A Blackboard Architecture for Integrating Process Planning and Production Scheduling

Norman M. Sadeh,† David W. Hildum,† Thomas J. Laliberty,‡
John McA'Nulty,‡ Dag Kjenstad† and Allen Tseng†

† Intelligent Coordination and Logistics Laboratory
Center for Integrated Manufacturing Decision Systems
The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
412.268.{8827,8818,2357,6676} · *fax:* 412.268.5569
{SADEH,HILDUM,DAG,YHTSENG}@CS.CMU.EDU

‡ Software Engineering Laboratory
Raytheon Systems Company
Tewksbury MA 01876-0901
508.858.{5756,1458} · *fax:* 508.858.5976
LALIBERTY_THOMAS@CAEMAC.MSD.RAY.COM
MCANULTY@CAESUN.MSD.RAY.COM

To appear in *Concurrent Engineering: Research and Applications*
Volume 6, Number 2, June 1998

The work described in this paper was sponsored by the Advanced Research Projects Agency under contract F33615-95-C-5523. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Advanced Research Projects Agency or the United States Government.

Abstract

As companies attempt to increase customization levels in their product offerings, move towards smaller lot production, and experiment with more flexible customer/supplier arrangements such as those made possible by electronic data interchange (EDI), they increasingly require the ability to (1) respond quickly, accurately, and competitively to customer requests for bids on new or modified products and (2) efficiently work out supplier/subcontractor arrangements for these products. This in turn requires the ability to (1) rapidly convert standard-based product specifications into process plans and (2) quickly integrate process plans for new orders into the existing production schedule to best accommodate the current state of the manufacturing enterprise.

This paper describes the IP3S system, an *Integrated Process Planning/Production Scheduling* shell for agile manufacturing. IP3S is based on a *blackboard architecture* that supports concurrent development and dynamic revision of integrated process planning/production scheduling solutions along with powerful workflow management functionalities for "what-if" development and maintenance of multiple problem assumptions and associated solutions. The IP3S blackboard architecture is designed to support coordinated development and revisions of solutions across the supply chain. The architecture is further shown to facilitate portability and integration with legacy systems.

IP3S has been validated in the context of a large and highly dynamic machine shop at Raytheon's Andover manufacturing facility. Empirical evaluation shows an average performance improvement of 23% in solution quality over a decoupled approach to building process planning/production scheduling solutions.

Relevant keywords:

PROCESS PLANNING, PRODUCTION SCHEDULING, AGILE MANUFACTURING, BLACKBOARD SYSTEMS, MIXED-INITIATIVE WORKFLOW MANAGEMENT, SUPPLY CHAIN MANAGEMENT, ELECTRONIC COMMERCE, KNOWLEDGE-BASED SYSTEMS

1 Introduction

As companies strive to increase customization levels in their product offerings, move towards smaller lot production, and experiment with more flexible customer/supplier arrangements such as those made possible by electronic data interchange (EDI) [Lee and Billington, 1992, Srinivasan *et al.*, 1994, Swaminathan *et al.*, 1995, Goldman *et al.*, 1995], they increasingly require the ability to (1) respond quickly, accurately, and competitively to customer requests for bids on new or modified products and (2) efficiently work out supplier/subcontractor arrangements for these bids. This in turn requires the ability to (1) rapidly convert standard-based product specifications into process plans and machine operations and (2) quickly integrate process plans for new orders into existing production schedules to best accommodate current loads and the availability of machines, labor, fixtures, raw materials, and components. To effectively support such capabilities requires bridging the gap between CAD/CAM (computer-aided design/computer-aided manufacturing) and production scheduling through the development of integrated process planning/production scheduling functionalities.

A number of factors contribute to the complexity of developing such integrated process planning/production scheduling solutions:

- the large number of interacting decisions (e.g., order promising, selection from among alternative processes and machines, sequencing and releasing decisions, procurement alternatives, overtime decisions)
- the ill-defined and often conflicting nature of objectives often based on different (incomplete) views of the problem (e.g., maximizing resource utilization, minimizing lead times, maximizing due date performance, minimizing costs)
- the inherent unpredictability of production systems (e.g., surges in demand, order cancelations, need for rework, delay in the delivery of supplies)

For this reason, effective decision support tools in this area require powerful

- *mixed-initiative decision support functionalities*, enabling the user to manually explore alternative problem assumptions and decisions while selectively relying on the system to complete and/or adapt solutions accordingly
- *workflow management functionalities* that assist the user in the management of a large number of interacting constraints and decision variables (e.g., assisting the user in keeping track of elements of a solution that are incomplete, inconsistent, or unsatisfactory and helping him/her improve the solution to address these problems)

This paper introduces a *blackboard*-based [Erman *et al.*, 1980, Nii, 1986, Corkill, 1991, Carver and Lesser, 1992] shell for integrated process planning and production scheduling, which supports:

1. *concurrent development and dynamic revision of integrated process planning and production scheduling solutions*, using new analysis and diagnosis tools that enable efficient process plan development through the early consideration of resource capacity and production constraints and greater optimization of production activities through direct visibility of process alternatives and tradeoffs

2. *mixed-initiative functionalities* that (a) support the maintenance of multiple problem instances and solutions, and (b) allow the user to control the development of solutions and explore alternative tradeoffs ("what-if" scenarios) by selectively addressing external events (e.g., new order arrivals, requests for bids, resource breakdowns) and imposing or retracting various assumptions (e.g., different delivery dates, work shifts, resource assignments, and requirements).

3. *workflow management functionalities* to alert and remind the user of new events or existing conditions that have not yet been addressed in a particular solution

4. the use of a *common representation* for exchanging process planning and production scheduling information

5. *supply chain coordination functionalities* enabling the coordinated development and dynamic revision of solutions across the supply chain (e.g., coordination with raw material and component suppliers)

6. *portability and ease of integration with legacy systems*, making it possible to quickly customize the system to support the integration of process planning, production scheduling and related activities (e.g., engineering, design, enterprise-level planning) across a broad range of environments

The IP3S shell has been evaluated in the context of a large and highly dynamic machine shop within the Raytheon Electronic Systems manufacturing facility in Andover MA. With roughly half of its incoming orders requiring the construction of new or modified process plans, over 150 CNC (computer numerical controlled) machine tools and over 100 people working three shifts, the Andover machine shop is a complex, highly dynamic, small-lot manufacturing environment that typifies many of the challenges involved in the development of effective solutions for integrating process planning and production scheduling. Empirical evaluation of the system has shown an average performance improvement of 23% in solution quality over a decoupled approach to building process planning/production scheduling solutions.

2 Integrating Process Planning and Production Scheduling

The technical challenges involved in effectively integrating process planning and production scheduling decisions in a complex and dynamic environment such as Raytheon's machine shop are many. From a pure process planning perspective, the number of orders that require the generation of new process plans and production of new fixtures for these plans, as well as the sheer variety of parts and machines present a significant challenge in their own right. As in other large machine shops, production scheduling in this environment is no easy task either. Major difficulties include (1) the presence of multiple sources of uncertainty, both internal (e.g., machine breakdowns) and external (e.g., new order arrivals, delays in the development of new fixtures or delivery of raw materials), (2) the difficulty in accurately accounting for the finite capacity of a large number of resources operating according to complex constraints, and (3) the need to take into account the multiple resource requirements of various operations (e.g., machines, labor, tools/fixtures, supplies, NC (numerical control) programs).

While considerable progress has been made with respect to software technologies for process planning and finite-capacity production scheduling, very little attention has been given to issues of integration. Except for a few attempts [Aanen, 1988, Iwata and Fukuda, 1989, Khoshnevis and Chen, 1989, Tönshoff *et al.*, 1989, Bossink, 1992, Zhang and Mallur, 1994, Huang *et al.*, 1995], often in the context of small manufacturing environments, process planning and production scheduling activities are typically handled independently, and are carried out in a rigid, sequential manner with very little communication. Process alternatives are traded off strictly from the standpoint of engineering considerations, and plans are developed without consideration of the current ability of the shop to implement them in a cost-effective manner. Likewise, production scheduling is performed under fixed process assumptions and without regard to the opportunities that process alternatives can provide for acceleration of production flows. Only under extreme and ad hoc circumstances (e.g., under pressure from shop floor expediters of late orders) are process planning alternatives revisited. This lack of coordination leads to unnecessarily long order lead times and increased production costs and inefficiencies, and severely restricts the ability to effectively coordinate local operations with those at supplier/customer sites, whether internal (e.g., a tool shop) or external (e.g., raw material suppliers).

Even with the support of sophisticated state-of-the-art computer-aided process planning and scheduling techniques, process planning and production scheduling remain highly interactive processes, where the user has to be able to evaluate alternative decisions based on experience and knowledge that is not easily amenable to computer modeling. Rather than committing to a prespecified decision flow, as in earlier approaches (see [Huang *et al.*, 1995] for a review of work in this area), the IP3S blackboard architecture emphasizes a more versatile integration framework where the user can dynamically select between alternative decision flows and control regimes. The resulting shell provides a customizable framework capable of supporting a wide range of integrated process planning and production scheduling decision flows, including all three of the approaches identified in [Huang *et al.*, 1995] as well as a number of more complex hybrids.

3 The IP3S Blackboard Architecture

The use of blackboard architectures as a vehicle for integrating multiple sources of knowledge to solve complex problems has been demonstrated in a variety of application domains (e.g., speech understanding [Erman *et al.*, 1980], signal interpretation [Nii *et al.*, 1982, Lesser and Corkill, 1983], planning [Hayes-Roth *et al.*, 1979, Currie and Tate, 1991], scheduling [Smith, 1994, Hildum, 1994] as well as some concurrent engineering applications [Prasad, 1997]). Blackboard architectures emphasize modular encapsulation of problem-solving knowledge within independent knowledge sources. These knowledge source modules work collectively to develop solutions to problems by communicating through a shared data structure, namely, the blackboard.

By explicitly separating domain knowledge (i.e., process planning knowledge, production scheduling knowledge, heuristic integration knowledge) and control knowledge, the IP3S blackboard architecture offers several key advantages:

- *flexibility of the control mechanism*, making it possible for the user to (1) dynamically select from among multiple control regimes (e.g., highly interactive control regimes where most decisions are made by the user versus more autonomous regimes where the user specifies high-level tasks or “goals” and lets the system decide how to accomplish them) and (2) support powerful workflow management functionalities

- *extensibility of the architecture*, making it particularly easy to add and enhance knowledge sources (e.g., new analysis and diagnosis knowledge sources)

- *ease of integration with legacy systems* through the encapsulation of existing problem-solving systems as knowledge sources

- *reusability of knowledge sources* across multiple domains (e.g., utilizing existing analysis and diagnosis knowledge sources in different scheduling applications)

Figure 1 provides an overview of the IP3S blackboard architecture. The system consists of a *blackboard*, a *controller*, a collection of *knowledge sources* (KSs)—including a process planning KS, a production scheduling KS, a communication KS and several analysis/diagnosis KSs (e.g., a KS to generate resource utilization statistics to help evaluate resource contention in different situations)—and a Motif-based GUI (graphical user interface). The IP3S blackboard, controller, KSs and GUI are implemented in C++. The blackboard (operating as a server), the controller and GUI (operating together as a single client), and the KSs (each operating separately and alternating between the roles of server and client) run as independent processes that communicate with each other using a CORBA (Common Object Request Broker Architecture)-based environment.

3.1 The IP3S Blackboard

The blackboard is the shared data structure on which KSs post solution components (e.g., new process plans and production schedules) and analysis results (e.g., resource utilization statistics). It is partitioned into an arbitrary number of user- or system-defined *contexts*, each possibly corresponding to different sets of working assumptions (e.g., the set of orders that need to be planned and scheduled, available resource capacities, supply delivery dates, etc.) and different solutions (i.e., integrated process planning/production scheduling solutions). Within each

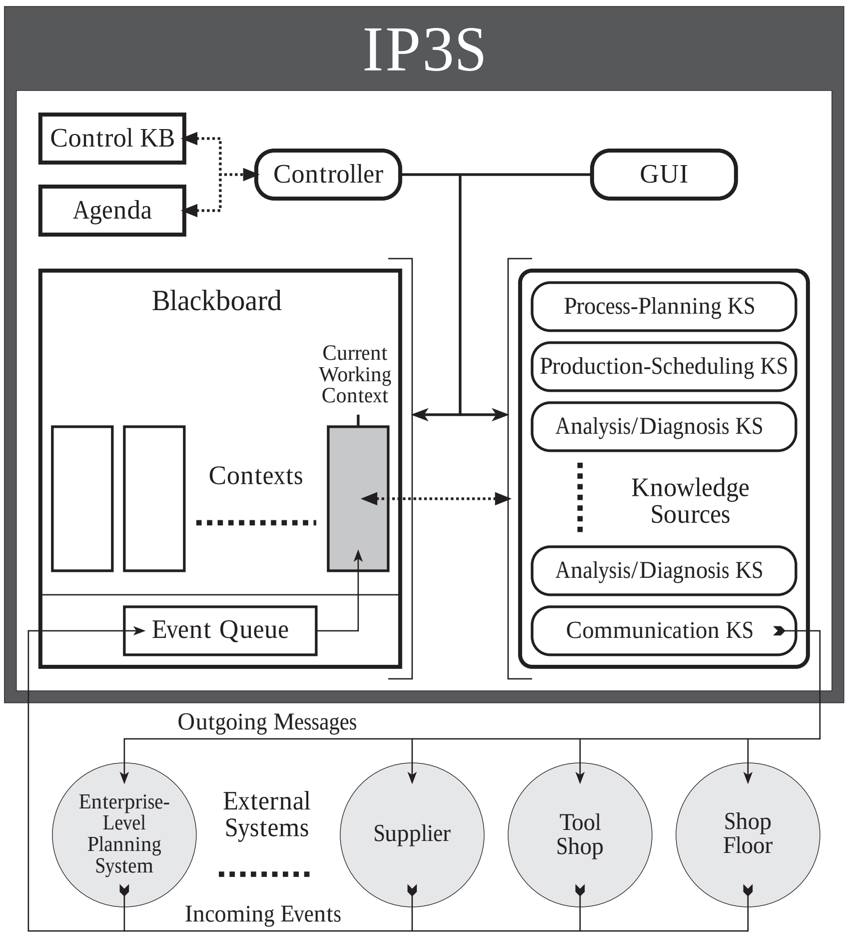

Figure 1: An Overview of the IP3S Blackboard Architecture

context, a summary of the current state of the solution is maintained in the form of a set of *unresolved issues*. An unresolved issue is an indication that a particular aspect of the current context solution is incomplete, inconsistent, or unsatisfactory. Problem solving in IP3S progresses through cycles during which one or more unresolved issue instances are selected to be resolved, a particular method of resolution is selected from among the set of methods applicable to the instance(s), and the method is executed by invoking the appropriate KS. Unresolved issues are created and deleted as a result of (1) KS invocations, (2) the incorporation of external events into a context, and (3) the modification of assumptions within a context while performing "what-if" analysis.

In the remainder of this section we describe the major architectural elements of the IP3S blackboard, with an emphasis on the mixed-initiative problem-solving and integration capabilities they support.

3.1.1 Contexts

The mixed-initiative decision-support capabilities of IP3S rely heavily on the use of contexts to support the representation of multiple problem instances. A context consists of a collection of resources (including machines, labor, and tools), raw material supplies, a collection of orders (and possibly requests for bids), and their corresponding process plans/production schedules. In addition, the set of unresolved issues represents inconsistencies within a partial solution that must be removed to produce a complete and satisfactory solution. As assumptions are modified and solutions are constructed within a context, the set of unresolved issues is updated to help the system and the user keep track of aspects of the current solution (within that context) that require further problem-solving attention.

The mixed-initiative power of the context mechanism comes from the capability it provides for the user to define a problem progressively and alternately. This can be done through either the selective incorporation of events into a context (e.g., requests for bid, shop floor events) or the modification of problem assumptions within a context (e.g., by changing various order and resource attributes like due dates and work shifts).

Contexts may be created either by the user or automatically by the system. It is through the creation of multiple contexts that "what-if" analysis is supported by IP3S. By creating multiple copies of a context, changing various assumptions within the copies and producing solutions for each, alternate solution paths can be explored. The user or the system can leave a particular context at any point in time and explore other potentially more promising alternatives in other contexts. Changes to order and resource attributes within one context remain local to that context and do not affect other contexts that may include the same entities. When a KS is invoked, its results are visible only to the context in which the user is currently working (called the "current working context").

3.1.2 Events

Events received from outside information sources (e.g., an enterprise-level planning system, raw material suppliers, a tool shop, the shop floor) are posted on the blackboard *event queue* in preparation for being *incorporated* within (or "pulled into") one or several contexts by the user or the system. These events include the notification of incoming orders, requests for bid, promised dates from suppliers, resource breakdowns and various shop floor updates. When an event is incorporated into a context, the blackboard translates the initial result (or implication) of the action described by the event into an appropriate unresolved issue. The objective for the user or the system is to resolve each such issue, through the activation and execution of one or more KSs, until all events have been incorporated into a context and no more unresolved issues remain.

The event processing mechanism in IP3S supports two important mixed-initiative capabilities:

1. It allows both the user and the system to ignore events that are unlikely to affect the part of the solution upon which work is currently being done. For example, when revising a plan for a part that needs to be processed within the week, incoming-order events for new orders due three months downstream can often be ignored.

2. It allows both the user and the system to process conditional events, such as requests for bid. For example, upon receipt of a request for bid on a possible order, a copy of the current context can be created, within which the order can be planned and scheduled. The resulting solution showing the impact of the possible order can then be evaluated to determine a realistic completion date and decide whether or not to submit a bid.

3.1.3 Unresolved Issues

As the assumptions within a particular context are modified or as new events are incorporated into a context, the set of unresolved issues within the context is updated automatically by the IP3S blackboard. The set of unresolved issues within a context defines areas in the current partial solution where further problem-solving effort remains to be done to produce a complete, consistent, and satisfactory solution. It provides a powerful *workflow management mechanism* that helps IP3S users keep track of the work that remains to be done in a given context, without imposing any particular decision flow (e.g., process plans or production schedules can be modified at any point in time).

The IP3S architecture distinguishes between three main types of unresolved issues, relating to (1) the *completeness* of the solution, such as an order lacking a process plan or production schedule, (2) *inconsistencies* within the solution, such as an order whose new process plan differs from the one currently assumed in the production schedule, and (3) potential areas for solution *improvement*, such as an order with an excessively late completion date or long lead time. Table 1 provides a sampling of IP3S unresolved issues implemented for Raytheon's machine shop. Because unresolved issues are defined in a declarative fashion, they can easily be modified as required when porting the system to new environments.

Parameterized unresolved issues allow the user to modify the problem-solving objectives by setting a threshold to control when a particular unresolved issue is created. Such is the case with the Tardiness issues, where the user may desire not to be alerted to tardiness in the schedule unless it reaches a critical point.

Table 1: A Sampling of IP3S Unresolved Issues

Order related:	Context related:
Completeness:	*Completeness:*
Order-w/o-Default-Process-Plan	Query-Awaiting-Response
Order-w/o-Approved-Process-Plan	*Inconsistency:*
Order-w/o-Production-Schedule	Outdated-Resource-Utilization-Statistics
Tool-Promise-Date-Required	Unprocessed-Shop-Floor-Update
Improvement:	*Improvement:*
Tardiness	Tardiness

The roles of the IP3S blackboard, contexts, events, and unresolved issues in supporting mixed-initiative problem solving are illustrated in detail in two example scenarios presented at the end of this paper.

3.2 The IP3S Controller

The IP3S Controller is responsible for directing solution construction, revision, and analysis, either through close interaction with the user, or on its own with the help of a knowledge base of control heuristics. The primary control-related mixed-initiative capabilities of IP3S manifest themselves in two key Controller functionalities:

1. *support for multiple control regimes*, ranging from a highly interactive mode where the user specifies each problem-solving action, to an autonomous mode where the Controller takes responsibility for (1) the selection of which events to incorporate into the current context, (2) the determination of which unresolved issues to resolve, and (3) the selection of the specific methods for their resolution

2. *support for multi-level customizable problem-solving tasks* to provide a range of low- to high-level modes of user interaction (e.g., the activation of a specific low-level KS service, the posting of high-level objectives or "goals", the activation of a sequence of services and goals)

IP3S allows the user to select from among different levels of interaction and different control regimes at any time. In addition, the set of high-level problem-solving tasks provided to the user can easily be augmented to accommodate changing user-interaction patterns. Specifically, a hierarchy of high-level goals and scripts can be defined in terms of the basic set of services provided by the particular problem-solving systems encapsulated as KSs and incorporated within IP3S.

To support these mixed-initiative capabilities, the IP3S Controller works off of an *execution profile* that records the assignment of responsibility for various problem-solving tasks (e.g., the incorporation of events into a context, the selection of unresolved issues to resolve and the methods for their resolution) to either the system (i.e., the Controller) or the user. The assignment of tasks can be interactively changed at any point by modifying the execution profile. To provide multiple levels of interaction with the system through the definition and activation of aggregate and goal-oriented problem-solving tasks, the IP3S Controller maintains its own declarative *control knowledge base* that links each unresolved issue to the set of problem-solving services applicable for its resolution. The control knowledge base also contains the collection of generic and domain-specific control heuristics that are used by the Controller to perform the tasks assigned to it, as recorded in the execution profile (e.g., different heuristics to improve the completion date of a particular order).

The IP3S Controller uses an *agenda* mechanism to keep track of the problem-solving tasks remaining to be executed. When a particular course of action is selected, either manually by the user or automatically through consultation with the appropriate control heuristics, one or several problem-solving task items are placed on the agenda, describing an action or sequence of actions to be performed by the system. The IP3S control architecture supports three types of agenda items:

1. *service* activations, which correspond directly to specific problem-solving services provided by the IP3S KSs (e.g., the Plan-Order service to construct a process plan for a new order (supported by the Process Planning KS), the Schedule-Order(s) service to incorporate an order with a process plan into the existing production schedule (supported by the Production Scheduling KS), the Send-Query service for requesting information from external systems such as the tool shop or raw material suppliers (supported by the Communication KS))

2. *goal* activations, which are used to specify high-level, objective-oriented problem-solving tasks that can be satisfied by the execution of either (1) a service, or (more likely) (2) a *sequence* (or "script") of services and subgoals (e.g., the Improve-Completion-Date goal to improve the completion date of a particular order; such a goal can be satisfied in a number of ways and will typically involve the application of one or more heuristics depending on the particular situation)

3. *scripts*, which specify a predefined sequence of KS services and goals generally known to accomplish a particular problem-solving task (e.g., a script that successively invokes Plan-Order to generate a process plan for a new order and then invokes the Schedule-Order(s) service to incorporate the new order into the existing production schedule

3.3 The IP3S Problem-Solving Cycle

All problem-solving activity in IP3S is triggered by either the incorporation of a new event (such as an incoming order or a shop floor status update) into the current working context, or the modification of an assumption within the current working context (e.g., "what-if" analysis to evaluate the benefits of adding work shifts or purchasing new machines), both of which can be performed by either the user or the Controller (as specified by the execution profile). The flow of problem solving in IP3S is summarized in Figure 2. It proceeds from the modification of the current working context in a clockwise direction through the following steps:

1. updating the set of unresolved issues within the current working context to reflect the initial problem-solving action

2. selecting an unresolved issue to resolve

3. selecting a resolution method for the selected unresolved issue

4. activating the selected resolution method

5. executing the problem-solving service that corresponds to the activated resolution method

The IP3S Controller is invoked whenever there are problem-solving tasks on the agenda remaining to be executed, or, when running automatically (and depending on the execution profile), there are events to incorporate or unresolved issues to resolve. During each problem-solving cycle, the Controller performs one of the following actions (in sequence):

1. execute the top-most item on the agenda

2. *if responsible for event processing*, incorporate an unincorporated event into the current working context

3. *if responsible for unresolved issue resolution*, select an unresolved issue to resolve and then select and activate a method for its resolution

Figure 3 provides a more detailed view of the actions taken by the Controller in the processes of selecting, activating, and executing agenda items. These actions are summarized below:

- the execution of a KS *service* leads directly to a modification of the current working context

- the execution of a *goal* warrants the selection and activation of a method for its satisfaction

- the execution of a *script* involves a sequence of service or goal executions

3.4 IP3S Knowledge Sources

Knowledge sources serve as the primary problem solvers in a blackboard system. They communicate their results by posting new information to the blackboard (e.g., new process plans, new production schedules, new analysis results) and modifying existing information (e.g., updated process plans and reoptimized production schedules).

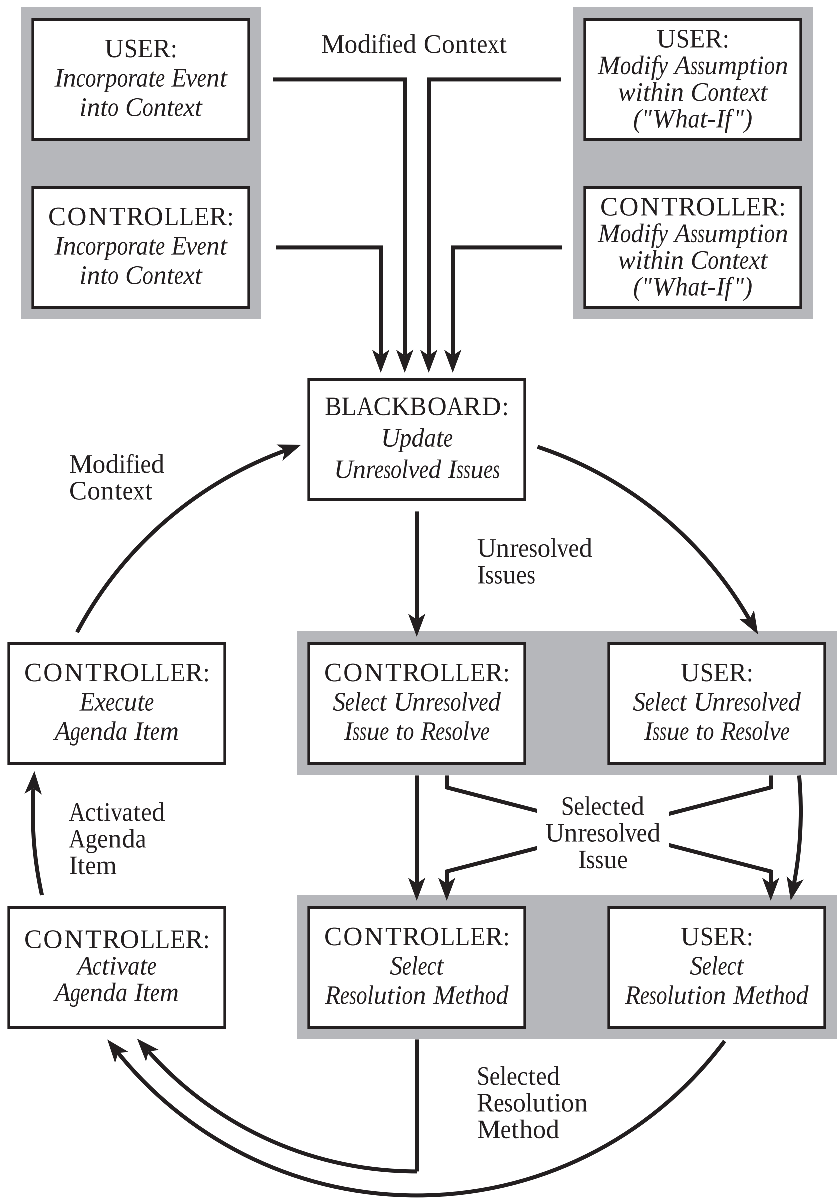

Figure 2: A Summary of the Problem-Solving Flow in IP3S

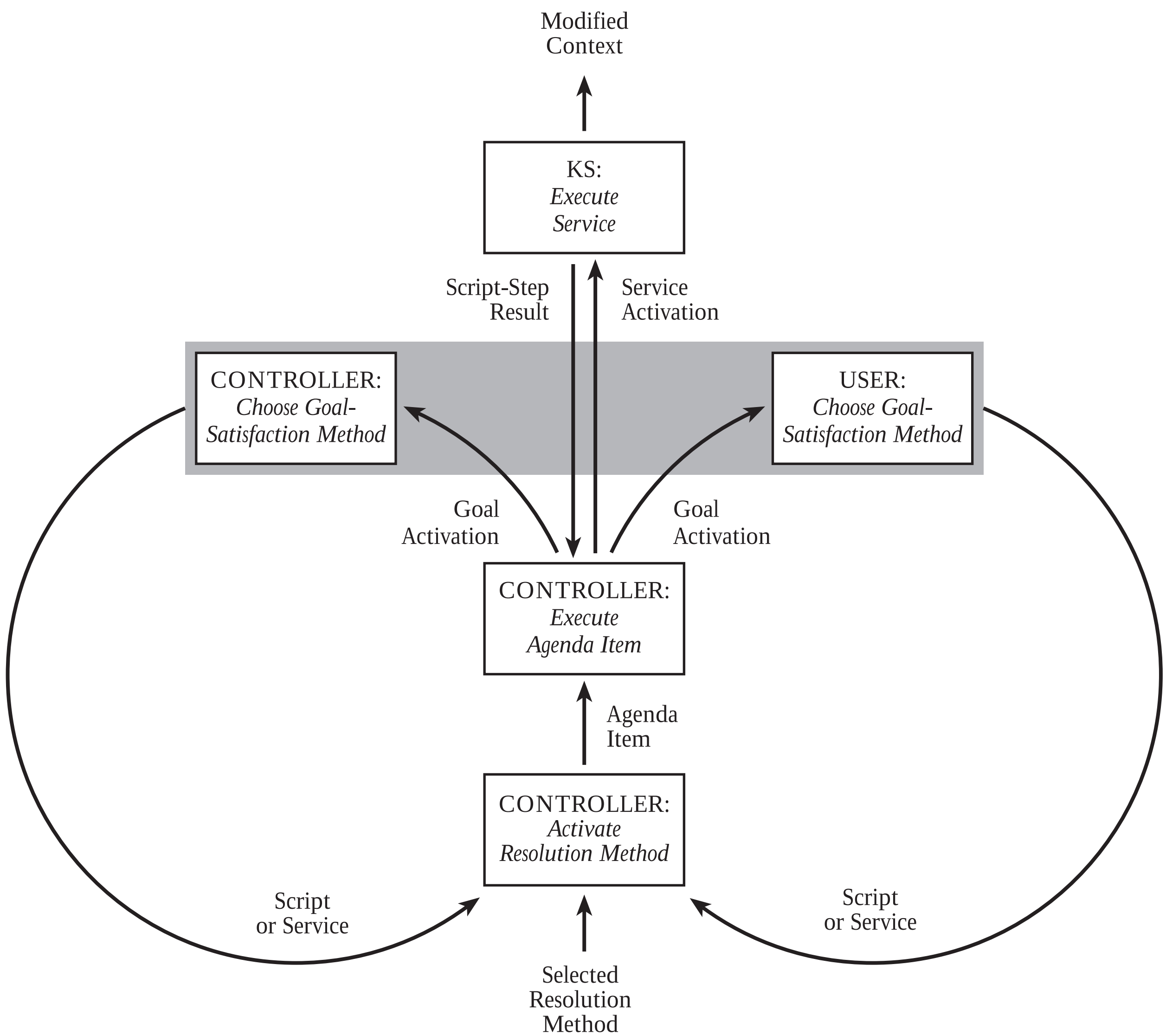

Figure 3: A Summary of the Selection, Activation, and Execution of Agenda Items in IP3S

In IP3S, each domain-level KS acts primarily as a server that supports a variety of problem-solving services. A KS service may require a set of parameters which are defined by the unresolved issue(s) for which the service is applicable.

The IP3S shell customized for Raytheon relies on three KSs responsible for performing various process planning, production scheduling, and analysis services. In addition, IP3S is equipped with an internal Communication KS for managing interaction with various external systems. These KSs are described below:

- The Process Planning KS is implemented by Raytheon's IPPI generative process planner [Raytheon Company, 1993a, Raytheon Company, 1993b], which constructs process plans consisting of machining operation sequences (including recommended tooling requirements and a bill of materials), given STEP (International Standard for the Exchange of Product Data)-based part specifications

  A key feature of IPPI is its ability to develop and revise process plans while considering existing and projected resource demand—information that is summarized in the resource utilization statistics that are posted on the IP3S blackboard by the Resource Utilization Analysis KS (described below). While some IPPI services are completely automated (e.g., generation of default process plans for an order), others require interaction with an industrial engineer.

- The Production Scheduling KS is implemented by the MICRO-BOSS system [Sadeh *et al.*, 1993, Sadeh, 1994, Sadeh, 1995], a dynamic finite-capacity scheduling tool developed at Carnegie Mellon University that has been shown to support efficient just-in-time operation in complex and dynamic manufacturing environments (e.g., MICRO-BOSS has been customized for environments as diverse as a printed wiring assembly area, a machine shop, a blending and packaging facility).

- The Resource Utilization Analysis KS estimates resource contention by accounting for both current reservations within the existing schedule and projected demand from unscheduled orders. Its results are posted on the blackboard (in the current context) for use by the Process Planning KS to identify promising process alternatives.

- The Communication KS facilitates communication between IP3S and various external systems (e.g., an enterprise-level planning system, raw material suppliers, a tool shop, the shop floor). Its responsibility is to formulate the outgoing messages transmitted to the outside environment.

4 Two Problem-Solving Examples

To better illustrate the mixed-initiative and workflow management functionalities of IP3S, we provide the following two problem-solving scenarios. The first demonstrates a highly interactive session where a process plan for a new order is generated and is then revised so as to avoid congested work areas. The resulting process plan is eventually incorporated into the existing production schedule. Figure 4 illustrates the progression of blackboard states and user actions described in the scenario. The second scenario demonstrates a less interactive session where the user attempts to improve an existing production schedule. For brevity, some of the lower-level details in these scenarios have been left out.

4.1 Scenario One

In this scenario, the latest update from the enterprise-level planning system shows that a new order has arrived, say Order-35. This new order is recorded as an incoming event in the IP3S event queue (state A). While this order is only due in three weeks, a number of work areas are already heavily congested. The production manager decides that (s)he should quickly determine the impact of Order-35 on the current production schedule. Accordingly, (s)he incorporates the corresponding Incoming-Order event into the current working context (step 1), a context that contains a solution for all orders expected to impact production over the next few weeks. The event includes all of the necessary information about the order (e.g., its STEP-based part specification, quantity, due date).

Upon incorporating the new event into the current working context, the IP3S blackboard automatically generates an Order-w/o-Default-Process-Plan unresolved issue for Order-35 indicating that the new order lacks a process plan (state B). The IP3S GUI has a window displaying all of the unresolved issues in the current working context to help the user understand, at any point in time, what issues have already been accounted for and what issues are still pending. The GUI also supports different ways of filtering unresolved issues, allowing the user to focus on only those issues of interest at any given time.

In this particular case, we assume that the Order-w/o-Default-Process-Plan issue for Order-35 is the only pending unresolved issue within the current working context. Naturally, the production manager selects this issue as the next one to resolve. (S)he is then presented with a menu listing all methods available to resolve this particular issue. Typically, such a menu will include a combination of goal, script and KS service activations. In our example, the menu includes (1) a KS service activation to request a process plan for the order, (2) a script that successively generates a process plan and then reoptimizes the current schedule, and (3) a goal activation to include the new order in the existing solution, while maintaining all prior delivery commitments and meeting the new order's due date (note that this goal is not necessarily achievable and may require exploring a number of alternative solution paths). To keep things simple, let us assume that our user decides to first evaluate the impact of the new order, with its default (preferred) process plan, on resource congestion. Accordingly, (s)he activates the Plan-Order service of the Process Planning KS (step 2). This service generates a default process plan for the new order and posts it to the blackboard.

Following the assignment of the new default process plan to the order, the blackboard automatically replaces the Order-w/o-Default-Process-Plan issue with a new Order-w/o-Approved-Process-Plan issue—approval of a process plan requires the intervention of an industrial engineer through the IPPI Process Planning KS—and generates a new Outdated-Resource-Utilization-Statistics issue to signal the need to recompute resource contention and account for the

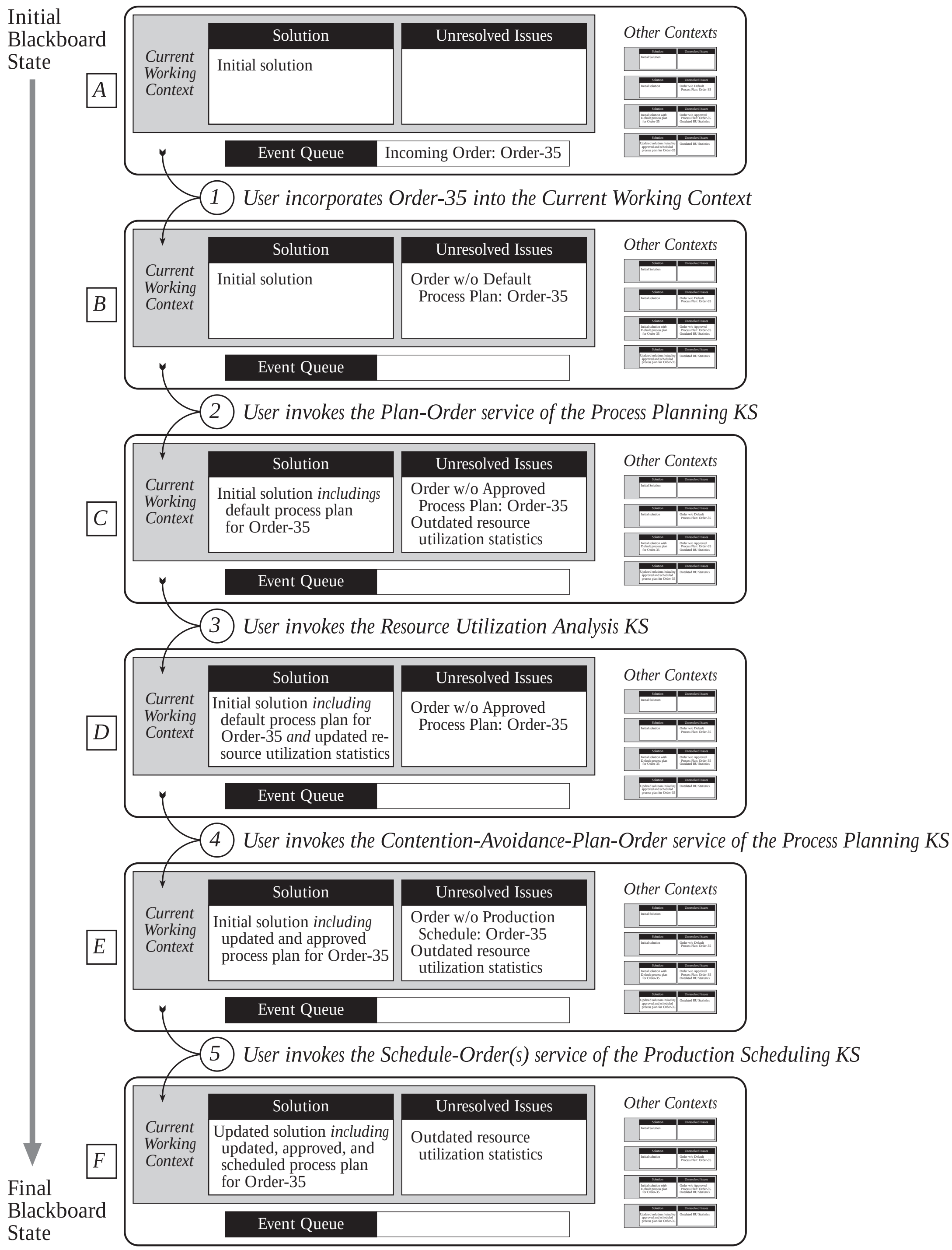

Figure 4: The progression of blackboard states and user actions described in Scenario One

new default process plan (state C).

Our user, who, in this session, is opting for a simple mode of interaction where (s)he successively selects unresolved issues and KS service activations, now decides to update the resource utilization statistics and evaluate the extra resource contention resulting from the process plan just generated for Order-35. To do so, (s)he selects the Outdated-Resource-Utilization-Statistics issue and then invokes the Resource Utilization Analysis KS to perform the Update-Resource-Utilization-Statistics service (step 3). Following the updating of the resource utilization statistics, the blackboard automatically deletes the Outdated-Resource-Utilization-Statistics issue (state D).

Upon looking at the new resource utilization statistics and noticing that the default process plan requires several resources for which contention is already high, our user now invokes the Process Planning KS to activate the Contention-Avoidance-Plan-Order service in an attempt to avoid using those specific resources (step 4). With the help of an industrial engineer, this service generates an approved process plan for the new order that avoids resources already in high contention and posts it on the IP3S blackboard within the working context. The blackboard replaces the Order-w/o-Approved-Process-Plan issue with a new Order-w/o-Production-Schedule issue indicating that the order now lacks a production schedule (state E).

The user can now invoke the Production Scheduling KS to perform the Schedule-Order(s) service (step 5). This service incorporates the order, according to its new approved process plan, into the existing production schedule within the current working context. Upon completion, the blackboard automatically deletes the Order-w/o-Production-Schedule unresolved issue (state F). With the exception of the remaining Outdated-Resource-Utilization-Statistics issue, all unresolved issues relating to the new order have now been resolved, and the current solution is complete (for the moment).

Note that, in practice, a scenario like the one above might also require some interaction with the tool shop (using the Communication KS) in the case where new tools are required by the approved process plan.

4.2 Scenario Two

Following the incorporation of several new orders into the current working context, the user is alerted that one particular order, say Order-12, is now scheduled to complete four weeks past its due date. This is indicated by the presence of an order-related Tardiness unresolved issue within the current working context. Our user decides to create a copy of the current working context, making it the new working context, and attempts to find a solution where Order-12 is completed earlier. Within this new working context, (s)he selects the Tardiness unresolved issue associated with Order-12 and is presented with a list of possible courses of action, some involving the activation of simple KS services and others involving script or goal activations. Examples of possible options include:

- activating the schedule reoptimization service of the MICRO-BOSS Production Scheduling KS, while increasing the tardiness penalty ("weight") associated with Order-12 (i.e., getting MICRO-BOSS to work harder at meeting Order-12's due date, even if this means sacrificing other orders)

- activating a script that will attempt to (1) identify possible bottleneck resources where production of Order-12 gets delayed and (2) reroute Order-12 by generating a process plan that avoids these congested work areas

- activating a script that will attempt to (1) identify possible bottleneck resources where production of Order-12 gets delayed and (2) reroute one or more orders competing with Order-12 to expedite Order-12 through the shop

- activating a goal (e.g., Improve-Completion-Date) with the objective of building a satisfactory solution where Order-12 completes earlier (i.e., without specifying the exact approach to be taken)

The number of steps in Order-12's process plan is quite large, and while the IP3S Gantt chart suggests there are a number of congested areas, none clearly dominates the others. Our user cannot easily identify a specific KS service activation or script activation likely to work in this particular situation. Instead, (s)he activates an Improve-Completion-Date goal, even though this means waiting a little longer for the system to try a number of alternatives. The corresponding goal activation gets placed on the agenda and the Controller begins processing it during the next control cycle. The IP3S control heuristics suggest, as a first attempt at satisfying this goal, to activate the schedule reoptimization service of the MICRO-BOSS Production Scheduling KS, while increasing the tardiness penalty of Order-12 fivefold. The corresponding activation is placed at the top of the agenda. Simultaneously, a copy of the current context is created to record the result of its execution.

Following the execution of the schedule reoptimization service (and the removal of the corresponding activation from the agenda), a newly generated schedule is written to the copied context. In this new schedule, Order-12 completes only two weeks past its due date, but another order, say Order-14, is now late by a week. Next, the IP3S control heuristics suggest trying a script that involves generating a new process plan for Order-12 that will attempt to avoid resource contention and, in the process, complete Order-12 earlier. An additional copy of the current working context is created and the script is activated. The result of each of its steps is stored in the new copied context. This time the resulting solution is one where Order-12 completes by its due date and all other orders still meet theirs. When this solution is written to the new context, the blackboard automatically deletes the Tardiness unresolved issue associated with Order-12 (in this context). The goal on the agenda has now been satisfied and is removed, and the new context becomes the current working context.

If, after trying all possible alternatives, the goal still cannot be satisfied, our user would still have the option to modify some assumptions within the context. For instance, (s)he could add an extra work shift to one or more bottleneck areas and see whether that helps satisfy the goal.

5 Empirical Evaluation

IP3S has been validated in the context of Raytheon's Andover manufacturing facility. Below, we present results of four sets of experiments, each representative of different shop load conditions. For each experiment, solutions were generated using two approaches: a traditional *decoupled* approach where process plans were built independently of load considerations, and an integrated approach where process plans were optimized by taking into account the presence of bottlenecks (as indicated by the statistics produced by the Resource Utilization Analysis KS). A threshold parameter was used to determine bottleneck conditions, with values of 30%, 50%, 70%, and 90% being tested in each experiment. Table 2 summarizes the results of these experiments.

Table 2: A Summary of IP3S Experimental Results

Experiment	Average *Decoupled* Cost	Average *Integrated* Cost	Average Improvement	Average Best Improvement	Average No. of Orders	Average Total Part Qty.
S1	840900	836310	1%	10%	81	7817
S2	3728124	3251212	10%	22%	174	5611
S3	6877796	3730082	37%	52%	66	3798
S4	1328717	945226	44%	69%	76	4727
Avg.	3193884	2190707	23%	38%	99	5489

To facilitate evaluation, process planning options were restricted to equally satisfactory choices, and solution quality was measured strictly in terms of due date, inventory, and lead time performance. Specifically, a cost was associated with each solution, computed as the weighted sum of the tardiness and inventory costs of all shop orders, with each order weighted by its part quantity. Tardiness penalties were adjusted to be substantially larger than inventory costs to reflect the importance of due date performance in this environment (as is the case in most just-in-time environments).

The results show an average 23% improvement in schedule cost obtained using the integrated planning and scheduling approach facilitated by IP3S, with an average improvement of 38% when taking the best bottleneck-threshold value. Closer analysis indicates that this mainly reflects significant improvements in due date performance, the most important objective in these experiments.

6 Summary and Concluding Remarks

The IP3S shell is designed around an innovative *blackboard architecture* that supports flexible, mixed-initiative decision making and user-oriented management of integrated problem-solving tasks within large-scale dynamic environments. Its architecture also makes it particularly portable and easy to integrate with legacy systems.

An operational prototype of IP3S has been implemented and evaluated within the context of Raytheon's

Andover manufacturing facility. The mixed-initiative and workflow management functionalities of IP3S enable users to quickly evaluate alternate modeling assumptions and solutions. Empirical results further indicate that the integrated approach to process planning and production scheduling facilitated by IP3S can yield significant performance improvements over traditional decoupled approaches across a range of load conditions.

Our current work focuses on generalizing the IP3S blackboard architecture to support a broader range of mixed-initiative and workflow management functionalities for supply chain coordination. This includes the ability to coordinate real-time available-to-promise responses across multiple sites (i.e., "lateral coordination"), and more generally, to dynamically maintain synchronized solutions across the supply chain. It also includes the development of workflow management and mixed-initiative functionalities to dynamically maintain consistent supply chain solutions across different levels of abstraction (i.e., "vertical coordination"), such as multi-facility master production scheduling (MPS) solutions and single facility solutions.

References

[Aanen, 1988] E. Aanen. *Planning and Scheduling in a Flexible Manufacturing System*. PhD thesis, University of Twente, Enschede Netherlands, 1988.

[Bossink, 1992] G.J. Bossink. *Planning and Scheduling for Flexible Discrete Parts Manufacturing*. PhD thesis, University of Twente, Enschede Netherlands, January 1992.

[Carver and Lesser, 1992] N. Carver and V.R. Lesser. The evolution of blackboard control architectures. CMPSCI Technical Report 92–71, Department of Computer Science, University of Massachusetts, Amherst, October 1992.

[Corkill, 1991] D.D. Corkill. Blackboard systems. *AI Expert*, 6(9):40–47, September 1991.

[Currie and Tate, 1991] K.W. Currie and A. Tate. O-Plan: The open planning architecture. *Artificial Intelligence*, 52(1):49–86, 1991.

[Erman *et al.*, 1980] L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy. The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–253, June 1980.

[Goldman *et al.*, 1995] S.L. Goldman, R.N. Nagel, and K. Preiss. *Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer*. Van Nostrand Reinhold, New York, 1995.

[Hayes-Roth *et al.*, 1979] B. Hayes-Roth, F. Hayes-Roth, S. Rosenschein, and S. Cammarata. Modeling planning as an incremental opportunistic process. In *Proceedings, Sixth International Joint Conference on Artificial Intelligence*, pages 375–383, 1979.

[Hildum, 1994] D.W. Hildum. Flexibility in a knowledge-based system for solving dynamic resource-constrained scheduling problems. PhD thesis and CMPSCI Technical Report 94–77, Department of Computer Science, University of Massachusetts, Amherst, September 1994.

[Huang *et al.*, 1995] S.H. Huang, H-C. Zhang, and M.L. Smith. A progressive approach for the integration of process planning and scheduling. *IIE Transactions*, 27(4):456–464, August 1995.

[Iwata and Fukuda, 1989] K. Iwata and Y. Fukuda. A new proposal of dynamic process planning in machine shop. In *Proceedings, CIRP International Workshop on Computer-Aided Process Planning*, Hannover Germany, September 1989.

[Khoshnevis and Chen, 1989] B. Khoshnevis and Q. Chen. Integration of process planning and scheduling functions. In *Proceedings, IIE Integrated Systems Conference & Society for Integrated Manufacturing Conference*, Dallas TX, November 1989.

[Lee and Billington, 1992] H.L. Lee and C. Billington. Managing supply chain inventory: Pitfalls and opportunities. *Sloan Management Review*, 33(3):65–73, Spring 1992.

[Lesser and Corkill, 1983] V.R. Lesser and D.D. Corkill. The distributed vehicle monitoring testbed: A tool for investigating distributed problem solving networks. *AI Magazine*, 4(3):15–33, Fall 1983.

[Nii *et al.*, 1982] H.P. Nii, E.A. Feigenbaum, J.J. Anton, and A.J. Rockmore. Signal-to-symbol transformation: HASP/SIAP case study. *AI Magazine*, 3(2):23–35, Spring 1982.

[Nii, 1986] H.P. Nii. Blackboard systems. *AI Magazine*, 7(2):38–53 & 7(3):82–106, 1986.

[Prasad *et al.*, 1997] Biren Prasad, Fujun Wang, and Jiati Deng. Towards a computer-supported cooperative environment for concurrent engineering. *Concurrent Engineering: Research and Applications*, 5(3):233–252, September 1997.

[Prasad, 1997] B. Prasad. *Concurrent Engineering Fundamentals: Integrated Product Development*, volume 2. Prentice Hall, 1997.

[Raytheon Company, 1993a] Raytheon Company. *Metal Fabrication Computer Aided Process Planning - Technical Specifications*, 1993. RAYCAM document #6685092A.

[Raytheon Company, 1993b] Raytheon Company. *Metal Fabrication Computer Aided Process Planning - User Manual*, 1993. RAYCAM document #6685092B.

[Sadeh *et al.*, 1993] N.M. Sadeh, S. Otsuka, and R. Schnelbach. Predictive and reactive scheduling with the Micro-Boss production scheduling and control system. In *Proceedings, IJCAI-93 Workshop on Knowledge-based Production Planning, Scheduling, and Control*, Chambery France, August 1993.

[Sadeh, 1994] N.M. Sadeh. Micro-opportunistic scheduling: The Micro-Boss factory scheduler. In M. Zweben and M.S. Fox, editors, *Intelligent Scheduling*, chapter 4, pages 99–135. Morgan Kaufmann, San Francisco CA, 1994.

[Sadeh, 1995] N.M. Sadeh. Micro-Boss: Dual-use ARPI scheduling technology helps improve manufacturing performance. *IEEE Expert*, February 1995.

[Smith, 1994] S.F. Smith. OPIS: A methodology and architecture for reactive scheduling. In M. Zweben and M.S. Fox, editors, *Intelligent Scheduling*, chapter 2, pages 29–66. Morgan Kaufmann, San Francisco CA, 1994.

[Srinivasan *et al.*, 1994] K. Srinivasan, S. Kekre, and T. Mukhopadhyay. Impact of electronic data interchange technology on JIT shipments. *Management Science*, 40(10):1291–1304, October 1994.

[Swaminathan *et al.*, 1995] J.M. Swaminathan, N.M. Sadeh, and S.F. Smith. Impact of supplier information on supply chain performance. Technical report, The Robotics Institute, Carnegie Mellon University, Pittsburgh PA, January 1995.

[Tönshoff *et al.*, 1989] H.K. Tönshoff, U. Beckendorff, and N. Anders. FLEXPLAN: A concept for intelligent process planning and scheduling. In *Proceedings, CIRP International Workshop on Computer Aided Process Planning*, Hannover Germany, September 1989.

[Zhang and Mallur, 1994] H-C. Zhang and S. Mallur. An integrated model of process planning and production scheduling. *International Journal of Computer Integrated Manufacturing*, 7(6):356–364, 1994.