AN INTEGRATED PROCESS-PLANNING / PRODUCTION-SCHEDULING SHELL FOR AGILE MANUFACTURING

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

CMU-RI-TR-96-10

May 1996

† Intelligent Coordination and Logistics Laboratory
Center for Integrated Manufacturing Decision Systems
The Robotics Institute
Carnegie Mellon University
Pittsburgh PA 15213-3890
412.268.7598 · *fax:* 412.268.5569
{SADEH,HILDUM,SFS,DAG}@ISL1.RI.CMU.EDU

‡ Software Engineering Laboratory
Raytheon Electronic Systems
Raytheon Company
Tewksbury MA 01876-0901
508.858.5756 · *fax:* 508.858.5976
LALIBERTY_THOMAS@CAEMAC.MSD.RAY.COM
MCANULTY@CAESUN.MSD.RAY.COM

Abstract

Increased reliance on agile manufacturing techniques has created a demand for systems to solve integrated process-planning and production-scheduling problems in large-scale dynamic environments. To be effective, these systems should provide user-oriented interactive functionality for managing the various user tasks and objectives and reacting to unexpected events. This paper describes the mixed-initiative problem-solving features of IP3S, an Integrated Process-Planning/Production-Scheduling shell for agile manufacturing. IP3S is a *blackboard*-based system that supports the *concurrent* development and dynamic revision of integrated process-planning and production-scheduling solutions and the *maintenance of multiple problem instances and solutions*, as well as other flexible user-oriented decision-making capabilities, allowing the user to control the scope of the problem and explore alternate tradeoffs ("what-if" scenarios) interactively. The system is scheduled for initial deployment and evaluation in a large and highly dynamic machine shop at Raytheon's Andover manufacturing facility.

This is a revised and expanded version of a paper entitled *Mixed-Initiative Management of Integrated Process-Planning and Production-Scheduling Solutions* that is to appear in the proceedings of the Artificial Intelligence and Manufacturing Research Workshop, Albuquerque NM, June 1996.

The work described in this paper is 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.

Contents

1 Introduction 1

2 Integrating Process Planning And Production Scheduling 2

3 The IP3S Blackboard Architecture 3
- 3.1 The IP3S Blackboard . . . 4
  - 3.1.1 Contexts . . . 4
  - 3.1.2 Events . . . 6
  - 3.1.3 Unresolved Issues . . . 6
- 3.2 The IP3S Controller . . . 8
- 3.3 The IP3S Problem-Solving Cycle . . . 10
- 3.4 IP3S Knowledge Sources . . . 11
  - 3.4.1 Process-Planning Knowledge Source . . . 12
  - 3.4.2 Production-Scheduling Knowledge Source . . . 13
  - 3.4.3 Analysis and Diagnosis Knowledge Sources . . . 13
  - 3.4.4 Communication Knowledge Source . . . 14

4 Two Problem-Solving Examples 15
- 4.1 Scenario One . . . 15
- 4.2 Scenario Two . . . 16

5 Summary 17

6 References 18

1 Introduction

Over the past two decades, considerable efforts have been expended in developing integrated Computer Aided Design/Computer Aided Manufacturing functionalities [Harrington, 1974, Rembold and Dillmann, 1986, Scheer, 1991]. Simultaneously, important progress has been made towards the development of integrated production-planning and control solutions [Orlicky, 1975, Goldratt, 1980, Smith, 1992, Sadeh, 1994], as well as integrated sales/marketing solutions, leading to what we can view as "islands of integration" within the enterprise [Kerr, 1991]. While the actual level of integration within each island can significantly vary from one enterprise to another and important progress still remains to be made within each of these areas, building the bridges between these islands is without any doubt the next major hurdle in developing Computer Integrated Manufacturing environments capable of effectively supporting Agile Manufacturing practices.

As companies increase the level of customization in their products, move towards smaller lot production and experiment with more flexible customer/supplier arrangements such as those made possible by Electronic Data Interchange (EDI) [Lee, 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 products and (2) efficiently work out supplier/subcontractor arrangements for these new products. These requirements depend on 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 the existing production schedule to best accommodate the current load of the facility, the status of machines, fixtures and tools, and the availability of raw materials. To effectively support such capabilities requires bridging the gap between CAD/CAM and production scheduling through the development of integrated process-planning and production-scheduling functionalities. The concurrent development and revision of integrated solutions is expected to significantly enhance the ability of manufacturing companies to adapt efficiently to changing conditions, and yield significant performance improvements (e.g., shorter lead times, increased resource utilization, enhanced due-date performance and coordination between customers and suppliers).

The process of solving integrated process-planning and production-scheduling problems in large-scale dynamic environments requires additional user-oriented interactive functionalities. The ill-defined nature of these problems, the need to interact with outside information sources (e.g., enterprise-level planning systems, raw-material suppliers, tool shops, the shop floor), the dynamic nature of requirements specification, solution generation, refinement and reoptimization, and the varying tasks and objectives of the user (e.g., process planning, production scheduling, evaluation of bids; minimizing process-plan costs, maximizing due-date performance and resource utilization) require special mixed-initiative problem-solving functionalities, such as (1) flexible decision-making control to facilitate rapid changes in the focus of attention, (2) a range of problem-solving services to address problems at multiple levels, and (3) the ability to investigate alternative problem solutions through "what-if" analysis.

Efforts to develop mixed-initiative decision-support frameworks in the areas of planning and scheduling have been reported in [Tate, 1994, Smith and Lassila, 1994, Smith *et al.*, 1996, Ferguson *et al.*, 1996]. In this paper, we describe the mixed-initiative problem-solving capabilities of IP3S, an Integrated Process-Planning and Production-Scheduling shell for agile manufacturing [Sadeh *et al.*, 1996]. IP3S is a *blackboard*-based system [Erman *et al.*, 1980, Nii, 1986a, Nii, 1986b, Corkill, 1991, Carver and Lesser, 1992] that supports the following mixed-initiative decision-making functionalities:

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 (e.g., the current load of the facility) and greater optimization of production activities through direct visibility of process alternatives and tradeoffs

2. *maintenance of multiple problem instances and solutions*, allowing the user to control the development of

the problem and explore alternative tradeoffs ("what-if" scenarios) by interactively addressing external events (e.g., new order arrivals, requests for bids, resource breakdowns) and imposing and retracting various assumptions (e.g., different delivery dates, work shifts, resource assignments and requirements), and evaluating the impact of these decisions through incremental process plan and production schedule modification

3. *flexible, user-oriented decision making*, allowing the user to take over and guide the construction and revision of solutions at multiple levels
4. *representation of declarative, domain-specific control information* for assisting automated problem solving
5. the use of a *common representation* for exchanging process-planning and production-scheduling information
6. *coordination* with outside information sources such as enterprise-level planning systems, raw-material suppliers, tool shops, and the shop floor
7. *portability and ease of integration with legacy systems*, making it possible to quickly customize the system to support the integration of process planning and production scheduling in new environments

The IP3S shell is scheduled for initial demonstration and evaluation in a machine shop at the Raytheon Electronic Systems manufacturing facility in Andover MA, an environment for which it is undergoing integration with Raytheon's IPPI process-planning system [Raytheon Company, 1993a, Raytheon Company, 1993b] and Carnegie Mellon's MICRO-BOSS scheduling system [Sadeh *et al.*, 1993, Sadeh, 1994]. With about 50% of incoming orders requiring the construction of new process plans or revision of existing ones and with over 150 CNC machine tools and over 100 people working over 3 shifts, Raytheon's Andover machine shop is a complex, highly dynamic, small-lot manufacturing environment that typifies many of the challenges agile manufacturing seeks to address.

2 Integrating Process Planning And Production Scheduling

Technical challenges in effectively supporting integrated process-planning and production-scheduling decisions in a complex and dynamic environment such as Raytheon's machine shop are multiple. From a pure process-planning perspective, the number of orders that require the generation of new process plans and production of new tools, and the sheer variety of parts and machines (and their various characteristics) present a significant challenge. As in other large machine shops, production scheduling in this environment is no easy task either. Major scheduling challenges include (1) the presence of multiple sources of uncertainty, both internal (e.g., machine breakdowns) and external (e.g., new order arrivals, delays in tool production and raw-material delivery), (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., tools, NC programs, raw materials, human operators).

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, Tonshoff *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).

In their survey of prior efforts to integrate process planning and production scheduling, Huang *et al*. identify three distinct approaches [Huang *et al.*, 1995]:

1. Non-Linear Process Planning, which generates all possible process plans ahead of time (i.e., based on static considerations) and dynamically selects between these alternatives at execution time. This is the approach taken in the FLEXPLAN system [Tonshoff *et al.*, 1989].

2. Closed-Loop Process Planning, also referred to as real-time or dynamic process planning (e.g., [Iwata and Fukuda, 1989, Tonshoff *et al.*, 1989]), where process planning attempts to take into account dynamic resource availability information.

3. Distributed Process Planning, which reduces the complexity of the closed-loop approach by subdividing integrated process-planning and production-scheduling decisions into multiple, more localized, decision phases (e.g., [Huang *et al.*, 1995]).

In practice, none of these approaches totally dominates the other two, as different manufacturing environments generally entail different levels of complexity and different operational requirements (e.g., different real-time requirements). In fact, the decision flows assumed in earlier work are rather restrictive and cannot accommodate some of the complexities of environments such as the Raytheon machine shop. For instance, at the Raytheon facility, new process plans often require the production of new custom tools. As a result, evaluation and scheduling of process plans in this environment requires tight coordination with an internal tool shop. In general, coordination with suppliers, whether internal or external, has been ignored in earlier work to develop integrated process-planning and production-scheduling solutions. Another challenging aspect of the Raytheon machine shop has to do with the fact that it is not a single-user environment. Instead, five industrial engineers work concurrently on the generation and modification of process plans, each one of them making decisions that influence the future load of the facility. The result is a dynamic environment where it is not always easy to predict future resource loads.

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, 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], scheduling [Smith, 1994, Hildum, 1994]). 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 (in the case of IP3S, process-planning and production-scheduling knowledge) and control knowledge, blackboard architectures offer several key advantages:

- *flexibility of the control mechanism*, making it possible for the user to select from among a dynamic set of 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 figure out how to accomplish them)
- *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., KSs to generate resource-utilization statistics to help evaluate resource contention in different situations)—and a Motif-based graphical user interface (GUI). These components are described further in the rest of this section.
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 client), and the KSs (each operating separately and alternating between server and client roles) run as independent processes that communicate with each other using Expersoft's CORBA (Common Object Request Broker Architecture)-based XShell™ 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 *contexts* that correspond to different sets of working assumptions (e.g., the set of orders that need to be planned and scheduled, available resource capacities) and different solutions (e.g., process plans and production schedules). Within each 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 issues are selected to be resolved, a particular method of resolution is selected from among the set of methods applicable to the unresolved issue(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 to perform "what-if" analysis.
In the remainder of this section we describe the major architectural features of the IP3S blackboard, with an emphasis on the mixed-initiative problem-solving 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 human operators), 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 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.

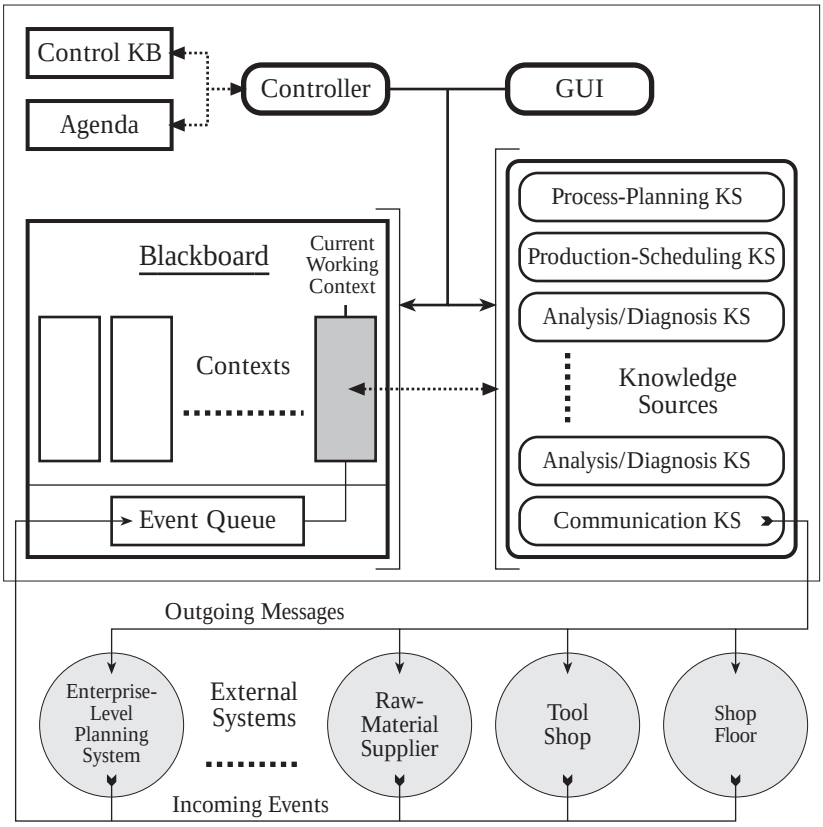

Figure 1: Overview of the IP3S Blackboard Architecture

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 incorporation of events into a context (e.g., from external sources like an enterprise-level planning system, raw-material suppliers, a tool shop, the shop floor) 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. This enables either the user or the system to evaluate the implications of alternative assumptions. When the Controller invokes a particular KS (either automatically or through interaction with the user), the result produced by the KS is only used to modify the context in which the user is currently working (called the "current working contest"). At any point (given proper permission), the user may release to the shop floor the solution (e.g., process plans and production schedule) associated with a particular context. This context then becomes known as the *released* (or *production*) context. The IP3S GUI enables the user to create contexts, navigate across them, compare alternative assumptions and solutions, and delete contexts.

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. Table 1 provides a listing of IP3S events. 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.[1]
The following two examples demonstrate the mixed-initiative capabilities supported by the event-incorporation mechanism:

1. It enables 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 be ignored.

2. It enables both the user and the system to process conditional events (e.g., in contrast to a mechanism where all incoming events would be incorporated automatically into all contexts). 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 on the order.

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

[1] A qualification on this condition will be introduced later.

Order related:

- *via enterprise-level planning system:*
  - Incoming-Order
  - Order-Update
  - Order-Cancelation
- *via the Process-Planning KS:*
  - Process-Plan-Update

Bid related:

- *via the enterprise-level planning system:*
  - Request-for-Bid

Resource related:

- *via the shop floor:*
  - Resource-Breakdown
  - Shop-Floor-Update
- *via the shop floor or a supplier:*
  - Estimated-Replenishment-Date
  - Resource-Replenishment
- *via the tool shop:*
  - Estimated-Tool-Completion-Date
  - Tool-Completion

Table 1: A sampling of IP3S events

effort remains to be done to produce a complete and satisfactory solution. It provides a powerful *workflow management mechanism* that helps IP3S users keep track of work that remains to be done in a given context and determine how to proceed to remove any remaining unresolved issues in that context.
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 current process plan is not the one used in the existing production schedule
3. potential areas for solution *improvement*, such as an order with an excessively late completion date or long lead time

Table 2 provides a list of IP3S unresolved issues. To refine a previous statement, note that a complete problem solution for a particular context exists so long as all *completeness*- and *inconsistency*-related unresolved issues have been resolved (unless, for an inconsistency, the issue fails to invalidate the current solution, as is the case with Order-to-be-Canceled). *Improvement*-related unresolved issues may exist in conjunction with a satisfactory solution. To provide a means for the user to alter some of the problem-solving objectives as conditions warrant, some unresolved issues can be adjusted so that they will only be created if conditions exceed a particular threshold set by the user. Such is the case with the Tardy-Order and Tardy-Schedule issues, where the user may desire not to be alerted to tardiness in the schedule unless it reaches some critical degree. In addition, the user can select which unresolved issues will be displayed through the GUI, so that unimportant issue types can be "turned off" to prevent distraction. While these issues will still be created on the blackboard, they will remain hidden from the user. The set of issues displayed through the GUI can be modified at any time.
The roles of the blackboard, contexts, events and unresolved issues in supporting mixed-initiative problem-solving in IP3S 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 the selection of which events to incorporate into the current context, the determination of which unresolved issues to resolve, and the selection of the specific methods for their resolution
2. *support for multiple-level customizable problem-solving services* 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)

The result is a system where the user can select from among different levels of interaction and different control regimes at any time. The dynamic nature of both of these features is important to emphasize. IP3S can alternate between different control regimes at any point in the problem-solving process, allowing the user to tailor the degree of interaction with the system according to the current state of problem-solving and conditions within the environment. In addition, the set of high-level problem-solving services provided to the user can easily be augmented to accommodate changing user-interaction patterns. A hierarchy of high-level problem-solving services 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.

Order related:

- *Completeness:*
  - Order-w/o-Default-Process-Plan
  - Order-w/o-Approved-Process-Plan
  - Order-w/o-Production-Schedule
  - Order-w-Unscheduled-Operation(s)
- *Inconsistency:*
  - Order-w/-Inconsistent-Production-Schedule
  - Order-to-be-Canceled
- *Improvement:*
  - Tardy-Order (parameterized)
  - Order-w-Long-Lead-Time (parameterized)

Production-Schedule related:

- *Inconsistency:*
  - Unprocessed-Shop-Floor-Update
  - Unprocessed-Replenishment-Date
  - Unprocessed-Tool-Completion-Date
- *Improvement:*
  - Tardy-Schedule (parameterized)

Resource/Raw Material related:

- *Improvement:*
  - Freed-Capacity

Resource-Utilization-Statistics related:

- *Inconsistency:*
  - Outdated-Resource-Utilization-Statistics

Communication related:

- *Completeness:*
  - Query-Awaiting-Response

Table 2: A sampling of IP3S unresolved issues

To support these mixed-initiative capabilities, the IP3S Controller works off of an *execution profile* that records the assignment of various problem-solving tasks (e.g., event incorporation, unresolved-issue and resolution-method selection) to either the system or the user. The assignment of tasks can be 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 *control knowledge base* that links each unresolved issue to the set of problem-solving services applicable for its resolution. When an unresolved issue is selected for resolution, the Controller assembles the list of methods that can be activated and executed to modify the solution and get rid of the selected unresolved issue. Constructed entirely from declarative domain-specific information, this knowledge base can be augmented easily (e.g., in response to the discovery of new patterns of user interaction). The 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.

The IP3S Controller uses an *agenda* mechanism to keep track of the problem-solving tasks that need to be executed. When a particular course of action is selected, either interactively 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 the action or actions to be performed next by the system. The IP3S 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 (e.g., Improve-Order-Completion-Date) 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., Process-Order, which calls the Process-Planning KS's Plan-Order service and the Production-Scheduling KS's Schedule-Order(s) service in succession), possibly depending on some context-specific considerations or control heuristics

3. *scripts*, which specify a predefined sequence of KS services and goals generally known to accomplish a particular problem-solving task (e.g., Process-Order)

3.3 The IP3S Problem-Solving Cycle

The decision flow through the IP3S architecture is summarized in Figure 2. Each problem-solving cycle in IP3S begins with either the incorporation of a new event into the current working context or the modification of an assumption within the current working context ("what-if" analysis), both of which are actions that can be performed by either the user or the Controller (as specified by the execution profile). In reference to the figure, the decision flow 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 (a task handled by the blackboard)

2. selecting an unresolved issue to resolve (a task handled by either the user or the Controller, depending on the execution profile)

3. selecting a resolution method for the selected unresolved issue (a task handled by either the user or the Controller, depending on the execution profile)

4. activating the selected resolution method (a task handled by the Controller)

5. executing the problem-solving service that corresponds to the activated resolution method (a task handled by the appropriate KS)

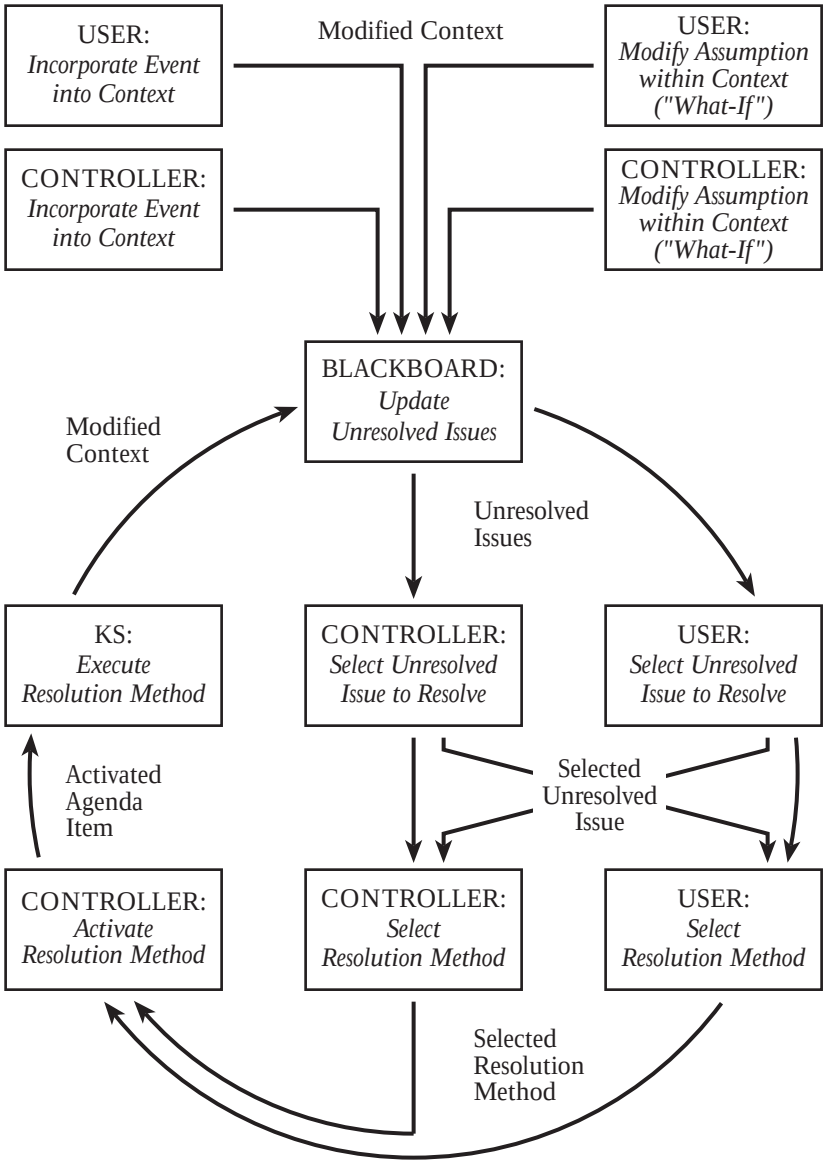

Figure 2: IP3S Control Loop

3.4 IP3S Knowledge Sources

Knowledge sources serve as the primary problem solvers in a blackboard system. KSs communicate their results by posting new information to the blackboard (e.g., new process plans and production schedules) and modifying existing information (e.g., updated process plans and reoptimized production schedules).
In IP3S, each domain-level KS acts primarily as a server that supports a variety of problem-solving services that can be activated, scheduled and invoked by the Controller. Each KS service may require a number of parameters.

The parameters for each problem-solving service are defined by the unresolved issue for which that service is applicable. When an agenda item (corresponding to a selected service, goal or script) is activated, the Controller assembles the collection of parameters as specified by the unresolved issue selected for resolution, and includes them with the item. When a script is activated, appropriate agenda items are activated for each of its steps, and the overall parameter list is passed with each item. The parameters provided to a KS service act as the traditional "stimulus units" to the knowledge source called to perform the designated service. It is the responsibility of each encapsulated problem-solving system that serves as an IP3S KS to provide a common dispatching method that (1) accesses the necessary parameters to a service, and (2) calls the appropriate internal method to perform the service. Each KS also returns a special "result" object indicating whether it has completed its designated task successfully. The information within these objects is used by the IP3S Controller to help determine the next course of action.

The design of the IP3S KS mechanism facilitates integration with legacy problem-solving systems and thereby provides a significant degree of architectural flexibility for dealing with a wide range of practical problem domains. By supporting a set of *core* problem-solving services and providing a *wrapper* around itself capable of communicating with the IP3S blackboard, an independent problem-solving system can be easily encapsulated as an IP3S KS. Furthermore, additional problem-solving services can be provided on-the-fly by any KS by simply notifying the Controller and specifying which unresolved issues they are capable of resolving.

In the sections that follow, we describe the core group of IP3S KSs and the core group of problem-solving services that each KS must provide to IP3S, plus some additional services provided as a benefit of the particular independent problem-solving systems encapsulated to implement each KS.

3.4.1 Process-Planning Knowledge Source

The IP3S Process-Planning KS is implemented by Raytheon's IPPI generative process planner [Raytheon Company, 1993a, Raytheon Company, 1993b], a system currently under development. IPPI utilizes knowledge bases populated with raw-stock configurations, process-selection logic and manufacturing-resource capabilities. Machine-tool selection and tool-path generation are handled by invoking CUTTECH™, a system developed by the Institute of Advanced Manufacturing Sciences (IAMS). Process plans produced by IPPI consist of:

- an ordered list of machining operations, including recommended tooling requirements and feeds and speeds
- process-routing information, showing predecessor and successor relationships between operations
- a bill of materials

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 posted on the IP3S blackboard by the Resource-Utilization KS. As a result, IPPI is able to make process-planning decisions that avoid the use of otherwise unknown bottleneck resources. This capability helps support the mixed-initiative power of IP3S by facilitating the interleaving of process planning and production scheduling so that valuable production-scheduling information (e.g., resource-utilization statistics) can be exploited during the process-planning phase.

The development of detailed process plans containing complete tooling information remains a time-intensive task, especially when new tools are required as part of a plan. IP3S therefore differentiates between "default" and "approved" process plans, depending on whether complete tooling information and approval by an industrial engineer (IE) is included. Default process plans use default machining requirements, estimated durations, and lack complete tooling information. Nor do they have the explicit approval of an IE. This distinction (and IPPI's support of it) makes it possible for IP3S to provide a type of process plan that can be constructed quickly and used by other IP3S KSs to inform their own problem-solving processes.

The IP3S problem-solving services provided by IPPI are described below:

- Plan-Order(s)-Default [core service]: generates a default process plan for one or more orders (i.e., for use by the IP3S Resource-Utilization KS)
- Plan-Order [core service]: generates a detailed and approved process plan (i.e., one approved by an IE)

3.4.2 Production-Scheduling Knowledge Source

The IP3S Production-Scheduling KS is implemented by the MICRO-BOSS system [Sadeh *et al.*, 1993, Sadeh, 1994], a dynamic finite-capacity scheduling tool developed at Carnegie Mellon University to support efficient just-in-time operation in complex manufacturing environments subject to rapidly changing conditions. MICRO-BOSS builds schedules by constantly monitoring resource contention during the construction or repair of a schedule and dynamically redirecting its optimization efforts towards areas of the search space subject to the highest contention (i.e., groups of operations contending for critical resource/time intervals). MICRO-BOSS supports predictive, reactive and interactive scheduling functionalities and has been shown to consistently yield significant improvements in schedule quality (i.e., due-date satisfaction, lead-time and inventory performance) over multiple combinations of priority dispatch rules and release policies as well as over sophisticated bottleneck-centered scheduling techniques. In its capacity as the IP3S Production-Scheduling KS, MICRO-BOSS is capable of building schedules for both default and approved process plans.
The IP3S problem-solving services provided by MICRO-BOSS are described below:

- Schedule-Order(s) [core service]: generates a schedule for one or more orders that attempts to complete all orders by their due dates while minimizing lead times and inventory costs
- Modify-Schedule [core service]: generates a reoptimized schedule that incorporates and then repairs specific changes to the existing schedule (e.g., orders with modified or inconsistent process routings, orders to be canceled, freed resource capacity)
- Update-Schedule [core service]: reoptimizes parts of an existing schedule to accommodate deviations from the original schedule (e.g., machine breakdowns and execution delays, raw-material replenishment dates, tool-completion dates) given updated information from external sources (e.g., the shop floor, raw-material suppliers, the tool shop)
- Schedule-Order(s)-Reoptimize: generates a reoptimized schedule that incorporates one or more additional orders into the existing schedule

To support "what-if" analysis in IP3S, MICRO-BOSS provides an additional editing capability that facilitates interactive user manipulation of an existing schedule. Using this functionality, the user can modify various scheduling assumptions within a context by altering the existing schedule in certain ways (e.g., by scheduling and repositioning operations, changing the number of shifts on a resource, changing the due date for an order). This capability is provided as one of the editing options accessible through the IP3S GUI.

3.4.3 Analysis and Diagnosis Knowledge Sources

The IP3S approach is based on the general premise that the complexity of the combined process-planning/ production-scheduling search space can effectively be reduced and solution quality enhanced by using process-planning considerations to help focus search within the scheduling subspace and, conversely, by using production-scheduling considerations to quickly identify promising alternatives within the process-planning subspace. Analysis/diagnosis KSs are central to achieving this integrated search behavior. They help identify sources of inefficiency in the current solution and determine how the solution can most effectively be improved (e.g., whether to generate an alternative process plan for a given part, modify its current tooling requirements, or reschedule operations

on a critical machine). Analysis results are summarized on the blackboard, where they are accessible to the user, other KSs, and the IP3S Controller.
One key analysis KS in IP3S is the Resource-Utilization KS (currently implemented by MICRO-BOSS). This KS estimates resource contention, by accounting for both current reservations within the existing schedule and projected demand from unscheduled orders. Specifically, given one or several groups of resources, the Resource-Utilization KS computes contention over specified time buckets (e.g., daily or weekly), and generates, for each resource/time bucket, a triplet consisting of (1) the total capacity available on the resource within the given time bucket, (2) the amount of capacity that has already been committed (i.e., existing reservations), and (3) an estimate of the capacity that would be required to accommodate all remaining unscheduled orders if they were each to be optimally scheduled.
The core service provided by this KS is Update-Resource-Utilization-Statistics. It accepts a variety of resource and time interval combinations for which to compute contention measures. In particular, it is possible to request statistics for all the groups of resources that could possibly be used to perform a given operation (e.g., groups of milling machines with different characteristics). In this case, projected demand from the operation itself is not included in the contention metrics computed by the KS. The results produced by this KS, which are posted on the blackboard (in the current context), can be displayed graphically through a menu option of the IP3S GUI. They can then be used by the Process-Planning KS to identify promising process alternatives.
Examples of more complex analysis/diagnosis KSs currently envisioned include:

- *KSs to identify solution inefficiencies and opportunities for solution improvement*, such as late orders, orders whose lead times or engineering costs seem particularly high, underutilized resources
- *solution-improvement KSs* to help identify promising ways of improving specific solution inefficiencies, such as an order whose completion date is delayed because of contention for a bottleneck machine (a KS could evaluate the impact of using alternate process plans for either the tardy order or the orders using the bottleneck machine and post its results to the blackboard, where they could be used by the IP3S control heuristics or the user to decide how to proceed)

3.4.4 Communication Knowledge Source

The IP3S Communication KS facilitates communication between the IP3S system 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. The Communication KS is equipped with the necessary knowledge for constructing messages for conveying specific kinds of information to external systems. It passes information in a particular format as determined by the destination and the type of information being communicated.
The IP3S system supports a basic message hierarchy for communicating with external systems. This hierarchy is divided into three classes, corresponding to queries, responses to external queries, and notifications. Query messages are sent to request specific information from external sources (e.g., estimated resource-replenishment dates, completion dates for new tools). Response messages are sent in response to previously received queries (e.g., for responding to requests for bid). Notification messages are sent to notify external systems about important system developments (e.g., order completion dates), in which case no response is expected. Table 3 provides a list of IP3S Communication KS message types.
Whenever a query is transmitted, a corresponding Query-Awaiting-Response unresolved issue is created within the current context to indicate that a reply to that message is expected. When the reply is received (in the form of a specific response event), it is only announced (or "made visible") to the context containing its corresponding Query-Awaiting-Response unresolved issue. This ensures that only the context from within which a query is transmitted will receive the reply. (Each response message contains a tag connecting it with the original query.) When a

Query:
- *to the shop floor or a supplier:*
  Replenishment-Date-Request
- *to the tool shop:*
  Tool-Completion-Date-Request

Response:
- *to the enterprise-level planning system:*
  Bid-Submission

Notification:
- *to the enterprise-level planning system:*
  Order-Completion-Date
- *to the shop floor:*
  Production-Schedule-Update
- *to the shop floor or a supplier:*
  Expected-Replenishment-Date
- *to the tool shop:*
  Expected-Tool-Completion-Date

Table 3: A sampling of IP3S Communication KS message types

response event (e.g., Estimated-Tool-Completion-Date) is incorporated into a context, the corresponding Query-Awaiting-Response unresolved issue is removed. The Communication KS provides three core communication services, namely Send-Query, Send-Response, and Send-Notification.

4 Two Problem-Solving Examples

To better illustrate the mixed-initiative capabilities of IP3S, we provide the following two problem-solving scenarios. The first demonstrates a highly interactive session of generating a process plan for a new order that considers existing levels of resource contention before adding it to an existing production schedule. The second scenario demonstrates a less interactive session where the user attempts to improve an existing schedule built automatically by the system. (For brevity, some of the lower-level details in these scenarios have been left out.)

4.1 Scenario One

The user incorporates an Incoming-Order event sent to IP3S by the enterprise-level planning system into the current working context (containing an existing schedule). The event includes all of the necessary information about the order (e.g., its part specification, quantity, due date). The IP3S blackboard generates an Order-w/o-Default-Process-Plan unresolved issue to indicate that the new order lacks a process plan and adds it to the context. In order to understand the impact of the expected demand resulting from the ideal resource requirements for the new order, the user decides to resolve this issue by invoking the Process-Planning KS to perform the Plan-Order(s)-Default service (one of a list of all applicable resolution methods for the issue that is displayed by the GUI). This service will generate a default process plan for the new order and post it to the blackboard. Following the assignment of the new default process plan to the order, the blackboard replaces the Order-w/o-Default-Process-Plan issue with

a new Order-w/o-Approved-Process-Plan issue and generates a new Outdated-Resource-Utilization-Statistics issue to signal the change in resource demand reflected by the new default process plan. The user now chooses to update the resource-utilization statistics for the current working context to include the demand from the new order's default process plan. This is done by invoking the Resource-Utilization KS to perform the Update-Resource-Utilization-Statistics service. Following the updating of the resource-utilization statistics, the blackboard deletes the Outdated-Resource-Utilization-Statistics issue. The user, upon noticing that the default process plan requires one or more resources for which contention is already high, now invokes the Process-Planning KS to perform the Plan-Order service in an attempt to avoid using those specific resources. With the help of an IE, this service will generate an approved process plan for the new order—taking into consideration the updated resource-utilization statistics—and send it to IP3S (the generation of an approved plan is not a fully automated process, so the new plan is transmitted to the system as a Process-Plan-Update event). 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.

The user proceeds by incorporating the Process-Plan-Update event into the current working context, which results in the replacement of the new order's previous default process plan with the new approved plan, and the replacement of 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 (or *process routing*). The user can now invoke the Production-Scheduling KS to perform the Schedule-Order(s) service. This service will incorporate the order, according to its new approved process plan, into the existing production schedule within the current working context. Upon its completion, the blackboard will delete the Order-w/o-Production-Schedule unresolved issue. 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 now complete (for the moment). The user has succeeded in interleaving the process-planning and production-scheduling processes to generate a higher-quality problem solution.

4.2 Scenario Two

As problem-solving proceeds automatically by the IP3S Controller, the user detects a Tardy-Order unresolved issue within the current working context. In order to perform some "what-if" analysis to attempt to resolve this issue, the user instructs the system to pause at the end of the current cycle. At this point, the user creates a copy of the current working context and makes the copy the new current working context. An Improve-Completion-Date goal is then posted to attempt to resolve the issue. The Controller's heuristics suggest the use of the Use-Alternate-for-Bottleneck-Resource script to satisfy the goal, upon detecting the tardy order's dependence on a number of heavily utilized resources. The script proceeds by calling (1) a Bottleneck-Analysis KS to select the most critical bottleneck resource being used by the order (among all of the resources that have alternates in its plan), (2) the Process-Planning KS to modify the plan to use an alternate resource, and (3) the Production-Scheduling KS's Modify-Schedule service to reoptimize the existing schedule using the modified plan for the tardy order.

Assuming, in this situation, that the execution of the Use-Alternate-for-Bottleneck-Resource script results in a schedule that delivers the previously tardy order by its original due date, the Improve-Completion-Date goal is marked as satisfied and the blackboard deletes the Tardy-Order unresolved issue (following the updating of the existing schedule by the Production-Scheduling KS). Automatic problem-solving by the Controller can now continue within the copied context. Had the execution of the script failed to satisfy the goal, alternate satisfaction methods—other scripts or services ranked by goal-specific control heuristics—could be attempted in different copied contexts (e.g., the Free-Capacity-on-Bottleneck-Resource could be activated to find another order whose use of the critical bottleneck resource could be eliminated by modifying its process plan), or the previous course of problem solving could be resumed.

5 Summary

A key requirement in supporting agile manufacturing practices is the ability to (1) rapidly convert standard-based product specifications into process plans and machine operations and (2) quickly and effectively integrate process plans for new orders into the existing production schedule. In contrast to traditional manufacturing practice, where process planning and production scheduling are treated as two independent processes, we have developed an integrated process-planning and production-scheduling shell (IP3S) capable of supporting concurrent process planning and production scheduling. IP3S is designed around an innovative *blackboard architecture* that provides flexible, mixed-initiative decision-making functionalities to support user-oriented management of integrated process-planning and production-scheduling solutions in large-scale dynamic environments. IP3S is expected to significantly boost the ability of companies to adapt to rapidly changing conditions, both external and internal, and yield significant improvements in manufacturing performance. An initial prototype of IP3S has been developed, and the system is currently undergoing further refinement and evaluation.

6 References

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

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

[Carver and Lesser, 1992] Norman Carver and Victor R. Lesser. The evolution of blackboard control architectures. CMPSCI Technical Report 92–71, University of Massachusetts, Amherst, October 1992.

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

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

[Ferguson *et al.*, 1996] George Ferguson, James F. Allen, and Brad Miller. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings, Third International Conference on Artificial Intelligence Planning Systems (AIPS-96)*, Edinburgh Scotland, May 1996. American Association for Artificial Intelligence (AAAI).

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

[Goldratt, 1980] Eliyahu M. Goldratt. Optimized Production Timetable: Beyond MRP: Something better is finally here. Speech given at the APICS National Conference, Los Angeles CA. American Production and Inventory Control Society (APICS), 1980.

[Harrington, 1974] Joseph R. Harrington. *Computer Integrated Manufacturing*. Industrial Press, New York, 1974.

[Hildum, 1994] David W. Hildum. *Flexibility in a Knowledge-Based System for Solving Dynamic Resource-Constrained Scheduling Problems*. PhD thesis, University of Massachusetts, Amherst, September 1994.

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

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

[Kerr, 1991] Roger Kerr. *Knowledge-based Manufacturing Management*. Addison-Wesley, Singapore, 1991.

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

[Lee, 1992] H.L. Lee. Managing supply chain inventory: Pitfalls and opportunities. *Sloan Management Review*, 1992.

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

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

[Nii, 1986a] H. Penny Nii. Blackboard systems (part one): The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, 7(2):38–53, 1986.

[Nii, 1986b] H. Penny Nii. Blackboard systems (part two): Blackboard application systems, blackboard systems from a knowledge engineering perspective. *AI Magazine*, 7(3):82–106, 1986.

[Orlicky, 1975] Joseph Orlicky. *Material Requirements Planning: The New Way of Life in Production and Inventory Management*. McGraw-Hill, New York, 1975.

[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.

[Rembold and Dillmann, 1986] Ulrich Rembold and R. Dillmann, editors. *Computer-Aided Design and Manufacturing: Methods and Tools*. Springer-Verlag, Berlin, 1986.

[Sadeh *et al.*, 1993] Norman M. Sadeh, Shinichi Otsuka, and Robert 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. International Joint Conferences on Artificial Intelligence (IJCAI).

[Sadeh *et al.*, 1996] Norman M. Sadeh, Thomas J. Laliberty, David W. Hildum, John McA'Nulty, Robert V.E. Bryant, Stephen F. Smith, David Flood, and Ann Gardner. Development of an integrated process planning/production scheduling shell for agile manufacturing. In *Proceedings, Fifth National Agility Conference*, Boston MA, March 1996. Agile Manufacturing Enterprise Forum (Agility Forum).

[Sadeh, 1994] Norman M. Sadeh. Micro-opportunistic scheduling: The MICRO-BOSS factory scheduler. In Monte Zweben and Mark S. Fox, editors, *Intelligent Scheduling*, chapter 4. Morgan Kaufmann, San Francisco CA, 1994.

[Scheer, 1991] August-Wilhem. Scheer. *CIM: Computer Integrated Manufacturing: Towards the Factory of the Future*. Springer-Verlag, Berlin, 1991.

[Smith and Lassila, 1994] Stephen F. Smith and Ora Lassila. Toward the development of flexible mixed-initiative scheduling tools. In *Proceedings, ARPA Planning Workshop*, Tucson AZ, February 1994.

[Smith *et al.*, 1996] Stephen F. Smith, Ora Lassila, and Marcel Becker. Configurable, mixed-initiative systems for planning and scheduling. In Austin Tate, editor, *Advanced Planning Technology*. AAAI Press, Menlo Park CA, 1996.

[Smith, 1992] Stephen F. Smith. Knowledge-based production management: Approaches, results, and prospects. *Production Planning and Control*, 3(4), 1992.

[Smith, 1994] Stephen F. Smith. OPIS: A methodology and architecture for reactive scheduling. In Monte Zweben and Mark S. Fox, editors, *Intelligent Scheduling*, chapter 2. 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*, 1994.

[Swaminathan *et al.*, 1995] Jay Swaminathan, Norman M. Sadeh, and Stephen F. Smith. Impact of supplier information on supply chain performance. Technical report, Robotics Institute, Carnegie Mellon University, 1995.

[Tate, 1994] Austin Tate. Mixed initiative planning in O-Plan2. In *Proceedings, ARPA Planning Workshop*, Tucson AZ, February 1994.

[Tonshoff *et al.*, 1989] H.K. Tonshoff, U. Beckendorff, and N. Andres. FLEXPLAN - a concept for intelligent process planning and scheduling. In *Proceedings, International Workshop on Computer Aided Process Planning*, Germany, September 1989. International Institution for Production Engineering Research (CIRP).

[Zhang and Mallur, 1994] Hong-Chao Zhang and Srinidhi Mallur. An integrated model of process planning and production scheduling. *International Journal of Computer Integrated Manufacturing*, 7(6), 1994.