Planning-Based Integrated Decision Support Systems

Marie A. Bienkowski
Marie E. desJardins
SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

{bienk,marie}@erg.sri.com

Abstract

This paper describes a system that uses AI planning and representation techniques as the core of a decision support system.[1] The planning technology is supplemented with other AI and non-AI technologies. The overall system and its initial application domain, military operations planning, are described first. We then describe the integration of SIPE-2, a generative planning system, with three independently developed AI systems: a temporal reasoning engine, a case-based force selection module, and a system for scheduling and capacity analysis.

1 Introduction

In tackling a real-world problem with an AI solution, it is not uncommon to find that a single AI system fails to meet all the requirements for solving the problem. In some cases, existing software can be integrated (or must be used) with the AI solution; or the AI system must be altered to fit the problem (e.g., with a customized user interface); or new software must be added to round out the capabilities of the single system. In this paper, we report on a unique experience with supplementing a mature generative planning tool with several substantial pieces of technology: a case-based reasoner, a temporal reasoner, and a scheduling system. These technology integration experiments are among the first examples of harnessing well-developed software systems whose development paths were completely orthogonal to solve the same problem.

The goal of these integration experiments was to improve the quality of the plans generated by the System for Operations Crisis Action Planning (SOCAP). The first integration project used a temporal constraint propagation and inference system (Tachyon[2]) to improve SOCAP's temporal reasoning capabilities. These improved capabilities enable SOCAP to represent more sophisticated temporal constraints within plans, and to reason more accurately about the times and durations of actions and about resource utilization over time. The second integration project incorporated a case-based reasoning (CBR) system (CAFS)[3] to extend SOCAP's ability to choose objects to participate in operations. The CBR system identifies new objects by modifying ones that were used in the past for similar operations. In the third, ongoing project, SOCAP's planning capabilities are being integrated with Distributed Transportation Scheduling in OPIS (DITOPS), a scheduling system developed at Carnegie Mellon University (CMU).[4] SOCAP's ability to generate robust, feasible plans with realistic allocation of resources is being improved by the use of feedback from the scheduling system early in the planning and replanning phases.

These integration projects are unique in two ways: first, they utilize existing, independently developed AI-based modules to supplement an existing generative planning system; second, they add capabilities that are novel or relatively unexplored in generative planning systems.

The rest of this paper is organized as follows. Section 2 describes the generative planner and SOCAP, the application that motivated the implementation and testing of the integrated system. Section 3 describes the temporal reasoning module, Section 4 the case-based reasoning subsystem, and Section 5 the scheduling and capacity analysis module. Each of these sections includes a description of future work for each module. Section 6 presents other future work and conclusions.

2 Background

SOCAP was developed as part of an ongoing project to test the ability of AI planning systems to provide decision support for planning a course of action in response to a crisis [Bienkowski, Desimone, & desJardins 1993]. Human planners find it difficult to keep track of the myriad details involved in a large operation with multiple plans generated at multiple distributed planning sites. The dependencies among the actions in these plans, indispensable for providing plan justifications and for

---

[1] This work was performed under the ARPA/Rome Laboratory Planning Initiative, Rome Laboratory Contract Number F30602-91-C-0039, SRI International (SRI) Project 2062.

[2] Jonathan Stillman of the GE-CRD Artificial Intelligence Laboratory developed Tachyon and performed the integration on the temporal reasoning side.

[3] Lauren Halverson of the GE-CRD Artificial Intelligence Laboratory was responsible for developing and integrating CAFS on the case-based reasoning side.

[4] Stephen Smith and Ora Lassila of CMU integrated the DITOPS side.

analyzing hypothetical modifications to plans or to the world state, are also difficult to monitor. The knowledge representation underlying SOCAP keeps track of the detailed dependencies and constraints in the plan, while the human planner retains control over the planning and replanning processes. SOCAP is significant because, to date, no applied research activity has tested a full-scale generative planning system in an operational crisis-response environment.

SOCAP's core reasoning engine is SIPE-2,[5] a hierarchical, domain-independent, nonlinear planning system with a powerful formalism for representing domains and generating partially ordered plans (possibly containing conditionals) in those domains [Wilkins 1988; Wilkins 1992]. Its representation of constraints, resources, and causal rules is more efficient and expressive than that of other planners, and it provides capabilities for resolving parallel resource interactions. SIPE-2 enables a user to guide the planning process by choosing among different plan operators that satisfy a goal and among different values that satisfy the constraints on a variable (or to have these choices be made automatically). Users may also display the different hierarchical levels of the plan, set the level of interaction with the user during planning, and explore alternate plans concurrently.

SIPE-2 has implemented several extensions of previous planning systems, including the use of constraints for the partial description of objects, the incorporation of heuristics for reasoning about resources, and replanning techniques. One of the most powerful heuristics for reducing complexity in SIPE-2 is the ability to avoid frequent consistency checks by temporarily producing invalid plans. The system relies on *plan critics* that check for and correct problems in the invalid plans at certain intervals. These critics proved useful as a modular way to add extensions to SIPE-2.

SIPE-2 has been tested on a variety of small-scale problems for travel, robot, and aircraft planning, and for extended blocks-world problems. It has been applied to a larger-scale planning problem in the brewery domain [Wilkins 1990]. The performance of SIPE-2 in the domain described in this paper (military operations planning) is discussed by Wilkins & Desimone [1993].

SOCAP, in its military operations planning mode, encodes knowledge derived from a scenario used at a military teaching college [Bienkowski 1993]. Its customized user interface guides a planner through the decision-making needed for producing plans and displays the results graphically as a network of actions or as icons on a color map. SOCAP has generated employment plans for dealing with specific enemy courses of action (COAs) and deployment plans for getting the relevant combat forces, supporting forces, and their equipment and supplies to their destinations in time for the successful completion of their mission. Input to SOCAP includes threat assessments, terrain analysis, apportioned forces, mission goals, and operational constraints (see Figure 1). Unlike other systems that might support COA generation, SOCAP is highly interactive; it checks the user's choices for consistency and adherence to constraints; it represents the dependencies among actions in a COA; and it can reason about resource utilization and conflicts.

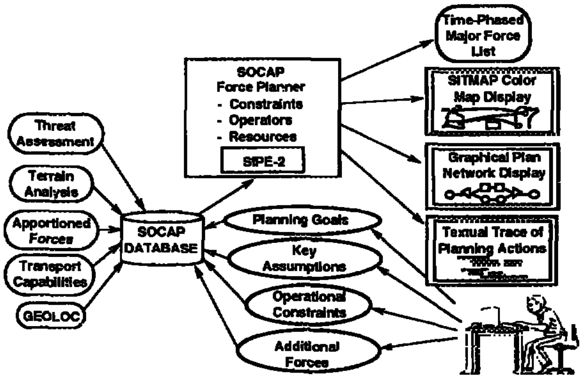

Figure 1. SOCAP Functional Overview

SOCAP is also being applied to two other domains: determining equipment configurations (type, quantity, and location) for response to oil spills for the U.S. Coast Guard; and generating assembly plans for manufacturing processes under an SRI Internal Research and Development Project.

SOCAP's intended users are concerned with potential heavy use of limited resources and thus require the ability to reason about resource utilization. In our development of SOCAP, we discovered that SIPE-2's existing mechanisms for reasoning about time were inadequate, that users required the ability to tailor forces, and that support for plan evaluation from different perspectives (e.g., logistical or transportation) was critical. However, SIPE-2's interactive style of planning and general architecture, as we show in the remainder of this paper, allowed other technologies to be integrated relatively easily to satisfy these requirements.

3 Temporal Reasoning

The addition of temporal reasoning to SIPE-2 addressed several of the problems identified during the development of the SOCAP application. Because of SIPE-2's limited temporal reasoning capability, SOCAP had previously been unable to reason about the utilization of resources and could not place temporal constraints between actions in the plans. Consequently, the plans generated did not represent certain important constraints that existed in the domain.

---

[5] SIPE-2 was developed by David E. Wilkins of SRI International's Artificial Intelligence Center.

In the previous version of SIPE-2, time was treated as a consumable resource that could be consumed but not produced, and whose consumption over parallel tasks was nonadditive. Each action specification could have associated ***start-time*** and ***duration*** slots containing variables with numerical constraints on them that would be satisfied by the planner.

SIPE-2 has several techniques for establishing the relative orderings of actions: inserting ordering links to avoid resource conflicts; using one action to meet several different requirements by ordering them after the action; and coordinating separate subplans by adding ordering links to goals that have been declared as external.

While the above capabilities enable SIPE-2 to solve many simple temporal problems, its inability to represent constraints relating the times of two possibly unordered actions remained a problem. Two SIPE-2 actions are either ordered with respect to each other, or they are unordered. If the latter is the case, the planner considers it possible to order them either way or to execute them simultaneously. Information about start times and durations is used only when it can be deduced that two actions should be ordered on the basis of this information. This limitation, for example, prevents modeling when the various effects of an action become true during its execution; nor can actions be modeled that *must* occur simultaneously. We felt that Allen's 13 temporal relations [Allen 1983] would permit more versatile operations, with explicit representation of actions starting or finishing at the same time, overlapping each other, or one occurring during another. Many dependencies between different military actions should be represented in this way in SOCAP. For instance, cargo off-load teams should arrive at an airport or seaport at the same time as the first air or sea transport.

The temporal constraints required for our application can be broken down into two sets: domain-dependent constraints (for example, required delivery dates and constraints between actions) and domain-independent or "common-sense" constraints (for example, that durations cannot be negative). The former constraints are described in the SIPE-2 plan operators, and are copied into the plan as goals are expanded via the operators. A plan critic then calls a temporal reasoning module that propagates these constraints and combines them with the "common-sense" constraints that it represents internally, returning an updated set of time windows on the plan nodes.

We initially extended SOCAP's ability to represent and reason about time by adding a layer on top of SIPE-2 that would keep track of the temporal constraints within the plan, using the Tachyon temporal reasoning system [Allen & Stillman 1992] to maintain and propagate these constraints. The interface to Tachyon was designed to be general enough to permit a different temporal reasoner to be substituted (such as TMM [Schrag, Carciofini, & Boddy 1992]). We are now beginning to integrate the temporal reasoner more closely with SIPE-2, as described below.

We developed a straightforward approach for the representation of temporal constraints between plan nodes, and determined that the requirements for an external temporal reasoning system would be the ability to represent both time windows on plan nodes and internode constraints, and to propagate these constraints to yield tighter time windows on the nodes. Tachyon, an efficient implementation of a constraint-based model for representing and maintaining qualitative and quantitative temporal information, has these capabilities.

A new plan critic was written for SIPE-2, to be run at the end of each planning level. This critic extracts all temporal information (time windows and internode constraints) from the plan, sends it to Tachyon, and stores the updated time windows returned by Tachyon in the plan. In addition, we implemented methods to maintain the constraints in the plan as goals are expanded to the next planning level in SOCAP. Temporal constraints between goals at one level are represented as multiple constraints between the actions and subgoals in the corresponding subplans in the expansion.

We extended SIPE-2's operator syntax to enable a designer to specify any of the 13 Allen relations or quantitative constraints (the permissible range of metric distances) between the end points of any pair of nodes in an operator. Part of a SOCAP operator description using the new syntax is shown in Figure 2. This operator divides a military unit into air and sea cargo, and moves the components from their origin to a destination airport. The temporal constraints in this example are that the goals of securing the areas around the destinations must overlap with the movement of the troops to these destinations, and that the air cargo must arrive at the final destination before the sea cargo arrives (via ground from `seaport2`).

The `constraints` slot shown in Figure 2 is used to specify the temporal relations between the actions in the operator. Nodes are described by the name of any action or predicate name of any goal in the operator plot. Multiple actions and goals with the same name are disambiguated by the specification of which appearance of the name to use (e.g., `ground-moved.1` refers to the first ground movement action in the plot).

The extended system found temporal inconsistencies that could be resolved only by changing the available dates of military units, or by reassigning units; previously, these inconsistencies were not resolved. Thus, the system now encodes a better model of the domain. The temporal information is especially important for integration with scheduling software (discussed in Section 5), because this information enables SOCAP to pass a more complete set of constraints to the scheduler. In addition, the temporal information supports better plan evaluation

```
CONSTRAINTS: (overlaps area-secured.1 move-by-airlift)
             (overlaps area-secured.2 move-by-sealift)
             (later (end ground-moved.3) (end move-by-airlift));
PLOT: PARALLEL
    BRANCH 1:  GOAL: (ground-moved air-cargo origin airport1);
               PROCESS ACTION move-by-airlift;
                       ARGUMENTS: airport1, airport2;
    BRANCH 2:  GOAL: (ground-moved sea-cargo origin seaport1);
    BRANCH 3:  GOAL: (area-secured airport2);
    BRANCH 4:  GOAL: (area-secured seaport2);
END PLOT END OPERATOR
```

Figure 2. Move-by-Airlift-and-Sealift Operator

and capacity analysis during planning, and is essential for improved interaction with map-based displays and for exploring plan execution and replanning.

The current implementation of temporal reasoning is only the first step toward developing a complete temporal reasoning capability in SOCAP. We plan to integrate temporal reasoning more tightly into SIPE-2, so that we can use the temporal information when choosing operators and exploring alternative plans, and can extend the truth criterion and all critics to use temporal information. Some of the research issues raised by closer integration include the need to find appropriate tradeoffs between completeness and/or correctness, and efficiency; the development of efficient heuristics for reasoning about the temporal constraints in the truth criterion; and the need to determine how often it is necessary or wise to compute the full set of implications of the temporal constraints in the plan. Since one of the strong points of SIPE-2 is that it utilizes very efficient heuristics for determining when two actions possibly or necessarily interfere with each other when only simple ordering links are considered, the development of such heuristics in the presence of temporal constraints is an important concern.

We have found that even the relatively limited temporal reasoning capabilities we have added to SIPE-2 provide a significant source of power in representing important domain constraints, and improve the generated plans substantially. Some of the difficulties in incorporating complete temporal reasoning capabilities into planning systems are discussed in [Allen 1991].

4 Case-Based Reasoning

Selecting the right force to participate in a military operation, or tailoring a force to meet special requirements, is an important part of operations planning. Considerations include a unit's potential to deter or defend against an enemy threat, its mobilization, its ability to handle the terrain, and its time to deploy. Initially, SOCAP prompted the user to select a unit from a list of available units that met the constraints of the operator being applied. The user could see what constraints were met by the units in the list, but had to rely on personal preference to make the selection. Users also expressed a desire to be able to modify the units in the list (e.g., its equipment list).

This force selection and tailoring was seen as an area where case-based reasoning [Kolodner 1993] would apply. We integrated a CBR system from GE-CRD called CAse-based Force Selection (CAFS) and built on GE's CAse-based REasoning Tool (CARET). This integration does not quite fit under the heading of case-based planning as the phrase is normally used, since the cases being extracted are not plans, but bindings for plan variables.

We modified SIPE-2 to call the CAFS module for major force selection instead of presenting a list to the user for selection.[6] CAFS itself was modified to handle SOCAP objects and operators. CAFS uses a force-selection case library to retrieve and return either a ranked list of cases or the best matching case for SOCAP to use. The match is computed by using mission requirement and force capability as indices to choose the closest matching force. If the closest matching force does not fit SOCAP's requirements, heuristics are used to modify the force in the appropriate manner. For example, a unit similar to the retrieved case, with additional support units, might be generated and returned as an appropriate unit.

Various approaches were discussed for presenting information about the cases in SOCAP. In the end, it was decided that the most effective approach would be to show the CAFS display running next to the SOCAP display. This approach avoids duplicating the graphical user interface of CAFS within SOCAP, and provides the user with multiple views of the force selection process.

SOCAP is initialized with a set of forces that are assigned for the planner to use. However, the modified units returned by CAFS may not be from this set (they may be notional, or abstract, rather than actual existing units); therefore, SOCAP must define them as new objects, attaching the necessary attributes to them. Because notional units are incompletely specified, we

---

[6] SIPE-2 can also call Gister, an evidential reasoning system, to select the best-rated unit. However, CAFS provides the important feature of allowing the user to modify the unit.

had to supplement the retrieved cases on the SOCAP side with default values for attributes such as location and available-to-load date in its further planning.

Another data mismatch concerns the semantics of constraints in CAFS as compared to those in SIPE-2. Because CAFS is capable of relaxing some of the constraints associated with a choice, the chosen unit may violate constraints represented in SOCAP. Because SIPE-2 does not handle relaxation of constraints, the violated constraints are removed from the plan so that the truth criterion does not detect an inconsistency. A preferable solution would be to extend SIPE-2 to allow the representation and relaxation of soft constraints; this is planned as future work.

5 Scheduling

An important feature of the type of decision support that SOCAP is intended to provide is the evaluation of the feasibility of plans (usually by external modules), followed by feedback into the planner, which then modifies the plan so that it better meets the evaluation criteria. Assessing the transportation feasibility (essentially the use of resources over time) of a plan was a focus for the users in the SOCAP military domain. To take full advantage of systems that are optimized for scheduling (unlike SIPE-2), we are investigating the integration of SIPE-2 with CMU's DITOPS.[7] Our model for this interaction is that SOCAP will utilize DITOPS at various stages of its search to assess the feasibility of the developing plan from the standpoint of capacity requirements; to recognize bottlenecks; and to detect projected resource conflicts under a variety of assumptions. This work will occur in three phases, as described below; we have completed Phase 1.

In Phase 1, we examined the utility of performing a capacity analysis during plan generation, to let the user check for potential overuse of resources before they are committed for use in a final plan. This early analysis aids in the assignment of resources to operations, based on projections of resource bottlenecks: e.g., either SOCAP or a user can use the analysis results to choose feasible deployment destinations for major forces during initial plan generation, or to reassign transportation resources. Our focus for Phase 1 was on the analysis of port capacities and sealift/airlift usage.

To simplify the integration, we isolated the capacity analysis module from DITOPS and called it as a subroutine from SOCAP. SOCAP and DITOPS have their own domain models of resources (including capacities and constraints on their utilization), locations, and distances. Although the domain models are resident in each system, they contain the same information (thus paving the way for a shared domain model accessed from a knowledge server). To focus DITOPS' attention on the relevant parts of the plan, SOCAP extracts a network of transport operations including sequencing constraints and the non-resource-using nodes that contribute temporal constraint information. For each operation, SOCAP includes its time bounds (earliest and latest start times and earliest and latest end times); the resources and amount of capacity it requires; and its duration (the resource is assumed to be used for the entire duration). Alterations to the resources (e.g., a lift being unavailable for a specified time period) can also be included. This information is analyzed by DITOPS' capacity analysis routines and the result is passed back to SOCAP in the form of absolute values for supply and demand of aggregate resources. This data is presented on a color graph showing the demand for a resource vs. capacity over time. A horizontal line indicates the maximum capacity, and points above the line indicate overuse. Users can then, by directly manipulating a bar chart showing the utilization of each resource over time, reassign resources or change the time of an operation.

In Phase 2, we will concentrate on automating the feedback from the scheduler: SOCAP will send a complete plan to DITOPS, which will then provide feedback to SOCAP on resource conflicts that it has found as a result of its analysis. Our plan for this phase is to demonstrate feedback from DITOPS to SOCAP only when minor perturbations of the plan or schedule occur—for example, minor changes in the composition or location of a threat, and changes in the availability of transportation assets. We will also explore issues concerning when to reschedule or when to replan, which rescheduling and replanning strategies to use, and the tradeoffs associated with replanning versus rescheduling.

In Phase 3, we will investigate more closely how guidance to scheduling can be provided by (1) the dependency structure of the plan, (2) the choices made during plan generation, and (3) the alternative choices that are recorded in the plan state. We will demonstrate feedback between SOCAP and DITOPS when major changes to the plan or schedule occur, such as additional missions, and when major resource contention exists for combat forces and transportation assets.

6 Conclusions and Future Work

The work reported in this paper is novel in integrating a planning system with other, compatible AI technologies. As a result of this work, we have identified key research issues, such as the representation and use of temporal information and scheduler feedback in a generative planner; application issues, such as ensuring that the same domain knowledge was understood by all modules; and system development issues, such as extending the heuristic ordering critics in SIPE-2.

We found that our integration efforts were simplified because the other systems that we used could be called as subroutines by SOCAP. This approach contrasts with an integration approach in which each module is viewed as

---

[7] DITOPS is based on the Opportunistic Intelligent Scheduler (OPIS) scheduling technology developed at CMU [Smith 1987].

a separate agent with independent control, where the communication of results and the negotiation of tasking is critical. The consequence of our approach was that there was no need for each module to retain state between calls; instead, the necessary information was sent each time an external module was invoked.

We were able to coordinate all of the necessary integration by e-mail and telephone conversations. This method works well for planning integration, especially designing an interface and working out differences in representations and assumptions. The largest limitation this method imposed was on getting the software running: tracking down and fixing bugs is slow. We expect that more face-to-face interaction would be needed under stricter time constraints or for tighter integration.

Integration was facilitated by systems that had a large number of options for execution: for example, the DITOPS module could return the contention intervals only, the daily supply versus demand ratios, or absolute values for demand and supply. While such variability required more discussion to enable the integrators to agree upon the desired input and output, the built-in flexibility required fewer changes to the code on both sides.

Formally specified integration experiments such as those we describe here give technology developers a chance to enhance the functionality of their systems to meet the requirements of other systems written by developers who have no preconceived notions of what complementary technology should do, only their own requirements for the processing or output they need. Success in these experiments paves the way for more tightly coupled processing (as in the case of the temporal reasoner and planner) or integration using meta-architectures (such as blackboards) to enable AI systems to tackle a larger part of more complex problems.

This work has also paved the way for a more structured integration of SOCAP, CAFS, Tachyon, and DITOPS using (1) a common knowledge representation language that provides an interlingua for the different systems; and (2) a client/server interface mechanism that supports location-transparent interprocess communication. This common language and interface ensure that modules that adhere to the standards can be substituted for other, comparable, modules, and facilitates the evaluation of the modules in an integration testbed.

Our future work includes a re-engineering of SIPE-2, to facilitate future integration work with other tools. In particular, we see the need for plan simulation and plan evaluation tools to test characteristics of plans such as resource utilization, logistics feasibility, and robustness. We also envision the use of other planning modules (such as case-based planners) to contribute more substantial parts of a plan than the variable instantiations produced in the current integration effort. Such an effort would require, at a minimum, exploiting SIPE-2's capability of integrating plan fragments.

Finally, we have recently begun a project that will utilize inductive learning techniques to refine and verify domain-specific planning knowledge based on feedback from simulators and on expert decisions during planning. These inductive learning techniques, along with the associated graphical knowledge editor under development, will support non-AI-expert users in developing domain knowledge.

7 References

Allen, J. F. 1991. Temporal Reasoning and Planning. In Reasoning About Plans. San Mateo, CA: Morgan Kaufmann Publishers, Inc. Allen, J. F., Kautz, H.A., Pelavin, R.N., and Tennenberg, J.D., eds.

Allen, J.F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11): 832-843.

Arthur, R., and Stillman, J. 1992. Tachyon: A model and environment for temporal reaso*ning*, Technical Report. GE Corporate Research and Development Center.

Bienkowski, M.A. 1992. Initial Requirements Analysis for Automation in Support of Joint Military Planning, Technical Report, ITAD-2062-TR-92-40. SRI International.

Bienkowski, M.A, Desimone, R.V., and desJardins, M.E. 1993. Decision Support for Transportation Planning in Joint COA Development, Annual Report, ITAD-2062-AR-93-101. SRI International.

Kolodner, J. 1993. *Case-Based Reasoning*. San Mateo, California: Morgan Kaufmann Publishers, Inc.

Schrag, B., Carciofini, J., and Boddy, M. 1992. ∀β-tmm manual (version b18bs), Technical Report, CS-R92-012. Honeywell Systems & Research Center.

Smith, S.F. 1987. A constraint-based framework for reactive management of factory schedules. In Proceedings of the International Conference on Expert Systems and the Leading Edge in Production Planning and Control, Charleston, SC: Oliff, M., ed.

Wilkins, D.E. 1992. *Using the SIPE Planning System: A Manual.*Artificial Intelligence Center, SRI International.

Wilkins, D.E. 1990. Can AI planners solve practical problems? *Computational Intelligence* 6(4): 232-246.

Wilkins, D.E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm*. San Mateo, CA (August): Morgan Kaufmann Publishers, Inc.

Wilkins, D.E., and R.V. Desimone. 1993. Applying an AI Planner to Military Operations Planning. In *Intelligent Scheduling*. San Mateo, CA: Morgan Kaufmann Publishers, Inc. Fox, M.and Zweben, M., eds.