Issues in the Integration of Planning and Scheduling for Enterprise Control

Karen L. Myers
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
myers@ai.sri.com

Stephen F. Smith
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@cs.cmu.edu

Abstract

Effective control of large-scale enterprises requires a combination of long-range planning to identify appropriate strategies for meeting objectives and efficient scheduling of resources to ensure timely completion of tasks associated with those strategies. Given the highly dynamic operating environment of many enterprises, planning and scheduling must both be adaptive to unexpected events and tightly linked to ensure responsiveness and agility. This paper discusses key issues in integrating planning and scheduling technologies to support ongoing management of enterprise operations in dynamic environments. It outlines a series of models for integrating planning and scheduling technologies, and discusses their strengths and weaknesses with respect to continuous operations management for complex enterprises. The paper also describes a new research effort that seeks to develop a dynamic integrated planning and scheduling system that can support a broad range of interactions and control strategies for the domain of air operations.

1 Introduction

The effective operation of large-scale enterprises poses significant planning and scheduling problems. Complex and frequently conflicting organizational objectives must be translated into coordinated sets of executable actions, and finite resources of the enterprise must be assigned to these actions to enable their execution in a timely and cost-effective fashion. The dynamics of the operating environment further complicate matters. As execution proceeds, unanticipated and evolving circumstances quickly and regularly force changes to strategic objectives, to planned activities, and to established resource assignments. The effectiveness of the enterprise is ultimately a function of its ability to respond to change, which in turn depends on the enterprise's ability to efficiently and appropriately adapt and manage plans and schedules over time.

One recognized obstacle to organizational responsiveness to change is poor integration of "planning" and "scheduling" processes. In manufacturing organizations, this problem has been characterized as the "wall between engineering and manufacturing". Similar sorts of barriers can be found in other large-scale enterprises. The crux of the issue is lack of communication; plans are developed with no consideration of resource availability and operational status, and likewise, schedules are developed and managed without knowledge of objectives and dependencies. Without such information exchange, reaction to unforeseen problems and opportunities necessarily proceeds in an undirected and hence inefficient manner.

A second, somewhat related obstacle to responsiveness and agility in dynamic environments is a lack of incrementality in planning and scheduling processes. In large-scale enterprises, where executing units are distributed and semi-autonomous, it is important to maintain stability and continuity in the planning and scheduling decisions that are made over time, and to minimize the impact of changes that are required to realign to current environmental circumstances. Many planning and scheduling tools/processes are not designed with incrementality in mind.

This paper considers issues related to the integration of planning and scheduling to support ongoing management of enterprise operations in a dynamic environment. By planning, we refer generally to the process of deciding *what* to do; i.e., the process of transforming strategic objectives into executable task (or activity) networks. We use the term scheduling generally to designate the process of deciding *when* and *how*; i.e., which resources to use to execute various activities and over what time frames.

In this paper, we outline a series of models for integrating stand-alone planning and scheduling processes, and summarize the merits of each from the standpoint of agile enterprise control (Section 2). Each model progressively increases the level of information flow between component processes and hence the degree of coupling between them. In Section 3, we consider issues that arise when planning and scheduling must operate over extended periods of time. Such continuous operation requires the ability to perform incremental extensions to plans and schedules, as well as to instigate repairs in response to good or bad partial execution results and unexpected events.

While there have been previous attempts to build integrated planning and scheduling technologies for complex enterprise control (as described in Section 4), the question of how best to design such systems remains open. In Section 5, we briefly describe a research effort focused on developing an integrated planner/scheduler framework in which to explore these issues further. The integrated system will build

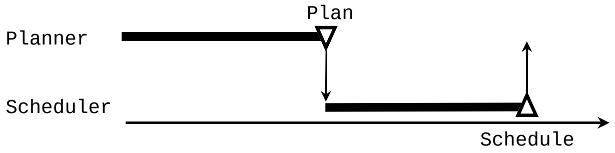

Figure 1: Waterfall Model for Integrated Planning and Scheduling

on component planning [8] and scheduling [14] technologies developed previously by the authors. Both technologies were designed to support incremental change capabilities, and hence are well-suited individually to the requirements of agile enterprise control.

2 Integration Models

The integration of planning and scheduling can proceed along a spectrum from *loose* to *tight* coupling, where coupling can be characterized both in terms of the frequency of interaction and the amount and type of information communicated or shared between them. In this section, we consider several alternative integration models, in order of increased degree of coupling. For each, we describe the basic integration model, requirements imposed by the model on planning and scheduling technologies, and the advantages and disadvantages of each.

2.1 Iterative Waterfall

The simplest approach to integrating planning and scheduling technologies is through an *iterative waterfall* model, very much akin to the classical organizational separation of planning and scheduling mentioned above. Under this scheme, the planner and scheduler operate in sequential, lock-step fashion (see Figure 1). The planner produces a complete set of tasks for achieving a set of objectives (i.e., a complete plan) and then passes them to the scheduler for time and resource assignments. Of course, not all generated plans will be schedulable. Depending on resource availability, some tasks may not be executable within satisfactory time frames or likewise some tasks may need to be dropped from the plan to achieve time and resource feasibility. In such cases, it is necessary to reinvoke the planner to produce an alternative set of tasks. In, general this process will repeat until an acceptable solution (plan/schedule) is obtained.

Realization of an integrated planning and scheduling system based on the iterative waterfall model requires little beyond standard capabilities found in current planning and scheduling systems. Other than the basic plan and schedule generation capabilities, the only requirement is that the planner be able to generate alternative solutions. Ideally, the planner would produce plans that are *qualitatively distinct* with respect to scheduling requirements [9]; otherwise, the differences between successive solutions may be insufficient to eliminate the sources of resource and/or time infeasibility.

In providing a loosely coupled form of integration, the simple waterfall model promotes compartmentalized reasoning. This approach can lead to inefficient and ineffective performance in domains where tasks are resource constrained. There are two contributing factors:

- *Low frequency of interaction* - In the basic case, complete plans are generated before any consideration is given to resource availability concerns, and hence significant effort can be spent expanding strategic process alternatives that could be quickly dismissed due to shortages in required resources.
- *Low information exchange* - Compounding the problem caused by infrequent interaction, there is also a minimal level of information exchange and feedback between planner and scheduler within the iterative waterfall model. In the simplest case, there is no “upward” flow of information from scheduler to planner in cases where a resource-feasible (or otherwise acceptable relaxed) schedule cannot be found, and hence no guidance is provided for producing a set of tasks that is “easier” to schedule. Likewise, no information is passed from planner to scheduler regarding the flexibility (or lack thereof) associated with satisfying various constraints associated with planned tasks. Without such knowledge of input constraints, any relaxation decisions that the scheduler must make to achieve resource feasibility (e.g., slipping deadlines of various tasks) are similarly unfocused from the standpoint of planner acceptability.

2.2 Feasibility Checking

Feasibility checking constitutes one method for increasing the likelihood that generated plans are viable with respect to available resources can be increased. Feasibility checking involves the use of scheduling actions to filter the strategic decisions under consideration by the planner, based on characterizations of the resource requirements that those decisions entail. In short, the scheduler is invoked at intermediate decision points during plan generation to provide estimates of resource feasibility of different planning alternatives.

In standard hierarchical planning approaches [3, 16], strategic choices (which encapsulate high-level tasks or processes) are refined into increasingly more detailed task networks as the planner’s search proceeds. Incorporation of feasibility checking into this sort of planning process requires a number of extensions. The *operators* used to encode planning strategies at different levels of abstraction must be extended to include explicit estimates of resource requirements. In addition, it must be possible to configure the scheduler to reason at comparable levels of abstraction. Finally, the planner must be capable of utilizing resource feasibility results produced by the scheduler to discriminate among decision options. In general, the problem of generating effective abstractions of detailed resource requirements is quite difficult. However, even conservative approximations can provide a basis for significant early pruning of alternatives in specific circumstances.

Feasibility checking can be performed at different levels of granularity and precision. A *capacity analysis* profiles the demand for various resources over time, identifying periods of definite, possible or likely resource contention or oversubscription. Capacity analysis reasons at the level of resource aggregates, rather than individuals, and may also ignore other resource allocation constraints. In cases where relaxed versions of the scheduling problem are solved (i.e., selected constraints are omitted), any detected conflicts are guaranteed to be conflicts in the full problem; however, there is no guarantee that all conflicts have been found. Similarly, it is possible to efficiently compute periods where some possibility of resource contention exists. Depending on the nature of the problem constraints and their interactions, however, neither relaxed formulations nor computation of possible conflicts may provide sufficient basis for conflict detection. In such cases, approximate computation of resource demand profiles may provide a more effective approach. However, this form of capacity analysis introduces the possibility of detecting false resource conflicts.

Capacity analysis provides a quick but possibly incomplete or inaccurate estimation of the resource feasibility of a given plan (or plan fragment). *Resource allocation,* alternatively, provides a more detailed approach by assigning individual assets to specific activities, thus precluding their use by other activities within designated time windows. While resource allocation provides a more accurate assessment of resource feasibility, it is more costly to perform than capacity analysis. The choice of which strategy to use for feasibility checking will depend on the characteristics of the underlying domain.

The key control decision when adopting feasibility checking is the frequency of interaction. Tighter integration (i.e., more frequent feasibility checking) will lead to earlier recognition of infeasible planning decisions, but as the frequency of feasibility checking is increased, both the computational cost and the likelihood of redundant (i.e., non-informative) results also increases. The computational tradeoff can be improved if capacity analysis and resource allocation processes are *incremental* in that they reuse partial results from earlier invocations rather than starting over from scratch for each feasibility check.

2.3 Resource Apportionment

Complex planning tasks usually involve multiple objectives that impose conflicting demands on resource usage. In general, users will have preferences for how resources should be allocated among the tasks selected to achieve those objectives. These preferences might reflect the relative importance or priority placed on different objectives or tasks, as well as their expected consumption requirements. The term *resource apportionment* is used to describe this type of high-level partitioning of resources among tasks. Resource apportionment need not completely assign resources, but rather may simply impose constraints on the resources that can be used by various tasks. Resources might be apportioned at different levels: splitting resource pools among high-level objectives, designating particular resource types for specific tasks, or distributing resources among designated sets of tasks.

Resource apportionment can be seen as a strategy for budgeting sets of resources or resource levels to various portions of a plan. In doing so, resource interactions in the plan are localized, making it possible to consider independently the resource feasibility of various plan fragments. Resource apportionment can be incorporated in a straightforward manner into the integration models discussed above. It can be used, for example, to decompose and decouple the problem of checking the feasibility of various plan fragments associated with different high-level objectives.

Overall, the use of resource apportionment simplifies the resource allocation problem. However, given that it makes certain commitments for resource usage, it limits the range of obtainable solutions and may eliminate otherwise valid (and perhaps superior) solutions.

2.4 Scheduler-Driven Plan Modifications

Resource feasibility checking enables increased efficiency of plan generation through earlier detection of resource conflicts. The *a priori* apportioning of resources provides a complementary benefit, by localizing resource interactions in the developing plan and compartmentalizing resource allocation concerns. While improvements over the waterfall model, neither method exploits information gained during the scheduling process. Rather, the inability to schedule a given plan triggers a restart of the planning and scheduling process, with some internal mechanism of the planner responsible for appropriately redirecting the plan generation process.

Here, we consider models of interaction that are driven by the resource allocation concerns of the scheduler. Within these models, information about resource availability and sources of resource infeasibility is used to redirect the planning process. Although not a prerequisite, this form of interaction is greatly enhanced by an incremental replanning capability. In this circumstance, scheduler/planner interaction can proceed by identifying problematic or sub-optimal aspects of a plan from a resource allocation perspective and suggesting localized modifications to the plan that eliminate the source of these problems.

Problematic aspects of plans can be characterized in two ways.

- *Task-oriented* characterizations identify a set of tasks that cannot be scheduled with respect to current time and capacity constraints. When notified of such tasks by the scheduler, the planner would perform plan revisions to replace them in the current plan with alternatives. Information about additional (perhaps higher priority) tasks that are competing for the same resources could also be communicated by the scheduler, as a way to provide the planner with additional revision guidance.
- *Resource-oriented* characterizations identify those re-

sources/intervals that are oversubscribed. Given a list of such resources, the planner could seek to develop plans that reduce their usage over intervals where they are currently oversubscribed.

Scheduler-driven plan modifications of this type could also be used to support a form of plan/schedule optimization, with the scheduler requesting plan modifications that would improve the overall effectiveness of resource usage. For example, information about excess (or underutilized) resource capacity could provide useful guidance to the plan revision process (as a basis for prioritizing task replacement choices).

3 Continuous Planning and Scheduling

For most complex enterprises, planning and scheduling must proceed on a continuous, open-ended basis. New goals and requests require extensions to existing plans and schedules, while unexpected events may require repairs to previously planned and scheduled tasks, possibly including partially executed activities. Once execution has commenced, time for decision-making is often limited, making it essential that plan and schedule modifications are performed efficiently. Furthermore, *stability* of the plan/schedule is critical, thus requiring that modifications minimize changes. For these reasons, incremental methods for extending and repairing plans and schedules are essential.

Changes to high-level objectives and guidance will require modifications to the plan, which in turn will generally require modifications to the schedule produced for that plan. The planner should succinctly characterize plan changes for the scheduler. Thus, rather than having the scheduler restart from scratch on the modified plan, it should modify the schedule incrementally based on the change in task requirements derived from the plan revisions [13, 17].

Changes to resource availability (either capacity increases or decreases) may necessitate schedule revisions or suggest opportunities for more effective resource usage. Certain resource changes can be accommodated by revisions to the schedule alone. However, more significant changes in resource availability could bring the viability of previously planned tasks into question. In such situations, the planner will need to modify the plan, with the scheduler in turn making corresponding changes to accommodate the revised plan fragment. The decision of whether (and when) to abandon a search within scheduler space for alternative resource assignments and instead request plan changes constitutes an important control problem for planner/scheduler integration.

Unexpected events in the operating environment (e.g., the onset of bad weather) can also impact the viability of both plans and schedules. In general, planners and schedulers incorporate assumptions about both the initial world state, and how the world will change (both naturally, and as a result of executed activities). In the event that such assumptions are violated, plan and schedule revision may be necessary.

To ensure stability, plan and schedule revision processes must be sensitive to the impact that they have on overall activity. Ideally, changes should minimize disruption to those portions of the plan and schedule that are currently under execution, both to ensure continuity and to avoid potentially high costs of redirection. Characterizing what it means to be minimally disruptive is challenging within an integrated planner/scheduler because changes that the planner views as insignificant may prove to be highly disruptive for the scheduler (and vice versa). For example, the insertion of an additional task may be straightforward for the planner to accommodate, but could result in significant rearrangement of resource assignments that could otherwise have been avoided.

4 Integrated Planning/Scheduling Systems

4.1 Characteristics of Planning/Scheduling Problems

It is difficult to identify problem domains that do not require some level of integration of planning and scheduling capabilities. At the same time, differing domain and problem characteristics will dictate varying degrees of emphasis on each capability, and in many cases, one will tend to dominate the problem-solving process.

Consider, for example, the problem of *manufacturing management*, which involves assigning available resources to produce a set of products that satisfy a set of customer orders. For any given set of products to be produced, planning is required to transform high-level specifications into manufacturing process plans. However, assuming that this set of products is to be produced regularly to fill various customer orders over time, scheduling will be the critical determinant of manufacturing system performance (i.e., how well resources are allocated over time). This problem is *resource-driven*, in the sense that the principle concern is to optimize the use of resources to satisfy current demands for known products.

In contrast, problems such as travel planning, project management, or air operations planning have a more *goal-driven* flavor, in that they emphasize the development of resource-feasible strategies for satisfying a fixed set of high-level goals. While effective use of resources is important, the driving motivation is to identify and schedule actions that will ensure attainment of stated objectives. Accordingly, in such domains, planning capabilities will tend to play a more dominant role.

4.2 Survey of Planning/Scheduling Systems

There have been relatively few efforts to develop integrated planning/scheduling systems, with those that have been built motivated primarily by resource-driven applications. Here, we describe a set of systems that are representative of approaches pursued to date.

The early Hubble Space Telescope scheduling application of the HSTS system [6] serves as a good example of a resource-driven approach to integrating planning and scheduling. A set of observation requests for the telescope must be accommodated, each of which requires a number of actions for setup, observation, and clean-up. Within HSTS, an abstract scheduling model is used to optimize the sequence of observations to be taken, and a planner is used to work out the detailed network of activities for taking each picture. A central contribution of HSTS is a common representational framework that supports both planning and scheduling processes.

The DCAPS [10] and ASPEN [11] systems, like HSTS, provide integrated planning and scheduling environments for transforming high-level goals for spacecraft operations into low-level command sequences. A planning component transforms individual goals into sets of activities that are subsequently scheduled. DCAPS and ASPEN both employ an *iterative repair* model, in which repair methods are applied repeatedly to scheduled activities to improve schedule quality. In addition to effecting direct schedule modifications, repairs can also result in the addition or deletion of activities to be scheduled; in this sense, it can be considered to be a form of scheduler-driven plan modification. The application domains for these two systems share the characteristics that (a) the plans are relatively decoupled, and (b) domain constraints have a strong temporal character. As a result, they both focus on efficient resource allocation and scheduling.

The FORBIN system [2] is an early example of an integrated planner/scheduler. FORBIN adopts a more goal-oriented approach built on a model of tightly-coupled feasibility checking. The action representation language for planning within FORBIN supports a rich model of resource requirements and deadlines. For each planning operation, a temporal simulation is run to test schedulability of the plan that would result. Only planning steps that satisfy the schedulability constraints are considered. FORBIN provides one-time plan/schedule generation capabilities only (i.e., it does not support extension or repair of plans and schedules).

The planning and scheduling system described in [5] is built on an iterative waterfall model, but employs a unique form of feedback from the scheduler to the planner. Based on a probabilistic state model, the planner generates *control plans* that are designed to direct behavior of a device to prevent runtime transition to failure states. Plans are generated relative to a specified probability threshold on states, with higher thresholds resulting in consideration of fewer eventualities and hence simpler plans. Actions within these plans have associated hard deadlines that must be satisfied. The scheduler attempts to generate a periodic schedule for action execution that ensures satisfaction of associated deadlines (and hence avoidance of failure states). In the event that a satisfactory schedule cannot be produced, the scheduler provides feedback to the planner in the form of a recommended higher probability threshold, which results in plans that are easier to schedule. Similarly, when schedules are produced in which resources are underutilized, the scheduler suggests a lower probability threshold for the planner to enable the incorporation of additional activities. The planning/scheduling process then repeats for the new threshold.

IP3S, a blackboard-based system, integrates a generative process planner with a finite-capacity production scheduler [12]. The approach is designed to support a custom, make-to-order manufacturing facility, where approximately half of the received orders entail new products for which process plans do not exist. The system exploits feasibility checking during process plan development to obtain visibility into the current shop load and to generate process plans that avoid current resource bottlenecks. The system is also designed to support incremental management of plans/schedules as updated status information is received. For example, a machine failure resulting in unavoidable resource contention can trigger a replanning process, in which alternative machining processes and resources are explored.

5 JFACC Planner/Scheduler

We have recently embarked on a project to develop an integrated planning and scheduling system that supports generation of plans and schedules as well as adaptations in response to changing tasks, resource availability, or world conditions. The integrated system will accept information relating to feedback from execution assessment, evaluate impact on the current plan/schedule, interleave schedule and plan revision actions as necessary to resolve detected problems and exploit opportunities, and identify/prioritize factors for continued monitoring of the revised plan/schedule. This cycle will emphasize flexible, human-in-the-loop repair, allowing incorporation of robust types of user guidance, constraints and control over the scope, magnitude and types of change.

Our work is motivated by the problem of supporting a Joint Forces Air Component Commander (JFACC) in the development and execution of air campaigns. The characteristics of this domain are representative of many complex enterprises: tasks are complex and open-ended, resource limitations impact the strategies that can be employed for achieving tasks, and the operating environment is both dynamic and hostile. Successful air campaign operations require a mixture of strategy development and accompanying deployment of resources, with tight linkage between them.

The scheduling component for the integration effort is based on OZONE [14], a framework (or toolkit) for configuring reactive and mixed-initiative scheduling systems. The planning and execution monitoring capabilities are based on the Continuous Planning and Execution Framework (CPEF) [8].

5.1 OZONE

OZONE is based on a model of scheduling as an ongoing change process, and is designed explicitly to promote minimally (or selectively) disruptive, incremental scheduling capabilities. At its core is a customizable, constraint-based modeling framework and search architecture, based around three principal components [13]:

- constraint propagation - to incrementally update solution constraints and recognize inconsistencies as changes (extensions, additions, external updates) are made to the schedule,
- constraint analysis - to estimate the critical tradeoffs and opportunities for solution revision (or extension) implied by the current state of the schedule, and

- a set of heuristic scheduling procedures for carrying out specific schedule revisions (or extensions), providing differential optimization and/or conflict resolution capabilities.

The OZONE modeling and scheduling framework consolidates the results of application development experiences in a range of complex, dynamic scheduling domains. One recent application effort has used OZONE to produce a scheduling tool for continuous, day-to-day management of airlift and tanker missions at the USAF Air Mobility Command (AMC) [1]. This "Barrel Master" scheduler is being incorporated into Version 2.0 of AMC's Consolidated Air Mobility Planning System (CAMPS), which is scheduled for operational release in February, 2000.

More recently, we have developed a prototype system for dynamic allocation of munitions and aircraft to air campaign operations which will provide one starting point for our investigation into development of an integrated planner/scheduler.

5.2 CPEF

CPEF embodies a philosophy of plans as dynamic, open-ended artifacts that must evolve in response to an ever-changing environment. CPEF provides a range of operations required for continuous plan management, including *plan creation*, *plan execution*, *monitoring*, and *plan repair*. CPEF has been applied successfully to generate, execute, and repair complex plans for air operations within a simulated operating environment.

Plan creation within CPEF is based on a combination of SIPE–2 [16] and the Advisable Planner [7]. SIPE–2 is a generative planner based on the *hierarchical task network* model of planning [3]. The Advisable Planner provides an advice-taking layer on top of SIPE–2 that enables a user to guide and direct the plan generation so that solutions are customized to his or her individual preferences. Advice enables users to express preferences for strategies with certain characteristics, or that use or avoid specified entities (such as certain resources) in designated situations.

CPEF provides timely adaptation of planned activities based on monitoring of critical events within its operating environment. Conditions to monitor are extracted automatically through a causal analysis of generated plans. A centralized process manager determines when to perform modifications to plans based on monitored conditions and execution results. Plan repair operations are guaranteed to mimize changes to the original plan, and are grounded in the analysis of plan dependency structures [15, 4]. The system also supports *advice-based repair*, in which changes in advice lead to minimally disruptive modifications to the plan that reflect the advice changes.

5.3 Research Objectives

Our goal is to produce an integrated planner/scheduler that is highly responsive to the dynamics of the operating environment. This framework will support a range of core interactions between the planner and scheduler, including resource feasibility testing, resource apportionment, scheduler-based plan revisions, and execution-time plan and schedule repair.

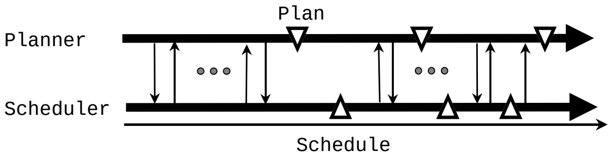

Figure 2: Mixed-initiative Model for Integrated Planning and Scheduling

Our work will involve investigation and development of several novel strategies for implementing those interactions. One such strategy involves the use of the advice-taking capabilities of the CPEF planner to effect schedule-based plan revisions. With this approach, the analysis of scheduling problems will produce a set of *repair advice* designed to eliminate the source of the scheduling problems from the current plan. For example, given an air campaign plan that is unschedulable due to a shortfall in available F-117s for neutralizing enemy air defences within a given sector, advice such as the following would be produced by the scheduler.

> *Avoid unnecessary use of F-117s for attacks on enemy air defences within Sector1.*

The CPEF planner would modify those portions of the plan that relate to the content of the repair advice produced by the scheduler. In this particular case, the planner would reconsider all decisions that resulted in the use of F-117s for the designated attacks, selecting alternative strategies to the extent possible to reduce reliance on the overconstrained resource.

In addition to building novel interaction methods between the planner and scheduler, we are interested in developing rich mixed-initiative models of control for the planner and scheduler that would enable each to make requests of the other to modify their problem-solving strategies, change their current solutions, or provide information about the space of possible changes (see Figure 2). We intend to use the resultant system as a framework in which to conduct experiments for evaluating different models and control strategies for planner/scheduler integration.

6 Summary

The ability to integrate planning and scheduling is critical for enterprise control in dynamic environments, where unpredictable changes necessitate rapid adaptations to both problem-solving strategies and resource assignments. This paper has outlined a series of models for the dynamic integration of planning and scheduling and provided a discussion of their relative strengths and weakness. Previous work on integrating planning and scheduling technologies was discussed and situated relative to these models. The paper also briefly described a project that seeks to provide tightly coupled planning and scheduling for a highly dynamic and continuous operating environment.

Acknowledgments

The work reported in this paper was supported by DARPA Contracts F30602-97-2-0066 (CMU) and F30602-97-C-0067 (SRI) under the supervision of Air Force Research Lab – Rome.

References

[1] M. Becker and S. F. Smith. Mixed-initiative resource management: The AMC barrel allocator. Technical report, Carnegie Mellon University Robotics Institute, 1999.

[2] T. Dean, J. Firby, and D. Miller. Hierarchical planning involving deadlines, travel time, and resources. *Computational Intelligence*, 4:381–398, 1988.

[3] K. Erol, J. Hendler, and D. S. Nau. Semantics for hierarchical task-network planning. Technical Report CS-TR-3239, Computer Science Department, University of Maryland, 1994.

[4] S. Kambhampati and J. Hendler. A validation-structure-based theory of plan modification and reuse. *Artificial Intelligence*, 55(2):192–258, 1992.

[5] C. B. McVey, E. M. Atkins, E. H. Durfee, and K. G. Shin. Development of iterative real-time scheduler to planner feedback. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1997.

[6] N. Muscettola, S. F. Smith, A. Cesta, and D. D'Aloisi. Coordinating space telescope operations in an integrated planning and scheduling framework. *IEEE Control Systems*, 12(1), 1992.

[7] K. L. Myers. Strategic advice for hierarchical planners. In L. C. Aiello, J. Doyle, and S. C. Shapiro, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifth International Conference (KR '96)*, pages 112–123. Morgan Kaufmann Publishers, 1996.

[8] K. L. Myers. Towards a framework for continuous planning and execution. In *Proceedings of the AAAI 1998 Fall Symposium on Distributed Continual Planning*, Menlo Park, CA, 1998. AAAI Press.

[9] K. L. Myers and T. J. Lee. Generating qualitatively different plans through metatheoretic biases. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*. AAAI Press, 1999.

[10] G. Rabideau, S. Chien, T. Mann, C. Eggemeyer, J. Willis, S. Siewert, and P. Stone. Interactive, repair-based planning and scheduling for shuttle payload operations. In *Proceedings of the IEEE Aerospace Conference*, 1997.

[11] G. Rabideau, R. Knight, S. Chien, A. Fukumaga, and A. Govindjee. Iterative repair planning for spacecraft operations using the ASPEN system. In *Proceedings of the International Symposium on Artificial Intelligence Robotics and Automation in Space (ISAIRAS)*, 1999.

[12] N. M. Sadeh, D. W. Hildum, T. Laliberty, S. Smith, J. McAnulty, and D. Kjenstad. An integrated process-planning/production-scheduling shell for agile manufacturing. In *Proceedings Fifth National Agility Conference*, Boston, MA, 1996.

[13] S. F. Smith. Reactive scheduling systems. In D. Brown and W. Scherer, editors, *Intelligent Scheduling Systems*. Kluwer Press, 1995.

[14] S. F. Smith, O. Lassila, and M. Becker. Configurable, mixed-initiative systems for planning and scheduling. In A. Tate, editor, *Advanced Planning Technology*. AAAI Press, Menlo Park, CA, 1996.

[15] D. E. Wilkins. Recovering from execution errors in SIPE. *Computational Intelligence*, 1(1):33–45, 1985.

[16] D. E. Wilkins. *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann, 1988.

[17] M. Zweben, B. Daun, E. Davis, and M. Deale. Scheduling and rescheduling with iterative repair. In M. Fox and M. Zweben, editors, *Intelligent Scheduling*. Morgan Kaufmann, 1994.