1998.

[10] Nicholas R. Jennings, Katia Sycara, and Michael Wooldridge. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems*, 1(1):8–38, 1998.

[11] Stanley M. Sutton Jr. and Leon J. Osterweil. The Design of a Next-Generation Process Language. UMASS Department of Computer Science Technical Report TR-96-30, 1996. Revised January 22, 1997.

[12] Henry Kautz, Bart Selman, Michael Coeh, Steven Ketchpel, and Chris Ramming. An experiment in the design of software agents. In Michael N. Huhns and Munindar P. Singh, editors, *Readings in Agents*, pages 125–130. Morgan Kaufmann, 1998.

[13] Brigitte Lâasri, Hassan Lâasri, Susan Lander, and Victor Lesser. A generic model for intelligent negotiating agents. *International Journal on Intelligent Cooperative Information Systems*, 1(2):291–317, 1992.

[14] 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.

[15] 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.

[16] T. Moehlman, V. Lesser, and B. Buteau. Decentralized Negotiation: An Approach to the Distributed Planning Problem. In K. Sycara, editor, *Group Decision and Negotiation*, volume 1, pages 161–192. Kluwer Academic Publishers, 1992.

[17] T. Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 256–262, 1993.

[18] T. Sandholm and V. Lesser. Advantages of a leveled commitment contracting protocol. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 126–133, 1996.

[19] Tumas Sandholm and Victor Lesser. Issues in automated negotiation and electronic commerece: Extending the contract net framework. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS95)*, 1995.

[20] Y. Shoham and M. Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 276–281, San Jose, July 1992.

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

[22] Milind Tambe. Agent Architectures for Flexible, Practical Teamwork. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, pages 22–28, July 1997.

[23] Thomas Wagner, Victor Lesser, Brett Benyo, Anita Raja, Ping Xuan, and Shelly XQ Zhang. GPGP2: Improvement Through Divide and Conquer. Working document, 1998.

[24] Thomas Wagner, Anita Raja, and Victor Lesser. Modeling Uncertainty and its Implications to Design-to-Criteria Scheduling. *Under review, also available as UMASS Department of Computer Science Technical Report TR-98-54*, 1999.

[25] M.P. Wellmen, E.H. Durfee, and W.P. Birmingham. The digital library as community of information agents. *IEEE Expert*, June 1996.

[26] Ping Xuan and Victor Lesser. Handling Uncertainty in Multi-Agent Commitments. *Under review, also available as UMASS Department of Computer Science Technical Report TR-99-05*, 1999.