Planning for Deep Space Network Operations

Randall Hill, Jr., Steve Chien, Crista Smyth, Kristina Fayyad, Trish Santos
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, M/S 525-3630
Pasadena, CA 91109-8099
{hill, kristina, crista, trish}@negev.jpl.nasa.gov chien@aig.jpl.nasa.gov

Abstract

We are developing a plan construction and plan execution monitoring system to automate the operation of the communications link in NASA's Deep Space Network (DSN). The objectives of this effort are: (1) to archive knowledge of operations procedures; (2) to provide a capability to assist in execution monitoring; and (3) provide a capability for automatically constructing the plan to be executed and monitored based on an augmented sequence of events (SOE). This paper describes the application area of DSN antenna operations, describes the plan representation of a Temporal Dependence Network (TDN), outlines the knowledge engineering tools being constructed to facilitate in representing DSN operations procedures, and describes preliminary work in automated TDN generation.

1. Introduction

One of the significant challenges to developing and fielding a real-world planner is in the representation and acquisition of the knowledge needed for it to function autonomously in a real-world problem domain. This paper describes how we are addressing the knowledge acquisition and representation problem in the development of an intelligent system for representation of procedural knowledge representation, plan execution monitoring, and plan construction, for a real-world domain: the communications link monitor and control system of NASA's Deep Space Network (DSN). The underlying motivation driving this automation is improving the productivity of the DSN while maintaining at least the current level of reliability. To accomplish this goal automated plan generation and execution monitoring must be capable of performing many of the same operations that are currently done manually by DSN station Operators, and thereby significantly reduce the amount of human effort needed to perform monitor and control tasks in the domain.

We have elsewhere identified a number of factors that make it difficult to implement a planner for a domain like the DSN (see Chien, Hill, and Fayyad, 1994). In this paper we describe our efforts to develop a plan construction, execution and monitoring system to automate the operation of the communications links in NASA's Deep Space Network (DSN). The objectives of this effort are: (1) to archive knowledge of operations procedures; (2) to provide a capability to assist in execution monitoring; and (3) provide a capability for automatically constructing the plan to be executed and monitored based on an augmented sequence of events (SOE).

We begin by giving a brief overview of the DSN task domain and the nature of problem solving there. Next, we describe how we are automating the operation of the DSN through the use of a plan execution and monitoring system called the Link Monitor and Control Operator Assistant (LMCOA), which uses a plan representation scheme called a Temporal Dependency Network (TDN). A description is given of how plan knowledge will be captured using some knowledge acquisition tools that are currently under development.

Next, in the section entitled "SOE-driven automation" we describe the planner that will eventually be capable of generating all of the plans needed to operate the DSN for a wide variety of telecommunications services. We call this "SOE-driven automation" because the input to the planner is a sequence of events (SOE) that describes the basic track being requested. The planner uses the SOE to create goals that must be satisfied by the plan it generates. Examples from the DSN domain are given that show the kinds of information the planner works with as well as what it produces.

2. Deep Space Network Domain

The Deep Space Network (DSN) is a worldwide network of deep space tracking and communications complexes located in Madrid, Spain, Canberra, Australia, and Goldstone, California. Each of these complexes is capable of performing multiple missions simultaneously, each of which involves operating a communications link. A DSN communications link is a collection of devices used to track and communicate with an unpiloted spacecraft.

Currently, most of the tasks requiring the control of a DSN communications link are performed by human Operators on a system called the LMC (Link Monitor and Control) system. The Operators are given tasks that involve configuring and calibrating the communications equipment, and then they monitor and control these devices while tracking a spacecraft or celestial body. The Operators follow written procedures to perform their mission tasks. A procedure specifies a sequence of actions to execute, where the actions are usually commands that must be entered via the link's monitor-and-control system keyboard.

Once issued, a command is forwarded to another subsystem, which may accept or reject it, depending on the state of the subsystem at the time that the command is received. The Operator receives a message back from the subsystem indicating whether the command was accepted or rejected, and in cases where there is no response, a message saying that the command "timed out" is sent. These messages do not indicate whether the action was successful or what the results of the action were. Rather, the Operator has to monitor subsystem displays for indications that the action completed successfully and that it had its intended effects. It is common for commands to be rejected or for commands to fail due to a number of real-world contingencies that arise in the execution of a plan or procedure.

3. Automating the DSN

A prototype called the Link Monitor and Control Operator Assistant (LMCOA) was developed to improve Operator productivity by automating some of the functions of the LMC. The LMCOA performs tasks by: (1) selecting a set of plans to execute, (2) checking whether a plan's preconditions have been satisfied, (3) issuing the commands, and (4) subsequently verifying that the commands had their intended effects. The Operator interacts with the LMCOA by watching the plans as they are being executed and pausing or skipping portions of the plan that need to be modified for some reason. When a plan fails, the LMCOA lacks the ability to recover on its own. Instead, the Operator is left to figure out how to recover from the failure. In observing Operators perform their duties, we find that they have an amazing ability to adapt the written procedures to the contingencies that arise in the course of their execution. A real-world planner has to have the same flexibility and knowledge as a human Operator in order to succeed in performing DSN missions.

Temporal Dependency Network (TDN)

We use a representation called a Temporal Dependency Network (TDN) to express the basic plan structure and interrelationships among plans. 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 relations are specified by the nodes and arcs of the network. The behavioral knowledge identifies system-state dependencies in the form of pre- and postconditions. 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 may, for example, provide a higher level of confidence in the data resulting from a plan if performed. Each node in the TDN is called a plan and contains actions to be performed. A plan also has goals, pre- and postcondition constraints and temporal constraints associated with it. More details about TDNs are provided in (Fayyad & Cooper 1992).

In our experience of building TDNs, the knowledge bases associated with even a single TDN are quite large and difficult to build and maintain. In addition, they rely on information from many different sources: documentation in paper form, interviews with experts, and text logs of operations procedures from actual missions, just to name a few. As we build more TDNs, much of this information can be used again.

Knowledge Acquisition Tools

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 postcondition 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 et al., 1993).

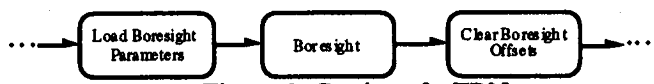

Figure 2. Portion of a TDN

With the TDN authoring tool, users can graphically build a TDN and specify all the information required in a TDN. For example, they graphically create boxes representing plans, and draw the links between the plans to represent dependencies between plans. A small segment of a TDN is shown in Figure 2 and contains three plans in sequence. This may be all that the users sees if she initially creates just a high level TDN including skeletal plans and a partial order of plans. When a plan is created, the user gives it a name and specifies what type of plan it is; there are different types of plans depending on the type of actions to be taken in a plan, e.g. a manual action to be taken by the Operator, or a list of commands to be sent to a subsystem. In our example, each of these plans is a command plan associated with the 34 meter Antenna Subsystem. The sequence of actions for the plan named Clear Boresight Offsets is shown in Table 1. The first two actions set the Elevation and Cross Elevation correction offsets to 0. The last action sends the structure containing these two variables to the subsystem.

Plan:	Clear Boresight Offsets	
Actions:	setvar A34CorrEl	0
	setvar A34CorrXel	0
	sendvar ActualCorrData	

Table 1. Plan actions.

Plan:	Clear Boresight Offsets
Action:	sendvar ActualCorrData
Postcondition:	ActualCorrData.el = 0
	ActualCorrDatax.el = 0

Table 2. Postcondition for Clear Boresight Offsets.

For each plan, the user can then specify the sequence of actions within the plan, the pre- and postconditions of each action, goals of plans, and temporal constraints on plans. Depending on the knowledge that the user has about the actions, he may or may not be able to specify all of this information. The postconditions of the last action in Table 1. fall out nicely from the action itself; though this isn't always the case. Since this last action sends data to the subsystem, it is necessary to make sure that the subsystem did indeed receive the new data values. In this case, we are only concerned with the two variables in the structure which correspond to the first two actions in the plan. The postconditions for the last command in Clear Boresight Offsets are shown in Table 2.

In our current prototype, most of the information about a plan is specified by filling in text forms. As the tool develops, knowledge bases of actions with their associated pre- and postconditions and knowledge bases of plans will be available so that new TDNs can be built from pieces of existing ones.

The outputs of the tool are a text file that completely defines the TDN and a file depicting the graphical representation of the TDN. These two files will be loaded by the execution monitor. The graphical representation is the basis for a dynamic display of the TDN in the execution monitor which shows the progress of the TDN as it is executing. The text file is loaded by the execution monitor which uses it to create an executable TDN.

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, 1994) 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.

4. SOE-Driven Automation

While the currently operational LMCOA uses hand-constructed, manually selected plans from a plan library, the current work focuses on automated generation of operations procedures for DSN antenna operation. This capability is to be demonstrated at the end of January 1995. In this task, several sources of information are available:

- *Project SOE* - The project sequence of events specifies events from the mission/project perspective. As a result, the project SOE contains a great deal of information regarding the spacecraft state which is relevant to the DSN track, as well as a large amount of spacecraft information unrelated to DSN operations. Relevant information specified in the project SOE includes such items as the one-way light time (OWLT) to the spacecraft, notifications of the beginning and ending times of tracks, spacecraft data transmission bit rate changes, and carrier and subcarrier frequency changes. An excerpt from a VOYAGER project SOE is shown in Figure 4.1.
- *Project profile* - This file specifies project specific information regarding frequencies and pass types. For example, the Project SOE might specify bit-rate = HIGH, and the project profile would specify the exact bit rate used. The project profile might also specify exact frequencies used, and default track types.
- *LSOE* - Using the project profile information, the project SOE is parsed into a more manageable form specifying only the information directly relevant to the specific pass. A sample LSOE for a VOYAGER telemetry downlink pass is shown in Figure 4.2 below.
- *TDN KB* - The TDN knowledge base stores information on the TDN blocks available for the DSN Planner and LMCOA to use. This database includes information regarding preconditions, postconditions, directives, and other aspects of the TDN blocks.

```
* ID=DSNISOE52B1        S/C=032 BEG=94 360 094000
  END=95 001 221500
! 032 42 00022 360 201500 ACE NCT S/C ANT
  STATUS, RCP/00.100/Y, HGA, RCP/00.100/Y, HGA
! 032 42 00022 360 201500 ACE NCT    S/C TXR
  STATUS, OFF, EXC1/TWT2/18, LOW,EXC1/TWT2/18
! 032 42 00022 360 201500 ACE NCT    S/C RCVR
  STATUS, 2/S, ON
! 032 42 00022 360 201500 ACE NCT    S/C RNG
  MODE, OFF, OFF, ON, OFF
! 032 42 00022 360 201500 ACE NCT    S/C TLM X-
  MOD, 160CD, LOW, 61DG
! 032 42 00022 360 201500    D42    BOT
! 032 42 00022 360 201500    D42  ACQ D/L, 1-WAY, X
! 032 42 00031 361 095000    D42    EOT
```

Figure 4.1: Sample VOYAGER Project SOE

95/005/16:30:00 ACQ_D/L: MODE=1-WAY; BAND=X;.
95/005/16:30:00 CONFIG: IF3=S; IF4=X; POLAR=RCP;
SKY_FREQ=10000000; PRED_POWER=50;
PO_FILE="/u/cm/lmcoa/lmcoa/src/kb/source.po";.
95/005/16:30:00 TPR_PRECAL: BAND=S,X;.
95/005/17:15:00 SOURCE: SRC=VGR2;.
95/005/17:15:00 CARRIER: PC_NO=34; FREQ=70000000;
BANDWIDTH=1.0; UPDATE_RATE=500; LOOP=2;.
95/005/17:15:00 SUBCARRIER: PD_NO=39; FREQ=21333;
BANDWIDTH=1.0; UPDATE_RATE=100; LOOP=2;.
95/005/17:15:00 SYMBOL: ES_NO=40; SYMBOL_RATE=500;
BANDWIDTH=1.0; UPDATE_RATE=100; LOOP=2;
CODE_FLAG=1;.
95/005/17:15:00 TELEM: BIT_RATE=10; MOD_INDEX=20;.
95/005/18:00:00 TELEM: MOD_INDEX=30;.
95/005/18:30:00 TELEM: BIT_RATE=20;.
95/005/19:00:00 TELEM: BIT_RATE=50; MOD_INDEX=60;.
95/005/20:00:00 END_OF_TRACK.

Figure 4.2: VOYAGER downlink telemetry pass LSOE

The overall flow of information in our approach to SOE driven automation is shown in Figure 4.3. In this approach, the project SOE is parsed into an LMCOA SOE (LSOE) which contains only that information relevant to the track of interest. The DSN planner uses the LSOE and the Project profile to determine the parameterized goals of the track. A sample set of track goals is shown below in Figure 4.4. This set of goals and a set of contextual facts derived from the LSOE are the planning inputs. The contextual facts are basically the planning system's representation of the LSOE information.

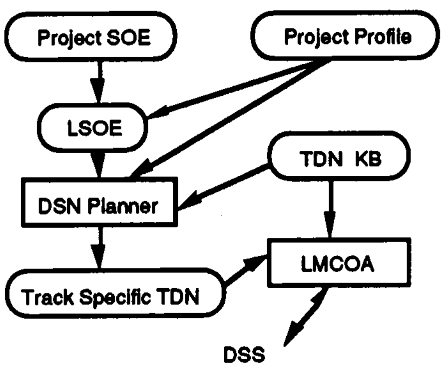

Figure 4.3: SOE Driven Automation Architecture

(PO_required track1)
(spacecraft_mode_changes track1)
(track_goal spacecraft_track telemetry track1)
(track_goal spacecraft_track downlink track1)
(track_goal decode_data)
(station-used track1 DSS13)

Figure 4.4: Example Tracking Goals

Using this problem specification, the DSN planner then uses task reduction planner (also called hierarchical task network or HTN) (Erol et al., 1994; Lansky, 1993) planning techniques to produce a parameterized track specific TDN to be used to conduct the track. The actual planner used to generate the TDN is a modified version of the task reduction planning component of the MVP system (Chien 1994). This track-specific TDN and the LSOE can then be used by the LMCOA system to operate the actual antennas to conduct the requested antenna track.

For example, consider the task reductions shown in Figure 4.5 and 4.6. As part of the pre-calibration procedure for the station DSS13 for a downlink type of track, there are a series of steps which must be taken. As Figure 4.5 shows, these tasks are inspecting the subsystems, connecting the subsystems, configuring the total power radiometer, loading antenna predicts, and configuring the receiver. Some of these tasks are composite tasks, and will be later expanded into more detailed tasks. For example, configuring the total power radiometer involves configuring the IF switch, configuring the UWC/TPR for pre-calibration, and performing the actual TPR pre-calibration. In the context of this specific track, the programmable oscillator (PO) has been requested, so that additional steps will be inserted into the TDN as shown in Figure 4.6. In the context of specific tracks, generic configuration blocks will be specialized as appropriate to the details of the track. For example, in the context of the VOYAGER track outlined above, the track type is downlink, telemetry, with symbol decoding requested. This will require a specific type of receiver configuration block. This process continues until all of the activity blocks in the TDN are operational - that is to say that known blocks in the TDN KB can be used to instantiate the activities. This fully instantiated TDN can then be used with the LSOE by LMCOA to perform the actual track.

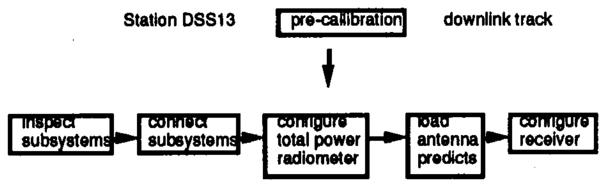

Figure 4.5: Sample Task Reduction

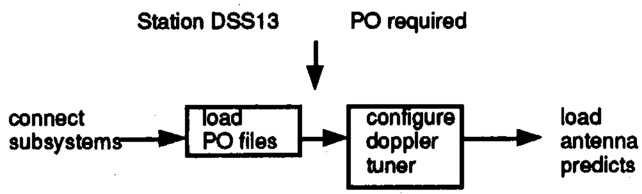

Figure 4.6: Augmentation of TDN Required by Additional Programmable Oscillator Track Goal

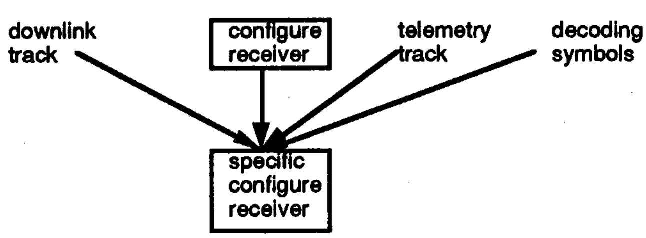

Figure 4.7: A Specialized Task Reduction

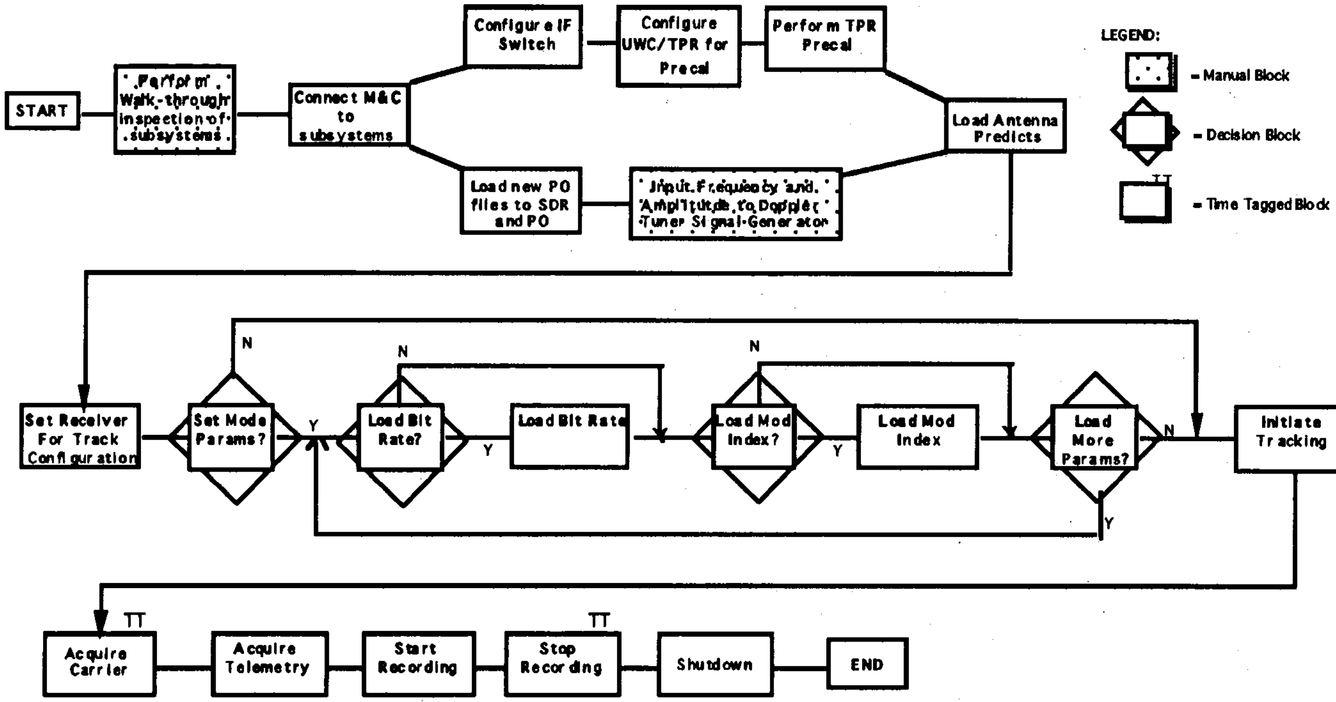

Figure 4.8: TDN for a VOYAGER telemetry downlink track using the 34-meter beam wave antenna

Considerable effort in computing the final TDN is devoted to determining the correct parameters for blocks in the TDN. For example, Figure 4.9 shows a configuration table used to determine the IF switch parameter for the TPR precalibration step. Depending on the communications bands used in the track, differing bands will assigned to each of the communications pathways in the UWC/TPR. Based on the bands being used, the TPR IF switch parameter is also determined. This parameter setting is also determined during the decomposition process, so a correctly parameterized TDN can be constructed.

	IF1	IF2	Parameter
S-band	*		1
X-band	*		1
Ka-band		*	2
Q-band		*	2
S&X-band	*	*	3
X&Ka-band	*	*	3

Figure 4.9: TPR IF Switch Parameter Determination

Thus, using these decomposition rules, a track-specific TDN can be derived. Figure 4.8 shows a TDN for a VOYAGER downlink telemetry track using the Programmable oscillator for the DSS13 DSN antenna station. This parameterized TDN can then be used by the LMCOA system to conduct the actual spacecraft downlink track.

5. Conclusions

This paper has described an effort to develop a plan construction and plan execution monitoring system to automate the operation of the communications link in NASA's Deep Space Network (DSN). The objectives of this effort are: (1) to archive knowledge of operations procedures; (2) to provide a capability to assist in execution monitoring; and (3) provide a capability for automatically constructing the plan to be executed and monitored based on an augmented spacecraft sequence of events (SOE). This paper described the application area of DSN antenna operations, described the plan representation of a Temporal Dependence Network (TDN), outlined the knowledge engineering tools being constructed to facilitate in representing DSN operations procedures, and described preliminary work in automated TDN generation.

6. References

(Chien, Hill, Fayyad, 1994) S. Chien, R. Hill, & K. Fayyad. "Why Real-world Planning is Difficult." Working Notes of the 1994 Fall Symposium on Planning and Learning: On to Real Applications, New Orleans, LA, November 1994, AAAI Press.

(Chien, 1994b) S. Chien, "Automated Synthesis of Complex Image Processing Procedures for a Large-scale Image Database," Proceedings of the First IEEE Int. Conf. on Image Processing, Austin, TX, November 1994, Vol 3, pp. 796-800.

(Erol et. al. 1994) K. Erol, J. Hendler, and D. Nau, "UMCP: A Sound and Complete Procedure for Hierarchical Task Network Planning," *Proceedings of the Second International Conference on AI Planning Systems*, Chicago, IL, June 1994, pp. 249-254.

(Fayyad & Cooper, 1992) Fayyad, K. E. and L. P. Cooper. "Representing Operations Procedures Using Temporal Dependency Networks," SpaceOPS '92, Pasadena, CA, November 1992.

(Fayyad et al., 1993) K. Fayyad, R. Hill, and E.J. Wyatt. "Knowledge Engineering for Temporal Dependency Networks as Operations Procedures." Proceedings of AIAA Computing in Aerospace 9 Conference, October 19-21, 1993, San Diego, CA.

(Lansky, 1993) A. Lansky, "Localized Planning with Diverse Plan Construction Methods," Technical Report FIA-93-17, NASA Ames Research Center, June 1993.

(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 of the 1994 AAAI Fall Symposium on Planning and Learning: On to Real Applications, New Orleans, LA, 1994.

(Hill&Johnson, 1994) R. Hill and W.L. Johnson. "Situated Plan Attribution for Intelligent Tutoring." Proceedings of the Twelfth National Conference on Artificial Intelligence, July 31-August 4, 1994, Seattle, WA.

(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.

(Zorman, 1994) Zorman, Lorna. "Requirements Envisaging by Utilizing Scenarios." Ph.D. Diss., in preparation, University of Southern California, 1994.