SOCAP: Lessons learned in applying SIPE-2 to the military operations crisis action planning domain

Roberto Desimone
SRI International
333 Ravenswood Avenue (EK335)
Menlo Park, CA 94025
roberto@erg.sri.com

Abstract

This report describes work funded under the DARPA Planning and Scheduling Initiative that led to the development of SOCAP (System for Operations Crisis Action Planning). In particular, it describes lessons learned in applying SIPE-2, the underlying AI planning technology within SOCAP, to the domain of military operations deliberate and crisis action planning. SOCAP was demonstrated at the U.S. Central Command and at the Pentagon in early 1992. A more detailed report about the lessons learned is currently being prepared [7].

This report was presented during one of the panel discussions on "The Relevance of Scheduling to AI Planning Systems".

Introduction

Many agencies, in addition to the military, have the need to manage crises. Good crisis management is characterized by quick response, decisive action, and flexibility to adapt to the changing situation. Developing a good course of action (COA) and modifying it as necessary must take into account a number of factors: approaches used in past cases that have worked well, novel features of the new situation, differing priorities for subparts of the crisis, and feasibility of suggested COAs. The objective of this program of applied research was to develop decision aids to enable more flexible and accurate joint military COAs to be developed in response to a crisis. To date, no research or development activity has integrated a full-blown generative planning system into an operational environment.

SOCAP (System for Operations Crisis Action Planning) embodies SIPE-2, together with a user interface tailored to military operations and a situation map display system. SIPE-2 (System for Interactive Planning and Execution) is a domain-independent, AI planning system that was developed during the 1980s by David Wilkins of SRI International's Artificial Intelligence Center [4, 5, 6]. It supports both automatic and interactive generation of hierarchical, partially-ordered plans. This system provides efficient methods for representing properties of objects that do not change over time, and uses these to constrain the choice of objects associated with actions in the plans generated. SIPE-2 has been tested out on a variety of small-scale problems for travel, robot, and aircraft planning, and for extended blocks-world problems. More recently it has been applied to a larger scale planning problem in the brewery domain.

In early 1992, SOCAP was demonstrated both at the U.S. Central Command in Tampa, Florida and at the Pentagon. The aim was to demonstrate the feasibility of applying the SIPE-2 technology within SOCAP for the generation of large-scale military operations plans (OPLANs). The overall objective is to generate several OPLANs that describe employment plans for dealing with specific enemy COAs, and identify 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. [3] provides a description of the some of the requirements for automating the joint military operations planning process.

The rest of this report will describe SOCAP and the lessons learned in applying SIPE-2 to the military operations crisis action planning problem.

SOCAP – System for Operations Crisis Action Planning

Figure 1 shows the SOCAP architecture, highlighting the necessary inputs for the generation of OPLANs, the available outputs, and the user interaction. It is assumed that the following inputs would be fed into the SOCAP database from available military databases:

- threat assessment – list of enemy threats, locations and dates.
- terrain analysis – information on terrain features that might affect mobility and observability.

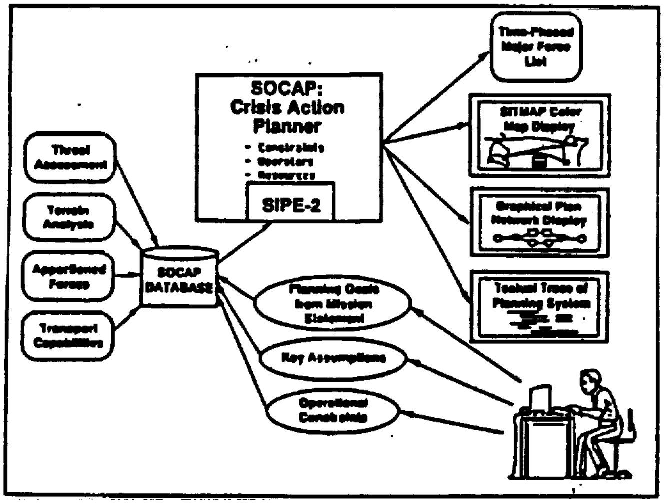

Figure 1: SOCAP Architecture

- apportioned forces – list of combat forces available for planning purposes.
- transport capabilities – list of available assets.

Other inputs would come from the user:

- planning goals – list of goals that match mission statement.
- key assumptions – e.g. rules of engagement, non-intervention of third party forces.
- operational constraints – e.g. overflight privileges, troop limits in country.

In this case, a typical user would be either the mission commander or one of his/her joint staff.

Most of the above information is inherently dynamic and is best represented in SIPE-2 as simple first-order predicates. However, a great deal of the available data are static, and for efficiency reasons are best represented in SIPE-2 using its hierarchy of classes and objects, together with (static) properties of objects. For example, cargo requirements, and combat capabilities for specific combat forces should be denoted as (static) properties of these forces.

SOCAP also requires a large set of plan operators to describe military operations that can achieve specific employment or deployment goals. For instance, there are a variety of military operations for deterring an enemy army, navy or air force. Each of these operations may be represented by a different plan operator which all have the common effect of deterring an enemy force. However, they may have different sets of preconditions that need to be satisfied before they can be brought into the plan, or different resource requirements.

The SOCAP user interface provides facilities for guiding the user through the plan generation process. The amount of user interaction can be varied during the planning process. It can range from being fully automated, in which case SOCAP generates a plan with no human interaction; to semi-automated, in which the user makes some choices; to fully manual, where the user makes all the choices. At each goal in the plan, the user can request the possible operators that achieve the goal to be displayed. Likewise, when attempting to bind a variable associated with an argument of an operator, the possible bindings can be displayed. For instance, the user may be presented with the set of military units that have the appropriate capabilities to deter an enemy threat, or a list of suitable locations for the military operation. This set may be constrained by the preconditions and other constraints associated with the arguments of the relevant plan operator. At the end of each plan level, the plan is checked for logical consistency, and then progresses to the next level until there are no more goals to be satisfied or actions to be decomposed further.

The plan may be displayed at each plan level, either as a partially-ordered network of actions and goals, or graphically on a time-based map display. The map display shows the actions that are occurring on different days during the mission. The temporal information for the map display is derived from durations associated with each action and from the dates when the enemy threats should be deterred or countered.

The following gives an idea of the size and complexity of the problems we are dealing with and the knowledge base within SOCAP. The size of plans we have generated have about 100-200 actions in the final plan level. The SOCAP knowledge base comprises: 200-250 classes/objects, 15-20 properties per object, around 1200 predicates, and 50-100 plan operators.

Lessons Learned

The lessons learned from applying SIPE-2 to the military crisis action planning domain can be divided into three main sections: successes and difficulties in applying the existing SIPE-2 technology, and open research issues.

Successes

The hierarchical plan decomposition process embodied within SIPE-2 maps well onto the military operations planning process, and delays the detail until the appropriate planning level. As a result, it was relatively easily to group sets of plan operators according to the various phases/levels of the operations planning process. For the purposes of the demonstration, these were:

Level 1: Select mission type.

Level 2: Identify threats and their locations.

Level 3: Select employment operations, major forces, and deployment destinations.

Level 4: Add deployment actions.

The class/object hierarchy provides a clear representation of static information within SOCAP, and also aids validation. A simple constraint language permits the properties associated with classes and objects to be posted on the arguments of operators. Thus, variable binding can be delayed until the constraints point to a single instance. It is also possible to force instantiations of these variables with user guidance. For instance, this facility might be used to force the selection of a favored military unit for a specific operation.

SIPE-2 provides a mechanism for permitting domain-specific knowledge to determine the number of iterations of an operator. For instance, in order to determine the number of enemy threats to deter or counter, SOCAP checks the number of enemy threat units identified in the threat assessment database, and generates a sub-goal for each. SOCAP has a variety of iterative operators that search for different types of enemy threats.

SIPE-2 permits a great deal of information to be presented to the user at a variety of levels of detail. The SOCAP user interface extracts the appropriate details and presents them to the user during the planning process. Thus, when a user is viewing the possible choices of military units for an operation, SOCAP presents the constraints that led to these choices. Nodes that contain certain predicates or arguments may be highlighted on the graphical display. Predecessors, successors and nodes in parallel may also be highlighted. This is especially useful when the plan display is large and convoluted.

The time-based map display provides another means of displaying the plan that is particularly appealing to military planners. It is possible to show the operations that occur on each day of the mission and display appropriate information about the type of military operation, the units involved and the boundary of the operation.

Difficulties

Although SIPE-2 does have capabilities for resource reasoning, specifically the representation of reuseable and consumable resources, we were unable to make use of them effectively, because of the lack of temporal reasoning within SIPE-2. Time windows associated with each action involved in a resource conflict would provide information that would help to resolve the conflict. Temporal information on the availability of the resource would permit simple conflict resolution without resorting to scheduling.

Continuing with the temporal reasoning issue, we found it would have been very useful to have had Allen's 13 temporal relations [1, 2]. This would have permitted more versatile operations including actions starting or finishing at the same time, overlapping each other, or one occuring during another, as opposed to just one strictly before another. There are many examples of dependencies between different miltary actions that could have been represented, if only...

Although SIPE-2 does have a mechanism for representing shareable resources between actions in parallel, it is very inflexible, in that you have to determine in advance how such resources might be shared over several actions. For instance, a large military unit, such as a division, may be employed in several operations simultaneously, where each operation uses some of the division's capabilities. The number of operations over which the division may be shared depends on the amount of resource required for each operation. Thus, the only way to reason about the shared resource is to consider the capabilities of the division as a consumable resource purely for this specific set of operations.

We would have liked to have had a flexible procedure for preferring to associate specific resources with actions. For instance, when choosing military units for operations, in order to minimize the number of troops involved in the operation, it is often wise to choose units already involved in the plan, provided they have not been overutilised. It is possible to write such heuristics in SIPE-2, but these are fairly rigid, and a trade-off between several heuristics is really what is required.

Another capability we would have liked is the ability to combine sub-goals at will, or serendipitously. For instance, at present, for every enemy threat identified, a friendly unit is identified to deter or counter it. If several small enemy forces are located close to each other, SOCAP attempts to deal with each threat individually, rather than considering them as an aggregate threat that might be countered with a single larger friendly force. Whether the aggregation was done by the user or by some conceptual clustering algorithm, it is important that the original sub-goals are replaced by a new sub-goal. One could write a large set of plan operators that attempt different ways of clustering sub-goals, but this is not practical for large problems.

Currently, it is difficult to represent the notion of a task force whose composition is determined by whichever military units were assigned to lower level actions. It is possible to represent a class of objects of type, task force, and make use of a part-of predicate to relate specific military units to a specific task force, but this is not an easy procedure.

We could have made greater use of deductive rules within SIPE-2 to highlight dependencies between parts of the plan that involve long chains of deduction. For instance, the arrival of communications equipment could have triggered deductive rules to fire that would have eventually, after several rules, pointed to the availability of the necessary command and control facilities for another operation.

It would have been very helpful to have had feedback from a "tame" combat simulator. Such feedback could have been used to guide the choice of operations, forces, locations and times. It could also have been used to compare the effectiveness of a variety of courses of actions and to provide appropriate metrics for identifying qualitatively different COAs.

Another problem involves SIPE-2's meta-level control of the goal achievement process. Unfortunately, this process can only be done by having additional operators that copy their goals down to the next level when certain preconditions are true. For instance, one may decide to achieve all employment goals first and only start on the deployment goals when the employment goals have been satisfied. This notion of encapsulating such meta-level heuristics for goal achievement in the preconditions is very rigid. Ideally, one would want a more flexible process that permits a trade-off between several heuristics.

As you can gather, we managed to deal with some of the above difficulties with less than acceptable solutions. In most cases these solutions were very rigid and might even work well for some problems, but certainly would not be flexible enough for a variety of situations.

Open Research Issues

We were continually asked by most military operations planners to whom we showed SOCAP about support facilities for updating and writing new operators. We explained that this would involve providing extensive facilities for making sure that the preconditions and effects were syntactically and semantically correct. It would also require flexible test algorithms to ensure that the revised or new operators did not adversely affect other existing operators. This may provide an excellent domain for machine learning techniques.

There are a whole set of research issues concerning the relationship between and integration of planning and scheduling techniques. Below, I have just listed a few questions below that ought to be addressed:

- How can information from plan structure guide constraint relaxation?
- When to stop plan generation and choose to generate schedule?
- When to repair schedule versus plan repair?
- When to project/simulate the plan/schedule?

Summary and Conclusions

The SOCAP work discussed in this report provides the first steps towards an operational prototype that will eventually be tested out on real military crises. So far, it has been tested on a single scenario developed at the Armed Forces Staff College. We will be extending the system significantly over the next few years, and will test it on a variety of different scenarios. You should expect a steady stream of progress reports!

Acknowledgements

This work was funded within the DARPA/Rome Laboratory Planning Initiative under contract number F30602-91-C-0039.

This report could not have been written without the tireless efforts of the SOCAP team headed by Marie Bienkowski, and comprising Marie desJardins, Roberto Desimone, Jeff DeCurtins, Kate Finn, and Robert Hicks. We are also grateful for support received from David Wilkins, Peter Karp and John Lowrance.

References

[1] J. Allen. Maintaining Knowledge about Temporal Intervals. Technical Report TR-86, University of Rochester, 1981.

[2] J. Allen and J. Koomen. Planning using a Temporal World Model. In *Proceedings of the Eighth IJCAI.* International Joint Conference on Artificial Intelligence, 1983.

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

[4] D.E. Wilkins. Domain Independent Planning: Representation and Plan Generation. *Artificial Intelligence*, 22, 1984.

[5] D.E. Wilkins. Recovering from execution errors in SIPE. Technical Note 346, SRI International, 1985.

[6] D.E. Wilkins. Hierarchical planning: Definition and implementation. In *Proceedings of ECAI-86.* European Conference on Artifiical Intelligence, 1986.

[7] D.E. Wilkins and R.V. Desimone. SOCAP: Lessons learned in applying SIPE-2 to the military operations crisis action planning domain. In preparation, 1992.