An Autonomous Manufacturing Collective for Job Shop Scheduling

Steven Y. Goldsmith and Leslie D. Interrante

Sandia National Laboratories[1]
Albuquerque, NM 87185-5800
(505) 845-8926, (505) 844-0670
(sygolds@sandia.gov,ldinter@sandia.gov)

Abstract

This paper describes an ongoing research project in distributed, agent-based manufacturing scheduling systems which exhibit collective system behavior. A scheduling system is described which exhibits purely low-level, reactive behavior. Two types of agents comprise the collective: part agents and machine agents. The collective is a mixed-initiative system in which agents representing parts attempt to maximize part flow to the next machine and agents representing machines attempt to maximize their utilization. Agents of the collective contain only local knowledge: a machine agent schedules operations on its machine and a part agent commits available parts to machine operations. This architecture provides an opportunity to use reinforcement learning techniques to allow an agent of the collective to improve performance by learning patterns of local interactions over time. Furthermore, both agent types of the collective exhibit anytime behavior by continuously computing the next decision until it is required to commit a decision to a fellow agent. Such agent information will be of value in further research, the goal of which is to allow the dynamic collective adapts its architecture at runtime in response to unresolvable scheduling conflicts by creating temporary, higher-level agents opportunistically to handle particular situations. The architecture of the Autonomous Manufacturing Collective (AMC) is described herein.

1. Introduction

Smith (1996) describes a set of research issues and goals which motivate modern, artificial-intelligence-based manufacturing scheduling research. The goals reflect the requirements of practical scheduling domains, including the following:

1. problem-solving under complex constraints and objectives,
2. near-optimal, "anytime" performance
3. generality across problem classes,
4. solution robustness and responsiveness to change, and
5. integration with other decision-making processes.

The authors have taken the general approach of combining a distributed, agent-based architecture with concepts from complex emergent systems to create a manufacturing scheduling system which exhibits the qualities described above. A typical approach to ongoing research in manufacturing scheduling is to begin with an architecture and a simple manufacturing environment, and then gradually increasing the complexity of the environment in which the scheduling system must perform. This paper describes a multi-agent scheduling system which exhibits only reactive behavior. Such an approach allows the authors to be concerned with achieving system-level emergent behavior which arises from the local knowledge and interactions of the low-level agents contained within the collective. Although the agents reason in a limited manner, they are quite sophisticated in their use of local knowledge and in the manner of inter-agent communication within the collective. The scheduling collective is created from a more general-purpose distributed agent system with a number of built-in high-level features which provides a basic architecture and associated communication protocol for achieving the desired scheduling system goals (Goldsmith 1997).

The architecture of the Autonomous Manufacturing Collective (AMC) is described herein. Section 2 provides an overview of the AMC. Agents of the collective are described in Section 3. The scheduling process is described in Section 4. Related work is covered in Section 5, and Section 6 contains a discussion of the benefits of the AMC and further work.

[1] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

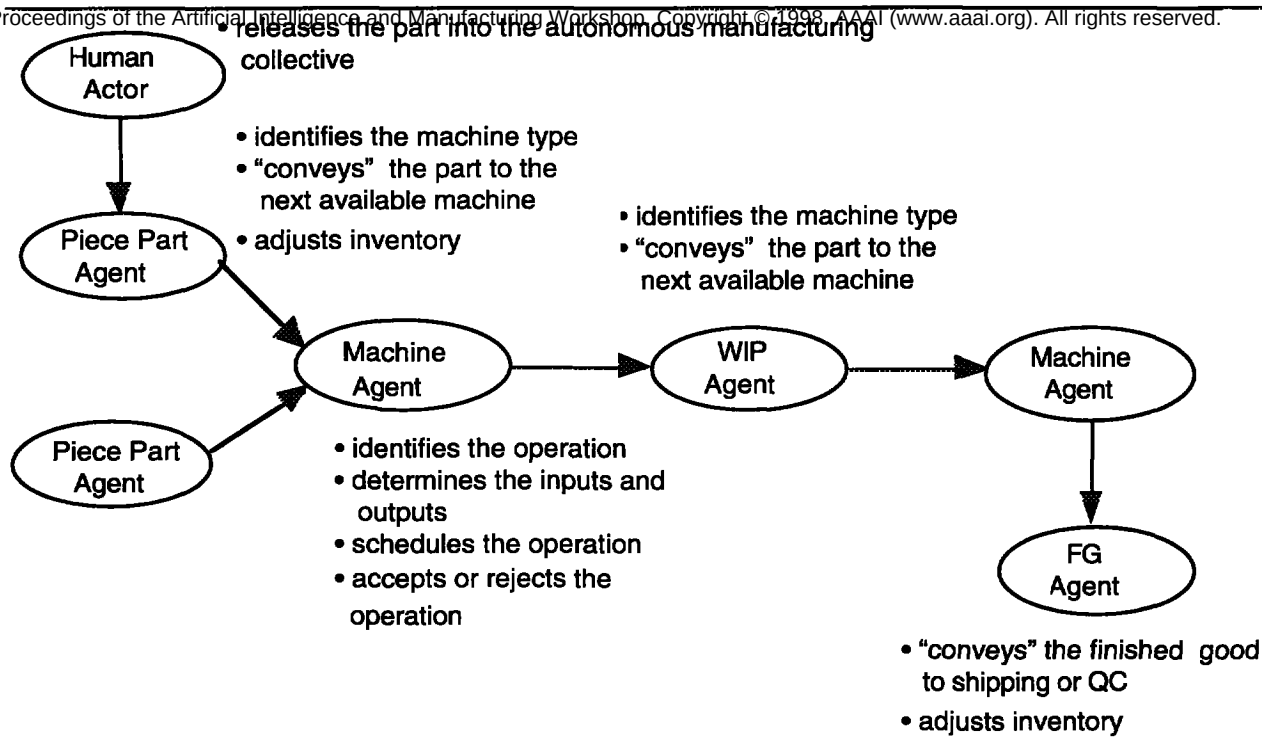

Figure 1. AMC agent interaction diagram.

2. Autonomous Manufacturing Collective

Figure 1 depicts a simplified view of the AMC. Two types of agents comprise the collective: part agents and machine agents. The collective is a mixed-initiative system in which part agents attempt to maximize part flow to the next machine and machine agents attempt to maximize their utilization. Agent reasoning is based on local machine or part knowledge. A part agent knows the machines and their respective agents that consume its parts and those machine agents that produce its part. A machine agent knows which part agents represent its input parts and which part agents represent its output parts. All constraints are handled in a local manner through the direct local interactions of these agents. Agents do not handle the propagation of constraints globally, and cannot take into account the pattern of the flow of parts or cascading constraints over multiple agents of the system.

Scheduling activity is triggered by the release of pieceparts into the manufacturing system, which is a characteristic of a "push" system. The architecture also supports pull and hybrid systems. (Most U.S. manufacturing systems exhibit a combination of push and pull characteristics.) Each piecepart agent advertises the availability of the part-type it represents to the appropriate machines. Machine agents determine which operations can be performed, given available machine time and parts. The particular machine to which available parts become committed and the particular operations performed on each machine are a product of the local decisions made by the agents after sharing part and resource information.

3. Agents and Manufacturing System Objects

AMC agents have the following attributes: (1) agents gain information about the state of the manufacturing system from manufacturing system objects; (2) agents interact and negotiate with one another to make commitments of parts to operations on machines, and; (3) agents impact the state of the manufacturing system by initiating events based on intra-agent commitments. Agent knowledge is restricted to its local environment, rendering overall manufacturing system behavior an emergent property of the collective. Agent behavior is motivated by goals which, when satisfied, attempt to maximize throughput and machine utilization.

Machine agents act on machine objects. Each machine agent represents a single machine in the system. The structure of a machine object is provided in Figure 2. A

machine object contains a unique ID, a schedule, a description of its current state, and several other slots. A machine object's agent can see its ID, place an operation in the machine's schedule, and query the machine for its current state. Since this is a purely reactive system, the agent does not attempt to look ahead in time in the machine's schedule to fit in a particular operation. The machine's schedule consists of a series of time intervals, each of which has an associated activity. Except for the

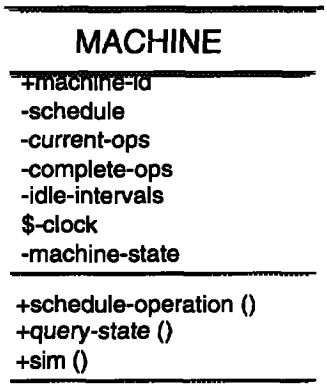

Figure 2. Machine class.

current activity, only "empty" intervals (with no activity scheduled) and scheduled maintenance exist in the schedule for this reactive system. Figure 3 depicts the state diagram for a machine in the system. A machine can be either "busy", "idle", or "down", with the associated events and state transitions as shown.

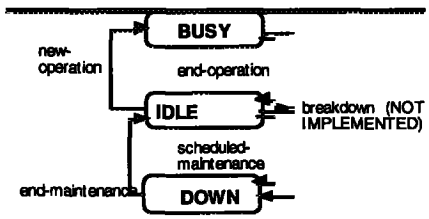

Figure 3. Machine state diagram.

Part agents act on part objects. A single part agent represents all instances of parts of a particular type in the manufacturing system. When a part undergoes a transformation as a result of a manufacturing process, the agent representing the input part type releases control of the part and hands it off to the agent representing the machine. The structure of a part object is depicted in Figure 4. Part objects can be one of three subclasses: "piecepart", "work in progress", and "finished good". "Work in progress" parts can be one of three subclasses: "assembly", with a many-to-one relationship; "disassembly", with a one-to-many relationship; and "treated-part", with a one-to-one relationship. Part agents can see the part ID, the

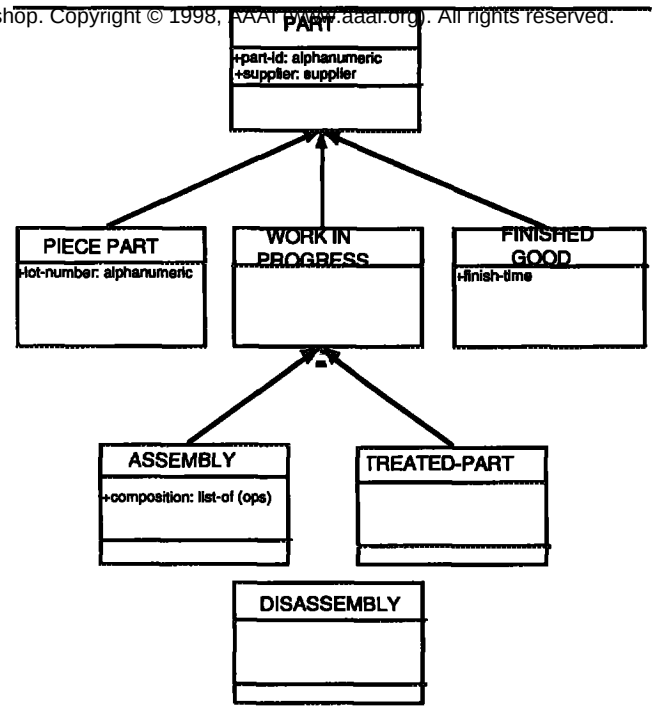

Figure 4. Part class hierarchy.

supplier (if applicable), and the lot number (if applicable) of all parts of the type it handles. Part object states include "available" and "committed".

4. The Scheduling Process

Each machine agent is responsible for scheduling operations on its machine. Each part agent is responsible for allocating parts to operations on machines in response to requests from machine agents. Thus, parts are manufactured according to a series of pairwise commitments between parts agents and machine agents along the routing for the finished product.

Figure 5 shows the high-level machine agent goals in the collective. The distributed agent architecture on which

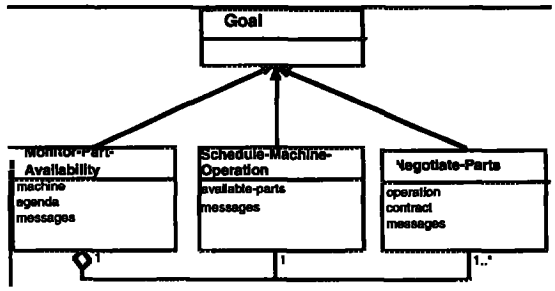

Figure 5: AMC machine agent goals.

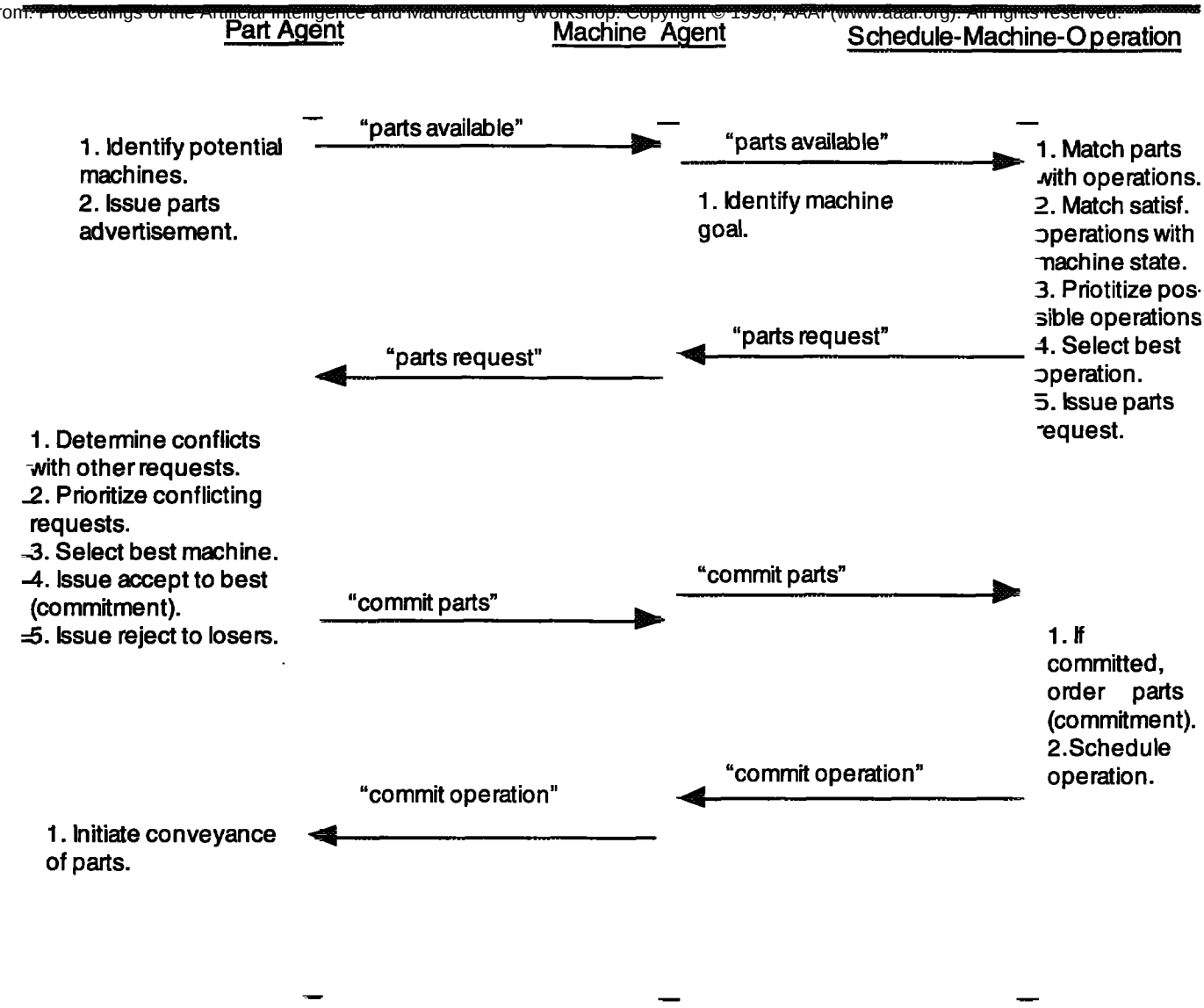

Figure 6. Part-machine agent protocol.

the AMC is built provides the following agent behavior.

Every machine goal inherits behavior and structure from the abstract class Goal. A "satisfy" method is called for each active goal, motivating the agent to repeatedly attempt to satisfy the goal until the goal is in the "satisfied" state. The machine agent contains two goals. The first goal, "schedule-machine-operation", causes the agent to repeatedly check the machine object's state and schedule. The agent continuously monitors the availability of parts for all operations which could be satisfied and checks the duration of the next idle time interval to determine which of the operations are feasible. The machine agent uses embedded knowledge to choose among the remaining operation candidates. In this simple system, the agent desires to maximize its utilization.

Thus, the agent will choose the operation or operations which fill the idle time interval in the machine object's schedule. The second machine agent goal, "monitor-part-availability", monitors messages from part agents and maintains a list of available parts compatible with the machine. The third goal, "negotiate-parts", is satisfied when a machine agent obtains a commitment from a part agent for parts. Part agents also continuously advertise part availability to machines in an "anytime algorithm" fashion (Dean & Boddy 1988; Zweben, Deal, and Gargan 1990).

Figure 6 shows the AMC agent communication protocol. The part agent advertises available parts. The machine agent, via the "schedule-machine-operation" goal, checks for operations with available parts, checks to insure that adequate time exists in the schedule to perform the

candidate operations, prioritizes the candidate set, and selects the best operation to next perform. Once the selection is made, the machine agent issues a part request. The part agent prioritizes conflicting machine requests for the same parts, selects the best machine, and accepts the request from the best machine. If the part agent accepts the machine agent's request for parts, the parts are ordered by the machine (a machine agent commitment) and the part agent then commits to supply the parts. The operation is placed in the machine object's schedule by the machine agent when the machine enters the idle state. This protocol is a variant of the contract net protocol (Van Parunak 1986; Baker 1991) with commitment deferred to a later phase of the negotiation.

5. Related Work

Most of the AI-based scheduling research performed to date involves the satisfaction of a system of constraints via search at a more global level such as ISIS (Fox 1994). OPIS (Smith 1994), the successor to ISIS, is able to switch from an order perspective to a resource (i.e., machine) perspective. In contrast, whereas the AMC exhibits machine-based scheduling only, it is a mixed-initiative system in which part agents attempt to maximize part flow to the next operation(s) and *machine agents attempt to maximize their utilization*. Micro-Boss (Sadeh 1994) selects and schedules only one activity at a time, but it focuses its search via features of the constraint graph.

O-Plan2 (Tate 1994) makes use of a blackboard architecture to manage planning, scheduling, and execution, whereas the AMC's low-level agents use agenda-type structures to manage their goals; machine agents store ordered, potential commitments in agendas which await the satisfaction of part availability and operation prioritization requirements.

Unlike the aforementioned scheduling systems and many others in the literature and in practice, the emergent behavior which is embodied by the AMC does not make use of any search techniques or a global data structure of any kind. Since agent decisions are purely local in nature, there is no guarantee of a particular kind of global scheduling behavior. Such behavior "emerges" as a function of the actions of the collective and the local data which drives individual agent behavior.

Since the system described in this paper is purely reactive, it cannot be compared to other well-known systems such as those described in (Zweben, Johnston & Minton, Sycara). Later versions of AMC will address these issues. Of this category of systems, DAS (Burke & Prosser 1991) has some architectural similarities to the AMC in that it is a distributed asynchronous scheduler. However, unlike the AMC, the DAS relies on agents at three levels of abstraction and incorporates a global constraint representation.

The allocation of agents to both parts and machines is an approach that has been around for over a decade (Duffie, Piper, Humphrey, and Hartwick 1986) and has been studied more recently in the context of autonomous agents (Lin and Solberg 1992). This work takes the same approach to agent allocation and uses a similar negotiation scheme but for different purposes.

6. Benefits and Further Work

This paper describes the AMC, a distributed agent system with emergent behavior for the accomplishment of manufacturing scheduling. The benefits of the AMC are relayed in terms of the list of scheduling research issues and goals described in the Introduction.

The system is very robust to a wide variety of sets of constraints and objectives. The robustness characteristic arises because of the simplicity of the pairwise interaction and the basic nature of the agents in that they communicate about resource availability in time. Agents use only local data and they attempt to meet the local goals of maximizing throughput and machine utilization.

Both agent types of the collective exhibit "anytime" behavior in the satisfaction of their respective goals by having the ability to dynamically trade time for more (local) system information in an attempt to better satisfy goals. An agent can make a commitment as soon as the appropriate communication protocol is satisfied, assuming there are resources which can be committed. Alternatively, the agent can delay its commitment in order to wait for a "better" solution. Strictly speaking, this approach does not fit the "anytime" criterion since there is no guarantee that the time delay will provide a better solution, but on the other hand, the agent is not required to delay making a commitment.

The AMC scheduling system exhibits generality across problem classes since the low-level, local nature of the agents of the collective is applicable to a broad range of application environments. The agent goals are very simple. Agent learning is fairly general in nature: reinforcement learning and simple "objective" functions cause the collective to tend toward maximizing local throughput and machine utilization. The emergent scheduling behavior is not the result of tuning or specializing to any particular domain.

Solution robustness and responsiveness to change is characteristic of the AMC since each agent can use reinforcement learning techniques to improve performance by learning patterns of local interactions over time. The same learning techniques allow each agent to adjust its behavior to changing manufacturing system conditions. This characteristic becomes more difficult to achieve as

scheduling moves from the simple, reactive system described herein to a more complex predictive-reactive model. Since the AMC is built on a more general distributed agent system architecture with associated communication protocols, it is relatively simple to integrate the scheduling agent collective with agents developed for other purposes both intra- and intercompany (e.g., material handling; supply chain reasoning). Agents with widely different goals can interact with the same part and machine objects.

The reactive system described in this paper is the first of a series of systems the authors plan to develop to investigate distributed agent systems for manufacturing scheduling, focusing on predictive-reactive scheduling and dynamic architectures which adapt their structure at runtime in response to scheduling conflicts by creating higher-level agents as needed.

7. References

Baker, A. (1991): Manufacturing Control with a Market-Driven Contract Net. Dissertation. Troy, NY: Rensselaer Polytechnic Institute, 1991.

Burke, P. and Prosser, P. (1991): A distributed asynchronous system for predictive and reactive scheduling. Artificial Intelligence in Engineering **6** (1991), 3.

Duffie, N., Piper, R., Humphrey, B. and Hartwick, J. (1986): Hierarchical and non-hierarchical manufacturing cell control with dynamic part-oriented scheduling. 14th NAMRC, North American Manufacturing Research Conference Proceedings, Minneapolis, MN, 1986.

Fox, M. (1994): ISIS: a retrospective. Intelligent Scheduling, M. Fox and M. Zweben, Eds. Morgan Kauffman, Palo Alto, 1994.

Goldsmith, S. (1997): The standard agent framework. AISL Technical Report, Sandia National Laboratories, Albuquerque, 1997.

Lin, G. Y.-J. and Solberg, J. (1992): Integrated shop floor control using autonomous agents. IIE Transactions **24** (1992), 3.

Smith, S. (1994): OPIS: a methodology and architecture for reactive scheduling. Intelligent Scheduling, M. Fox and M. Zweben, Eds. Morgan Kauffman, Palo Alto, 1994.

Smith, S. (1996): Artificial intelligence and manufacturing: a research planning report. Technical report, AI & Manufacturing Workshop, Albuquerque, 1996.

Sadeh, N. (1994): Micro-opportunistic scheduling: the micro-boss factory scheduler. Intelligent Scheduling, M. Fox and M. Zweben, Eds. Morgan Kauffman, Palo Alto, 1994.

Tate, A., Drabble, B. and Kirby, R. (1994): O-Plan2: an open architecture for command, planning, and control. Intelligent Scheduling, M. Fox and M. Zweben, Eds. Morgan Kauffman, Palo Alto, 1994.

Van Parunak, H. (1987). Manufacturing experience with the contract net. Distributed Artificial Intelligence: Research Notes in Artificial Intelligence, M. N. Huhns, Ed. Morgan Kauffman, Los Altos, 1987.

Zweben, M., Deal, M. and Gargan, R. (1990): Anytime rescheduling. Innovative Approaches to Planning, Scheduling, and Control: Proceedings of the 1990 DARPA Workshop, Morgan Kauffman, San Francisco, 1990.