Why Real-World Planning is Difficult: A Tale of Two Applications

Steve Chien, Randall Hill, Jr., XueMei Wang, Tara Estlin,
Kristina Fayyad, and Helen Mortenson
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109-8099

Abstract. In this paper we describe a number of obstacles hampering the application of planning technology to real-world problems, as encountered in two real-world planning projects at JPL: MVP - a planning system for automated generation of image processing procedures; and LMCOA - an intelligent system for assistance in antenna operations. First, we describe how existing planning representation must be enhanced to represent and reason about aspects of plans besides goal achievement - resource usage, quality, execution time, flexibility, and generality. Second, planning systems must be able to fit into a wide range of operational contexts - most planning tasks cannot be completely automated, therefore at a minimum the plans produced must be easily understandable and modifiable by the users. In some cases the user must be intimately involved in the plan construction process itself. Third, planning systems must be able to compare favorably in terms of software lifecycle costs to other means of automation such as scripts or rule-based expert systems. This means that development of intelligent tools and environments to facilitate knowledge acquisition, validation, and maintenance are of prime importance. We hope that our description and elucidation of these issues will lead to increased work in these areas.

1 Introduction

Why have so few actual planning applications been fielded? In this paper we describe three types of issues hindering such efforts - lessons learned from two fielded planning applications: an automated image processing system (called MVP - for Multimission VICAR Planner) and a decision support system for antenna operations (called LMCOA - for Link Monitor and Control Operator Assistant). We hope that our description of these issues will encourage research in these areas of great importance to fielding real-world planning systems. We categorize these planning issues into three general classes. The first set of issues relates to expressiveness of representations for planning knowledge (such as more expressive action and temporal representations). Within this issue, we

This paper appears in New Directions in AI Planning, M. Ghallab and A. Milani, ed., IOS Press, Washington, DC, 1996, pp. 287-298.

This paper describes work performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

Other past and present members of the MVP team are Christine Ying, Shouyi Hsiao, Darren Mutz, Alex Gray, Joe Nieten, and Jean Lorre. Other past and present members of the LMCOA team include Crista Smyth, Trish Santos, Kathy Sturdevant, and Lorrine Lee.

particularly highlight the importance of representing and reasoning about aspects of a plan other than goal achievement. These measures can be broadly thought of as plan quality, but include evaluation of aspects of the plan relating to: how well it achieves the goal (degree of goal satisfaction), resource utilization, flexibility, robustness, and generality. The second issue is that a planning system must fit into an overall operational context. Because most problems cannot be fully automated, there is a significant need for a natural mode of interaction between the user and the system - a clear and convenient division of labor and control between the user and the system. This means that most likely the plans produced by a planning system must be easily understandable and modifiable by the user. While most planning systems produce a plan with a dependency structure at the tactical level (in terms of causal supports, protections, etc.), explanations at the strategic level are often required. In some cases, the user may need to be intimately involved in aspects of the plan construction process. Third, a critical factor in determining the feasibility of automating a planning application is the comparison of software lifecycle costs compared to other methods of automation - such as scripts or rule-based systems. Thus, development of intelligent tools to assist in knowledge acquisition, verification, and maintenance are of prime importance. The rest of this paper is organized as follows. Section 2 provides background information on the MVP and LMCOA applications. Section 3 describes representational difficulties encountered in the MVP and LMCOA applications. Section 4 describes the difficulties of integrating a system into the MVP and LMCOA operational contexts. Section 5 outlines some of the issues relating to knowledge acquisition, knowledge verification, and knowledge maintenance relevant to the MVP and LMCOA applications.

2 The MVP and LMCOA Applications

We begin by providing an overview of the two applications which we use to illustrate our points. We first briefly describe the Multimission VICAR Planner application, in which planning techniques are used to automatically generate image processing programs from user specified image processing goals. We then briefly describe the LMCOA application, in which an automated reasoning system provides monitor, control, and decision support capabilities for operating Deep Space Network Antennas.

2.1 MVP: Automated VICAR Image Processing

The Multimission VICAR Planner (MVP) [Chien 1994a, Chien 1994] system is an Artificial Intelligence (AI) Planning system, which automatically constructs executable complex image processing procedures using models of the smaller constituent image processing subprograms in response to image processing requests made to the JPL Multimission Image Processing Laboratory (MIPL). The MVP system allows the user to specify the image processing requirements in terms of the various types of corrections required. Given this information, MVP derives unspecified required processing steps and determines appropriate image processing programs and parameters to achieve the specified image processing goals. This information is output as an executable image processing program which can then be executed to fill the processing request. In the manual approach, a group of human experts, called analysts, receive written requests from scientists for image data processed and formatted in a certain manner. These analysts then determine the relevant data and appropriate image processing steps required to produce the requested data and write an image processing program in a programming language called VICAR (for Video Image Communication and Retrieval [1]) [Lavoie et al. 1989]. Unfortunately, this current mode of operations is extremely labor and knowledge intensive. This task is labor intensive in that constructing the image processing procedures is a complex, tedious process which can take up to several months of effort. This task is knowledge intensive in that it requires substantial knowledge of image processing, specifics of VICAR image processing programs, VICAR language constructs, and file and database organization and content. VICAR procedure generation is a common task - there are currently tens of analysts at MIPL alone whose primary task is to construct these VICAR programs. Many other users at JPL and other sites also write VICAR scripts, with the total user group numbering over 100. MVP2.0 is currently operational and available for use by analysts at JPL's Multimission Image Processing Laboratory (MIPL). MVP2.0 is written in C and operates with a Motif-based GUI on Sun workstations. For radiometric correction, color triplet reconstruction, and mosaicking tasks. MVP reduces effort to produce an initial PDF for an expert analyst from 1/2 a day to 15 minutes, and it reduces the effort for a novice analyst from several days to 1 hour. Thus MVP has achieved roughly a 15-fold reduction in the effort to complete these tasks.

2.2 LMCOA and the Deep Space Network

The Jet Propulsion Laboratory manages a world-wide network of antennas, the Deep Space Network (DSN), which is responsible for providing the communications link with a multitude of spacecraft. Operations personnel are responsible for creating and maintaining the communications link by configuring the required subsystems and performing test and calibration procedures. This task of creating the communications link, known as precalibration, is a manual and time-consuming process which requires operator input of over a hundred control directives and operator monitoring of over a thousand event messages and several dozen displays to determine the execution status of the system. The existing Link Monitor and Control (LMC) system requires the operator to perform a large amount of textual keyboard entries, to monitor and interpret a large number of messages to determine the state of the system and to selectively cull out relevant information from dozens of pre-defined, data-intensive displays. This results in an environment in which it is difficult to operate efficiently. The Link Monitor and Control Operator Assistant (LMCOA) uses automated operations techniques which improve operations efficiency and reduce precalibration time. The LMCOA is a knowledge-based prototype system that provides semi-automated monitor and control functions to support operation of the DSN 70-Meter antenna at the Goldstone Deep Space Communications Complex (DSCC). Improved operations is achieved by using a flexible and powerful procedural representation, by reducing operator keyboard entry and by providing explicit closed loop communications and control through an expert system module. An operational version of the LMCOA is in use at a DSN 34-meter antenna station at Goldstone, California to perform Ka band antenna performance experiments [Hill 1994]. The current prototype reduces the amount of operator inputs per antenna track under nominal conditions from about 700 to less than 10. While originally the general plans used by the LMCOA were developed manually (for example,

[1] This name is somewhat misleading as VICAR is used to process considerable amounts of non-video image data such as MAGELLAN synthetic aperture radar data.

the general plan which implements the Ka band antenna performance experiments), in February 1995 a planning system was demonstrated which automatically generates plans based on information describing the track type and the equipment configuration.

3 Representation Issues

Many of the obstacles hindering application of planning techniques to real-world problems can be characterized as representational difficulties. Of the representational issues we have encountered in the LMCOA and MVP applications, several can be attributed to the general area of representing and reasoning about plan quality [2]. Other representation issues include representing complex actions and action effects.

3.1 Representation Issues in MVP

In the MVP application, an important concern is output image quality. For a planning system to be able to represent large portions of an analyst's expertise, the planner must be able to represent and reason about the effect of various image transformations on image quality. For example, one of the most common image processing requests is for mosaicking, which is the process of combining a number of smaller images into a larger image. A frequent situation in mosaicking is that some of the images can be navigated absolutely - that is to say that some images contain features that make it possible to exactly align these images (this is called absolute navigation). However, the remainder of the images can only be correctly placed on the output image by matching up points which are believed to be common between them and other images (tiepoints). This is a more difficult process known as relative navigation. When performing relative navigation, there are various measures of the confidence of the navigation process (such as residual errors). When processing these images, in order to produce a high quality image, the VICAR script must take into account the relative weights of alignment information from these different sources. Navigation information from absolute navigation should be weighed more than relative navigation information and relative navigation information is of varying degrees of confidence. Ideally, an expert image processing planning system would be able to reason about the navigation process and various measures of image quality, to determine at runtime the best order in which to process the images. In the best situation, the operator and plan representation would be able to explicitly represent these measures of image quality and at run-time execute the steps to ensure a high quality image. This would require characterizations of plan quality relating to measurable runtime attributes. A secondary, but also important concern for MVP is the computational efficiency of the produced plan. If the image quality will be equivalent, there are sometimes different methods of achieving the same image processing goals but with different characteristics of computer runtime or disk storage. If MVP can reason about these types of costs for plans it will be able to produce plans which are more acceptable to the analysts and scientists.

[2] For other work on planning and reasoning about plan and schedule quality see [Haddawy and Hanks 1993, Williamson and Hanks 1994, Perez and Carbonell 1994, Gratch et al. 1993]. Decision theoretic notions of utility functions also capture plan and schedule quality metrics [Decision 1994].

3.2 Representation Issues in LMCOA

The LMCOA uses a temporal dependency network (TDN) [Fayyad 1993] to represent and automate LMC operations procedures. A TDN is a directed graph that incorporates temporal and behavioral knowledge and also provides optional and conditional paths through the network. The directed graph represents the steps required to perform an operation. Precedence relationships are specified by the nodes and arcs of the networks. The behavioral knowledge identifies system-state dependencies in the form of pre- and post-conditions. Temporal knowledge consists of both absolute (e.g. Acquire the spacecraft at time 02:30:45) and relative (e.g. Perform step Y 5 minutes after step X) temporal constraints. Conditional branches in the network are performed only under certain conditions. Optional paths are those which are not essential to the operation, but, for example, may provide a higher level of confidence in the data if performed. Each node in the TDN is called a "block" and contains a sequence of actions to be performed. Each block also has pre- and post-condition constraints and time tags associated with it.

Representing and reasoning about plan quality in TDNs is a key concern in the LMCOA application domain in several ways. First, overall execution time to setup (precalibration) and reset (post-calibration) the communications link subsystems should be minimized. Reducing this execution time allows more data to be returned per operating time for the link. For instance, it can take up to two hours to manually pre-calibrate a DSN 70-meter antenna communications link for certain types of missions. Using the LMCOA, this time can be reduced to approximately thirty minutes, where further reductions in set-up time are limited by physical constraints of the subsystems themselves. Changes in post-calibration can also reduce pre-calibration time for a subsequent track. For instance, if a following track requests a similar antenna operation to the one being currently executed, it may be unnecessary and wasteful to reset many of the antenna subsystems. Since many of the system settings will not vary between the two tracks, resetting these systems will only cause extra time to be spent on recalibration during the second track. These types of reductions in operations time can save thousands of dollars each time precalibration is performed. For this reason, plan execution time is a primary measure of plan quality.

Plan execution time can often be significantly reduced by exploiting parallel path execution where the control of multiple subsystems is involved. When developing a planning system to automatically generate TDNs, we would like the system to reason about plan execution time as a measure of plan quality. Since there can often be more than one correct plan for a particular antenna operation, it is important for a planning system to be able to compare a set of final plans using user identified plan quality measures. Our planner currently uses the critical path length of a plan to help identify better plans. Critical path length is calculated using time information attached to a TDN block which specifies the average time it should take to execute the block. By comparing critical path lengths of competing plans, our planner can choose the plan which will provide a minimal plan execution time.

Another measure of plan quality is generality. One of the missions frequently performed in the LMCOA domain is called the Ka-band Antenna Performance (KaAP) experiment. The KaAP TDN is currently implemented for the operational LMCOA; it is considered a generalized TDN since it represents the many different ways that a KaAp experiment can be executed. The support data for a particular KaAP experiment identifies a particular path through the TDN. For example, there is a data capture loop in the KaAP TDN which allows data to be captured from either a star or a planet, thus requiring different antenna modes. One experiment may specify that data be acquired from the following sources in sequence: star1, star2, star3. Whereas another experiment may specify that data by acquired from: star1, planet1, star1, star2. In current operations, TDNs are manually generated, verified, and refined. Because of the considerable effort in manually generating, maintaining, and refining TDNs, a single generalized TDN is cheaper than hundreds or thousands or experiment-specific TDNs. Even in the current development to automate TDN generation, the planning knowledge base must be constantly updated and verified. Fewer generalized TDNs are cheaper to update and verify, and thus support more efficient knowledge base maintenance.

Flexibility is another aspect of plan quality that has been a requirement in the LMCOA. For instance, the support data for a particular experiment may specify a particular path through the TDN, however, the operator has the flexibility to alter this path in real-time. The TDN and LMCOA must be able to handle these real-time changes. Some of the changes that the operator can make to the TDN are skipping blocks, deleting commands in blocks, adding commands in blocks, and editing time tags on blocks. It may also be necessary (or desirable) for an operator to reorder blocks. For example, some TDN blocks cannot execute in parallel due to resource conflicts. The ordering of such blocks can often affect plan quality by making a plan more robust or more efficient, depending on the particular antenna operation and current track status. If a better ordering is known prior to TDN generation, this information can be input to the planning system which will incorporate it into the final TDN. However, these ordering constraints may often be best determined at runtime by the operator.

There are also standard blocks that may be inserted into TDNs at various points (such as transmission rate changes, etc.). If such commands are executed in the middle of an inflexible TDN, it may not be possible to continue execution. Depending on the steps inserted, preconditions, postconditions, and time tags of other blocks may become invalid. Flexible TDNs that allow for the insertion of common steps while still retaining their applicability are greatly valued.

Finally, the plan representation must be expressive in order to provide robustness; however, an expressive representation usually increases an application's complexity and often results in a loss of generality. In the LMCOA application, the TDN representation was initially kept extremely simple, although it did include parallelism. As the intricacies of a particular domain's procedure became evident, more expressive representations were required. Constructs such as loops, metric time, and actions with temporal scope were added. As a prototype, the LMCOA became more complicated and very specific to a particular TDN. For example, a "loop until time" construct was required where the actions in the loop would be executed until a pre-specified time occurred. At that time, execution of the loop would continue until a pre-specified exit point had been reached where the loop could be safely exited. This actually caused the exit time of the loop to be after the time specified in the looping construct. The alternative of abruptly executing the loop at a particular time is not always acceptable. Such a construct was necessary for one particular TDN, however, it may not be as applicable to other TDNs.

Lesson 1 *Current plan representations are impoverished; planning representations need to be able to represent many aspects of plans other than goal achievement, (e.g., plan quality) such as: MVP-image quality, resource usage (e.g., disk usage), execution time, generality, flexibility, and robustness.*

4 Operational Contexts

Several of the difficult aspects of the MVP and LMCOA applications relate to what we call the operational context of the application system. In planning research, the planning problem typically is characterized as a batch problem, where the inputs and outputs of the system are carefully specified, and the planning system must produce a complete solution without user intervention. In the real world, this is rarely possible. Commonly the solution produced by the planner must be verified and applied with some human intervention. This has strong ramifications for the plan produced by the planning system - it must be understandable and modifiable by the user. In some cases the operational context is much more demanding - the actual plan generation process will have to be an interactive mixed-initiative process. [3]

4.1 Operational Contexts and MVP

In the MVP application, plans may be formed which require user inputs. For example, MVP may need to construct plans which involve determination of tiepoints between overlapping images (tiepoints are common reference points which appear in adjacent images and allow determination of points on one image relative to the other). In some cases the tiepoints can be determined automatically, but in other cases analyst intervention may be required to produce a high quality image. If the user specifies a pattern of goals which requires user interaction, MVP must produce a plan which contains appropriate interaction points. Fortunately, so far in the MVP domain encoding, we have been able to structure these points as simple loops, where the user can redo certain steps until they are satisfied with the end result. Furthermore, there are occasionally program parameters which may need to be adjusted by human analysts in a subjective fashion after inspecting the final image. In other more rare cases, the analysts may need to modify the produced image processing scripts to add further processing steps. Thus, because analysts must be able to modify MVP output, it is key that human analysts be able to understand and interpret MVP generated plans. In order to fulfill this requirement, MVP uses a hierarchical task network (HTN) planning component to produce an abstract plan to solve a problem. This abstract plan is annotated with high-level comments generated during the plan construction process. These comments detail at a conceptual level why MVP decomposed the problem in the manner it chose and which high-level goals are being attacked in which portion of the plan. This annotation greatly assists the analysts in understanding the structure of the produced image processing plans. At a lower level, the plan dependency structure itself can be used to explain the plan. This structure can be used to explain why certain image processing steps are needed, why certain parameters were set to the values used, or why image processing steps occur in the produced ordering [4]. More generally, planning systems need to be able to interact more gracefully with users and produce more understandable results (see [Arpa1994, Ferguson 1994]).

[3] For a more detailed discussion of these issues see [Arpa1994].

[4] For further discussion on the relative merits of HTN and operator-based planning see [Drummond 1994, Kambhampati 1994].

4.2 Operational Contexts and LMCOA

The LMCOA application has to deal with several aspects of the operational context that affect planning: the domain is asynchronous, real-time, and interactive. By asynchronous we mean that the effects of an action cannot be immediately observed and it may not have its intended effect. This affects the execution of the plan by forcing the LMCOA to monitor the state of the devices to which the plan's control actions have been sent. It must be able to recognize whether the action had its intended effect, and it must be able to deal with situations where the action had no effect at all or an unintended effect. For instance, an action may be sent to a device and there may be no response indicating that the action was received and executed. The LMCOA must take a corrective action once a time limit has passed and it has not been able to verify that the effect has occurred. More generally, a planner often must explicitly consider execution monitoring and verification of successful goal and subgoal achievement as an active task.

The LMCOA domain is real-time in that there are temporal constraints on the achievement of a plan's goals, which forces the LMCOA to continually monitor the plan's execution status as well as progress toward achieving the plan's goals. As previously indicated, the temporal constraints of the domain have to be taken into account when making decisions about re-planning after a plan failure.

The LMCOA domain is interactive, meaning that the plan is not simply executed, rather, it is often necessary to re-plan or otherwise compensate for an interaction with the plan or the environment during its execution. Many external events may interrupt a plan's execution. For example, a TDN block may fail to achieve its effects during its execution; a subsystem may fault during precalibration; additional services (such as telemetry, commanding, or ranging) may be added in during execution; equipment may be removed due to external requests; or the human operator may intervene by adding or deleting steps in the TDNs. Such interaction with external events requires the planning system to have the ability to re-plan upon these external events or failures.

The re-planning component needs to deal with the following issues in the DSN operational contexts:

(1) Re-planning requires knowledge that is usually not represented in the TDNs. For each execution failure of a block, there is a series of specific corrective actions that the operator takes to repair the failure. But since there can be hundreds of blocks, it is not practical to hand-code in the repair mechanisms for each specific block. A general framework for representing repair knowledge is desirable.

(2) There is a tradeoff in the granularity level used to represent the TDN blocks. The blocks are the lowest level primitives that the planner reason about, each block may contain tens of directives (commands). Sometimes during an execution failure, instead of re-executing a whole block, it is possible to only re-execute a subset of all the directives in the block, so that the total execution time may be shortened. To capture this plan repair knowledge, we can break a block down to a number of blocks, but then the planner must reason at a lower level of abstraction. This may result in a less maintainable knowledge base for the planner and degraded planner performance (planning speed). This tradeoff is similar to the generality issue for plan quality discussed in section 3.1.

(3) Actions take time. If the recovery actions take an extended amount of time, there may not be enough time to perform a planned equipment performance test as well as starting the acquisition of data at the required time. In this case, a tradeoff must be

evaluated. For example, should the data be captured without doing the performance test? Or would the data be useless without the performance test?

(4) Once corrective actions are taken, or the operator's intervention has been completed, the LMCOA must be given the command to continue execution of the TDN in the new context. The preconditions of the blocks in the original TDN that have not yet been executed may have changed. What actions are necessary in order to satisfy these preconditions?

(5) During execution, some subsystems may be removed due to competing requests. Usually, these subsystems are not needed any more by the task, and are requested to be used by other tasks. What is the proper way to remove the equipment from the system? How do we unlink it with other subsystems?

(6) The state of the subsystems has a large amount of information. Although in principle, all relevant state information can be inferred by an expert operator, practice often deviates from this standard. How does planning system help the operator attend to failures?

When examining the above sample issues that arise from the operational context of LMCOA, we see that a fully functional operational system requires the ability to integrate planning, execution, and re-planning, rather than simply just do one-pass (batch) planning.

Lesson 2 *Planners must fit into the operational context of the application. Most planning tasks involve user interaction - this requires that planners generate plans that the user can understand and modify. In some cases the user must be able to insert actions during execution with the plan recovering and resuming execution. In other cases the user may need to interact with the planner during plan construction. Planners must also often operate in environments which are asynchronous and real-time.*

5 Knowledge Acquisition and Knowledge Base Maintenance

One of the key elements in determining the feasibility of fielding a planning application is an assessment of the amount of effort and expertise required to construct the knowledge base and update and maintain the planning knowledge base. This has been particularly true in our experiences with the LMCOA and MVP applications. As a result, we have expended considerable effort in customizing the knowledge representations used for these applications and developing tools to facilitate knowledge base development and maintenance. While considerable work has been done in knowledge acquisition environments, this work has not focused on the specialized planning representations (task reduction rules and operators) and constraints (ordering, codesignation, etc.).

5.1 Knowledge Acquisition and Maintenance in MVP

In the MVP application, knowledge is represented in the following forms: task decomposition rules for reducing high-level tasks or goals into lower level tasks and goals, planning operators, syntax rules for generating correct syntax VICAR programs from image processing plans, and rules for generating the initial state from database label information. While the knowledge base is of only moderate size (as of 12/94: approximately 60 planning operators, 50 decomposition rules, tracking about 70 file attributes), since becoming operational in 5/95, the vast majority of the project effort has been devoted to verification, maintenance, and extension of the planning knowledge base (almost 1 full-time person). In the MVP application, we have developed two types of tools to assist in knowledge base development and maintenance. Static analysis tools analyze the knowledge base to detect simple cases where goals cannot be achieved. These cases are flagged and the user notified of these pathological cases. Completion analysis tools allow the user to detect cases where plans were almost able to be completed, but a certain subgoal could not be achieved or a certain protection could not be enforced. Completion analysis tools allow the user to quickly focus his attention on a specific portion of the knowledge base. These tools are described in further detail in [Chien 1994b]. These knowledge engineering tools are essential to providing a software lifecycle cost competitive with other software automation alternatives. Initial efforts by image processing personnel to automate VICAR processing by writing extremely general VICAR scripts for general problem classes. However, manual generation of scripts is expensive in both expertise and effort - due to the knowledge required to generate scripts and the many problem types. Additionally, maintaining these scripts is a time consuming task requiring significant VICAR expertise. Subsequent efforts involved the application of rule-based expert systems technology to automate VICAR procedure generations. This approach encountered problems in scaling up due to the difficulty of developing a modular, maintainable knowledge base. Planning technology offers an alternative superior to manual script generation in that representation of general VICAR knowledge can allow automatic generation of scripts to fill a wide range of requests. Additionally, the planning knowledge representation is a natural match for the VICAR procedure generation task and encourages modularity and explicit representation of dependencies.

5.2 Knowledge Acquisition and Maintenance in LMCOA

Knowledge represented in the LMCOA application includes TDN block information (preconditions, postconditions, related TDN blocks (predecessors and successors), directives), TDNs themselves, hierarchical task reduction rules to indicate which operations procedures are relevant for particular antenna passes and equipment assignments, and subsystem models (to track antenna subsystem state via event status notices from the subsystems and issued directives). Because of the complexity of these representations, the process of building the knowledge bases to represent a single TDN is manual and tedious. To date 7 TDNs have been constructed; these TDNs include from 21 to 73 blocks, and contain over 100 directives. Each TDN might involve models of 3-7 subsystems. Several tools are under development to assist in the acquisition and maintenance of the plan knowledge base (for more details see [Hill et al. 1994]). A TDN authoring tool is being developed to automate the specification of TDNs. Developers as well as operations personnel will be able to graphically specify the TDN and its contents. TDNs can be composed from parts of existing TDNs and libraries of actions at the least. A database will efficiently store a complete specification of a TDN as part of a TDN library. The same database will serve as a central repository for the TDN in the LMCOA, thus simplifying the LMCOA implementation. In addition, the TDN authoring tool will include the capability to verify certain aspects of the TDN such as incompatible block ordering based on pre- and post-condition constraints of blocks. This TDN database will also allow TDN developers to quickly and easily access TDN information related to the TDNs, blocks, subsystems, and directives being modified. For example, when constructing a new TDN block, the developer will be able to easily access other blocks containing the same directives as well as access other blocks affecting the same subsystem state variables as the current block. The knowledge engineering effort for the LMCOA prototype is described in more detail in [Fayyad 1993]. Besides the TDN authoring tool, two other tools, RIDES and REBUS, are being developed and used for knowledge acquisition. The RIDES simulation authoring tool kit [Munro et al. 1993] is used to capture device models of the communications link equipment and subsystems. Besides using these models in the planner, the simulator also permits us to test the LMCOA's ability to cope with the operational context issues described in the previous section. REBUS, which stands for Requirements Envisioning By Utilizing Scenarios, [Zorman 1995] is used to capture knowledge about the domain by using different scenarios to provide contextual information needed for planning. This provides us a way to understand how the subsystems controlled by the LMCOA actually work and how to control them under both normal and anomalous conditions. Another method being considered for automating DSN antenna operations is the use of a library of general purpose scripts. We feel that the general TDN/plan representation can offer automation of a wider range of antenna operations tasks. Additionally, we believe that explicit representation of dependencies among operations procedures will facilitate maintenance of this knowledge base and allow for use of the knowledge base for other purposes such as documentation and training. However, development of tools to facilitate in knowledge acquisition, verification, and maintenance to reduce the planning system software lifecycle costs are of prime importance. These requirements are likely to require algorithms and techniques specialized to the particular representations used by MVP and LMCOA.

Lesson 3 *Planning systems must have reduced software lifecycle costs as compared to other means of automation such as scripts or rule-based systems. Development of intelligent knowledge acquisition, verification, and debugging tools for planning knowledge bases is essential.*

6 Summary

We have described a number of issues which complicate the application of planning technology to real-world problems. While we have described these issues in the context of two planning projects at JPL: MVP - a planning system for automated generation of image processing procedures; and LMCOA - an intelligent system for assistance in antenna operations, these issues are general issues applicable to other application areas. First, we described how existing planning representation must be enhanced to represent and reason about aspects of plans besides goal achievement - resource usage, quality, execution time, flexibility, and generality. Second, planning systems must be able to fit into a wide range of operational contexts. Most importantly, most planning tasks cannot be completely automated, therefore at a minimum the plans produced must be easily understandable and modifiable by the users. In some cases the user must be intimately involved in the plan construction process itself. Third, planning systems must be able to compare favorably in terms of software lifecycle costs to other means of automation such as scripts or rule-based expert systems. This means that development of intelligent tools and environments to facilitate knowledge acquisition, validation, and maintenance are of prime importance. We hope that our description and elucidation of these issues will lead to increased work in these areas.

References

[Arpa1994] M. Burstein and D. McDermott, A. Tate, S. Smith, D. Weld, Position statements on Mixed Initiative Planning, Proceedings 1994 ARPA Workshop on Planning, 1994.

[Chien 1994a] Chien, S. A., "Using AI Planning Techniques to Automatically Generate Image Processing Procedures: A Preliminary Report," Proceedings of the Second International Conference on AI Planning Systems, Chicago, IL, June 1994, pp. 219-224.

[Chien 1994] S. A. Chien, "Automated Synthesis of Image Processing Procedures for a Large-scale Image Database," Proceedings of the First IEEE International Conference on Image Processing, Austin, TX, November 1994.

[Chien 1994b] S. A. Chien, "Towards an Intelligent Planning Knowledge Base Development Environment," Proceedings of the 1994 AAAI Fall Symposium on Planning and Learning: On to Real Applications, New Orleans, LA, 1994.

[Decision 1994] Proceedings of the 1994 Spring Symposia on Decision Theoretic Planning, March 1994, AAAI Press.

[Drummond 1994] M. Drummond, "Precondition Achievement Planning: framework sans application?," Proceedings of the AAAI94 Workshop on Planning Benchmarks.

[Fayyad 1993] Fayyad, K. E., R. W. Hill, Jr. and E. J. Wyatt, "Knowledge Engineering for Temporal Dependency Networks as Operations Procedures", AIAA '93, Computing in Aerospace 9 Conference, San Diego, CA, October 1993.

[Ferguson 1994] Ferguson, G. and J. Allen, Arguing about Plans: Plan Representation and Reasoning for Mixed Initiative Planning," Proceeding of the Second International Workshop on AI Planning Systems, pp. 43-48.

[Gratch et al. 1993] Gratch, J. M., S. A. Chien, and G. F. DeJong, "Learning Search Control Knowledge to Improve Schedule Quality," Proceedings of the 1993 Workshop on Knowledge-based Production Planning, Scheduling, and Control, Chamberry France, August 1993, pp. 159-168.

[Haddawy and Hanks 1993] P. Haddawy and S. Hanks, "Utility Models for Goal-directed Decision-Theoretic Planners," Technical Report 93-06-04, University of Washington, Department of Computer Science and Engineering, September 1993.

[Hill 1994] Hill, R. W., Jr. ""Automating Monitor and Control", DSN Technology Program News, JPL, Pasadena, CA, December 1994.

[Hill et al. 1994] Hill, R.W., Jr.., Fayyad, K., Santos, T., Sturdevant, K. "Knowledge Acquisition and Reactive Planning for the Deep Space Network." Proceedings 1994 AAAI Fall Symposium on Planning and Learning: On to Real Applications, New Orleans, LA, 1994.

[Kambhampati 1994] S. Kambhampati, "A Comparative Analysis of Partial Order Planning and Task Reduction Planning," SIGART Bulletin Vol. 6 No. 1.

[Lavoie et al. 1989] S. LaVoie, D. Alexander, C. Avis, H. Mortensen, C. Stanley, and L. Wainio, VICAR User's Guide, Version 2, Internal JPL Document D-4186, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 1989.

[Lee and Cooper 1993] Lee, L. F. and L. P. Cooper, "Operations Automation Using the Link Monitor and Control Operator Assistant", AIAA '93, Computing in Aerospace 9 Conference, San Diego, CA, October 1993.

[Munro et al. 1993] Munro, A., Johnson, M.C., Surmon, D.S., and Wogulis, J.L.. "Attribute-Centered Simulation Authoring for Instruction. "Proceedings of AI-ED 93, World Conference on Artificial Intelligence in Education, Edinburgh, Scotland; 23-27 August 1993.

[Perez and Carbonell 1994] Perez, M. A., and J. G. Carbonell, "Control Knowledge to Improve Plan Quality," Proceedings of the Second International Conference on AI Planning Systems, Chicago, IL, June 1994, pp. 323-328.

[Williamson and Hanks 1994] Williamson, M. and S. Hanks, "Optimal Planning with a Goal-directed Utility Model," Proceedings of the Second International Conference on AI Planning Systems, Chicago, IL, June 1994, pp. 176-180.

[Zorman 1995] Zorman, L, “Requirements Envisaging by Utilizing Scenarios,” PhD. Dissertation in preparation, University of Southern California, 1995.