N88-17246

ROBOTIC PLANNER EXPERT SYSTEM (RPLANES)

Ervin O'Neal Grice

Artificial Intelligence Section
Mission Planning And Analysis Division
Johnson Space Center
Houston, Texas 77058

ABSTRACT

The Artificial Intelligence Section of the Mission Planning and Analysis Division of the Johnson Space Center has developed a prototype of an expert system for robotic planning. A robot is given a high-level goal to perform an action (i.e. swap, adjust, or stow) on a component unit of an object such as a satellite and the Robotic Planner Expert System (RPLANES) generates the necessary goals for arm actions. RPLANES is designed using the Inference Corporation Automated Reasoning Tool (ART) development tool. It resides on a SYMBOLICS 3670. This paper describes RPLANES and its evolution.

1. INTRODUCTION

Robotic technology is an anticipated essential during the Space Station operational era. Robots will be given high-level goals to perform that will be accomplished by executing a series of lower-level goals. These lower-level goals and the order of performance are derived by a planner expert system. The initial development effort for the prototype detailed in this paper is attributed to the Sequence Automation Research Group of the Jet Propulsion Laboratory of the California Institute of Technology. The forty-eight rules expert system had a limited capability of planning the two arm actions of a robot to swap a component unit from a simple satellite structure (Figure 1) by dismantling all components of the structure. Enhancement of this expert system has evolved to eighty-two rules to:

a. Performing a swap, adjust, or stow goal.
b. Dismantling only those component units required to reach the target component.
c. Defining a more complex satellite structure (Figure 2).

2. INITIALIZATION

2.1 Define The Architectural Structure Of The Serviceable Object

Using the Inference Corporation Automated Reasoning Tool (ART) concept of schemata, the architectural structure of the serviceable object is defined. The relations "comes-before" and its inverse "comes-after" are defined and employed to define the architectural integrity of all components within the structure. Figure 2 is a diagram of the Laser Battlestation used in this prototype and Figure 3 details the definitions of the "door1" architecture using schemata.

2.2 Define The Serviceable Component Units

Each serviceable component unit of the object is defined with a collection of attribute slots (Figure 4). Specific actions that can be performed on the component and the action by the robot is defined. In addition to defining the component units of the object, replacement units used in a swap action are defined having a current location of a "tool-box" or "storage-bin" (Figure 5).

2.3 Define The Initial State Of The Robot

The initial state of the robot is defined using schemata and attribute slots (Figure 6). The left and right arms are statused "free" and the left and right arms are defined as empty by assigning tool values of "null-tool-1" and "null-tool-2" to each arm respectively.

2.4 Define The Action Scripts

An action on a component unit can be described as a sequence of sub-actions. For example a swap action of a component unit involves:

a. Rendezvousing with the component unit to be swapped.
b. Removing the component unit.
c. Stowing the component unit.
d. Unstowing the replacement component unit.
e. Replacing with the new component unit.

If a node is used to represent each sub-action, this sequence of nodes represents the action script. Each node in the script and its order within the script is defined using schemata and attribute slots (Figure 7).

3. USER INTERFACE

RPLANES employs mouse and menus for user interface (Figure 8). The serviceable object definition, type of action to perform, and the component unit(s) involved, are defined by the user. Any conflicts in these selections are determined by the expert system and redefinition is required.

4. GENERATING THE SUB-GOALS

4.1 Logically Dismantle The Required Component Units

Based upon the architectural structure of the object defined during initialization, all serviceable component units that "comes-before" the target component unit must be dismantled. A node representing each component unit along with attribute slots are placed in a knowledge base.

4.2 Retrieve The Action Script

Following the logical dismantling of the necessary component units, the specified action script is retrieved. A node and attribute slots representing the sequence of steps in the specified action script are placed in the knowledge base. A "comes-after" relationship attribute is associated with this set of nodes to denote the order of the action on the target component unit with respect to the dismantling of preceding units.

4.3 Logically Reassemble The Required Component Units

Once the action script has been logically performed on the target component unit, the object must be reassembled. Nodes representing each unit and the attribute slots detailing each node are also placed in the knowledge base. Reassembling the component units is performed in reverse with respect to dismantling.

4.4 Determine The Robot's Arm Actions

RPLANES assumes the robot has two arms that can work in parallel and/or serially. Progressing through the ordered nodes constructed in the knowledge base, the specific arm to be used by the robot, the tool required, and the component unit to service is determined.

5. OUTPUT

Using the nodes of attribute slots generated in the knowledge base above, the robot primitives (commands) are generated. The output is displayed to CRT (Figure 9).

6. CONCLUSION

Development of this prototype demonstrates the planner expert system functional capabilities. Applications employing graphics and hardware with the capability to input and execute these generated primitives are potential extensions.

7. ACKNOWLEDGEMENTS

I would like to thank Mark Rokey of the Sequence Automation Research Group of the Jet Propulsion Laboratory for the baseline concepts expanded within the expert system and Dr. Joseph Giarratano of the University of Houston-Clear Lake for the design of the laser battlestation used in the prototype expert system.

8. REFERENCE

1. Inference Corporation, "AUTOMATED REASONING TOOL REFERENCE MANUAL," 5300 W. Century Blvd., Los Angeles, Ca. 90045.

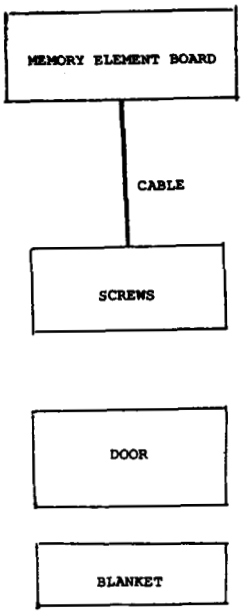

FIGURE - 1 SIMPLE SATELLITE

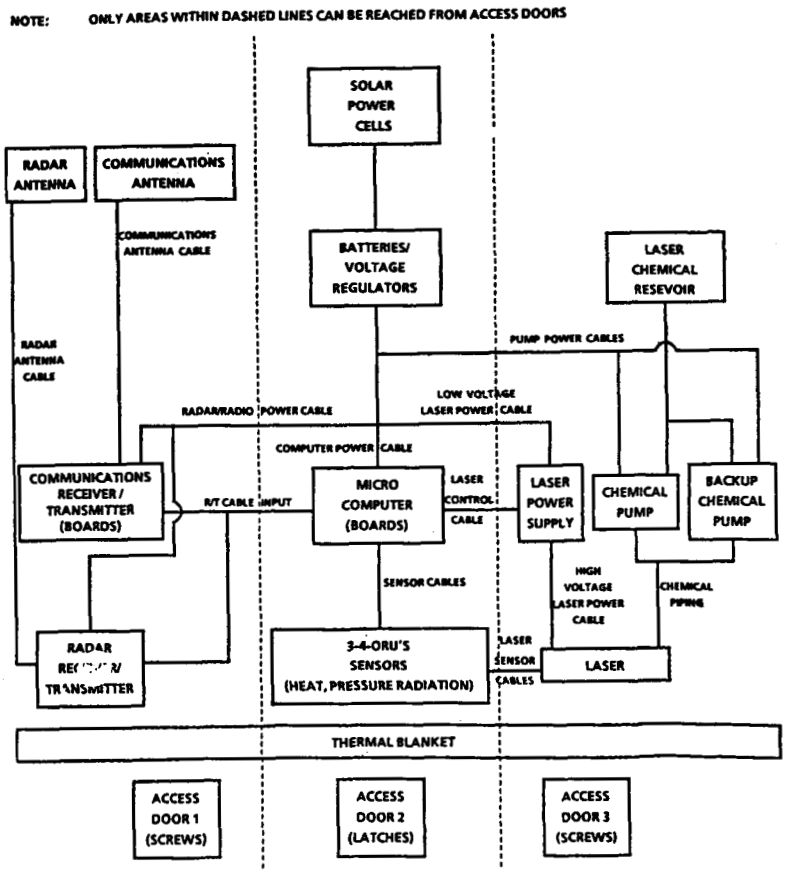

FIGURE - 2 LASER BATTLESTATION

```
;;;                                    DOOR1

(defschema laser-door1-part1 ""
  (structure-type laser-station)
  (part-type screw1)
  (comes-before laser-door1-part1)
  (instance-of part-item))

(defschema laser-door1-part1 ""
  (structure-type laser-station)
  (part-type screw2)
  (comes-before laser-door1-part1)
  (instance-of part-item))

(defschema laser-door1-part1 ""
  (structure-type laser-station)
  (part-type door1)
  (comes-before laser-door1-part2)
  (instance-of part-item))

(defschema laser-door1-part2 ""
  (structure-type laser-station)
  (part-type blanket)
  (comes-before laser-door1-part3)
  (comes-before laser-door1-part4)
  (comes-before laser-door1-part5)
  (instance-of part-item))

(defschema laser-door1-part3 ""
  (structure-type laser-station)
  (part-type radar-antenna-cable)
  (comes-before laser-door1-part4)
  (comes-before laser-door1-part6)
  (comes-before laser-door1-part7)
  (comes-before laser-door1-part8)
  (instance-of part-item))

(defschema laser-door1-part4 ""
  (structure-type laser-station)
  (part-type radar-power-cable)
  (comes-before laser-door1-part6)
  (comes-before laser-door1-part8)
  (comes-before laser-door1-part9)
  (instance-of part-item))

(defschema laser-door1-part5 ""
  (structure-type laser-station)
  (part-type radar-rt-cable)
  (comes-before laser-door1-part4)
  (comes-before laser-door1-part6)
  (comes-before laser-door1-part8)
  (instance-of part-item)).

(defschema laser-door1-part6 ""
  (structure-type laser-station)
  (part-type radar-receiver-transmitter)
  (comes-before laser-door1-part8)
  (comes-after laser-door1-part2)
  (instance-of part-item))

(defschema laser-door1-part7 ""
  (structure-type laser-station)
  (part-type radar-antenna)
  (comes-after laser-door1-part6)
  (comes-after laser-door1-part8)
  (instance-of part-item))

(defschema laser-door1-part8 ""
  (structure-type laser-station)
  (part-type comm-receiver-transmitter)
  (comes-after laser-door1-part6)
  (instance-of part-item))

(defschema laser-door1-part9 ""
  (structure-type laser-station)
  (part-type comm-antenna-cable)
  (comes-before laser-door1-part8)
  (comes-before laser-door1-part10)
  (instance-of part-item))

(defschema laser-door1-part10 ""
  (structure-type laser-station)
  (part-type comm-antenna)
  (comes-after laser-door1-part8)
  (instance-of part-item))
```

FIGURE - 3 LASER BATTLESTATION SCHEMATA DEFINITION

```
(defschema door1 ""
  (family door)
  (first-choice-arm right-arm)
  (second-choice-arm left-arm)
  (status attached)
  (current-location laser-station)
  (actions-by-this-object-slot (slot-how-many multiple-values)
              (detach open)
              (attach close)
              (adjust-in close)
              (adjust-out open)
              (stow stow)
              (unstow unstow)
              (replace attach)
              (remove detach))
  (state-or-location-after-action-slot (slot-how-many multiple-values)
               (detach laser-station)
               (attach laser-station)
               (adjust-in closed)
               (adjust-out opened)
               (stow rack)
               (unstow freespace)
               (replace laser-station)
               (remove rack))
  (tool (slot-how-many multiple-values) null-tool-1 null-tool-2))

(defschema screw-characteristics ""
  (family screw)
  (actions-by-this-object-slot (slot-how-many multiple-values)
              (detach unscrew)
              (remove unscrew)
              (adjust-in screw)
              (adjust-out unscrew)
              (stow slide-in)
              (unstow slide-out)
              (attach screw)
              (replace screw))
  (state-or-location-after-action-slot (slot-how-many multiple-values)
               (detach outset)
               (remove toolbox)
               (adjust-in inset)
               (adjust-out outset)
               (stow tool-box)
               (unstow freespace)
               (attach inset)
               (replace inset))
  (tool screwdriver))

(defschema screw-1 ""
  (first-choice-arm right-arm)
  (second-choice-arm left-arm)
  (instance-of screw-characteristics)
  (current-location inset)
  (status attached))
```

FIGURE - 4 SERVICEABLE COMPONENT DEFINITION

```
(defschema new-door ""
  (family door)
  (first-choice-arm right-arm)
  (second-choice-arm left-arm)
  (status attached)
  (current-location rack)
  (actions-by-this-object-slot (slot-how-many multiple-values)
              (replace attach)
              (adjust-in push)
              (adjust-out pull)
              (stow stow)
              (unstow unstow)
              (remove detach))
  (state-or-location-after-action-slot (slot-how-many multiple-values)
               (replace satellite)
               (adjust-in rack)
               (adjust-out rack)
               (stow rack)
               (unstow freespace)
               (remove satellite))
  (tool (slot-how-many multiple-values) null-tool-1 null-tool-2))
```

```
(defschema new-cable ""
  (family cable)
  (first-choice-arm left-arm)
  (second-choice-arm right-arm)
  (status attached)
  (current-location toolbox)
  (actions-by-this-object-slot (slot-how-many multiple-values)
              (replace plug-in)
              (remove slide-out)
              (stow fold)
              (unstow slide-out))
  (state-or-location-after-action-slot (slot-how-many multiple-values)
               (replace satellite)
               (remove satellite)
               (stow tool-box)
               (unstow freespace))
  (tool (slot-how-many multiple-values) null-tool-1 null-tool-2))
```

FIGURE - 5 REPLACEMENT UNITS DEFINITION

```
(defschema script "Base definition of a script schema"
  (script-type)
  (script-object)
  (script-state)
  (comes-before))

;;;*****************************************************************
;;;  Here is the definition of the network of nodes which describe
;;;  a SWAP action.  This is typical of what will be many such
;;;  "script-like ACTION networks" in the final system.
;;;*****************************************************************

(defschema swap-source ""
  (instance-of script)
  (script-type swap)
  (script-object source)
  (script-state rendezvous)
  (comes-before swap-1))

(defschema swap-1 ""
  (instance-of script)
  (script-object swappee)
  (script-type swap)
  (script-state remove)
  (comes-before swap-2))

(defschema swap-2 ""
  (instance-of script)
  (script-object stowee)
  (script-type swap)
  (script-state stow)
  (comes-before swap-3))

(defschema swap-3 ""
  (instance-of script)
  (script-object unstower)
  (script-type swap)
  (script-state unstow)
  (comes-before swap-4))

(defschema swap-4 ""
  (instance-of script)
  (script-object swapper)
  (script-type swap)
  (phase 3mark)
  (script-state replace)
  (comes-before swap-sink))

(defschema swap-sink ""
  (instance-of script)
  (script-type swap)
  (script-state rendezvous)
  (script-object sink))
```

FIGURE - 7 ACTION SCRIPT DEFINITION

```
(defschema left-arm ""
  (status free))

(defschema right-arm ""
  (status free))

(defschema screwdriver ""
  (status free))

(defschema null-tool-1 ""
  (held-by right-arm)
  (status free))

(defschema null-tool-2 ""
  (held-by left-arm)
  (status free))
```

FIGURE - 6 INITIAL ROBOT STATE

```
Primitives developed for robot

(TASK REQUESTED: STOW THE LASER-STATION RADAR-RT-CABLE)

(MOVE RIGHT-ARM TO TOOLBOX)
(REPLACE NULL-TOOL-1)
(ACQUIRE SCREWDRIVER)
(MOVE RIGHT-ARM TO SCREW1)
(UNSCREW SCREW1)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-IN SCREW1 IN TOOL-BOX)
(MOVE RIGHT-ARM TO SCREW2)
(UNSCREW SCREW2)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-IN SCREW2 IN TOOL-BOX)
(MOVE RIGHT-ARM TO TOOLBOX)
(REPLACE SCREWDRIVER)
(ACQUIRE NULL-TOOL-1)
(MOVE RIGHT-ARM TO DOOR1)
(OPEN DOOR1)
```

```
Primitives developed for robot

(TASK REQUESTED: STOW THE LASER-STATION RADAR-RT-CABLE)  page 2

(MOVE RIGHT-ARM TO BLANKET)
(REMOVE BLANKET)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-IN BLANKET IN TOOL-BOX)
(MOVE LEFT-ARM TO RADAR-RECEIVER-TRANSMITTER RADAR-RT-CABLE CONNECTION)
(UNPLUG RADAR-RT-CABLE)
(MOVE LEFT-ARM TO COMM-RECEIVER-TRANSMITTER RADAR-RT-CABLE CONNECTION)
(UNPLUG RADAR-RT-CABLE)
(MOVE LEFT-ARM TO RADAR-RT-CABLE)
(DETACH RADAR-RT-CABLE)
(MOVE LEFT-ARM TO TOOL-BOX)
(FOLD RADAR-RT-CABLE)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-OUT BLANKET)
(MOVE RIGHT-ARM TO SATELLITE)
(REPLACE BLANKET TO SATELLITE)
```

```
Primitives developed for robot

(TASK REQUESTED: STOW THE LASER-STATION RADAR-RT-CABLE)  page 3

(MOVE RIGHT-ARM TO DOOR1)
(CLOSE DOOR1)
(MOVE RIGHT-ARM TO TOOLBOX)
(REPLACE NULL-TOOL-1)
(ACQUIRE SCREWDRIVER)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-OUT SCREW1)
(MOVE RIGHT-ARM TO INSET)
(SCREW SCREW1 TO INSET)
(MOVE RIGHT-ARM TO TOOL-BOX)
(SLIDE-OUT SCREW2)
(MOVE RIGHT-ARM TO INSET)
(SCREW SCREW2 TO INSET)
```

FIGURE - 9 SAMPLE OUTPUT

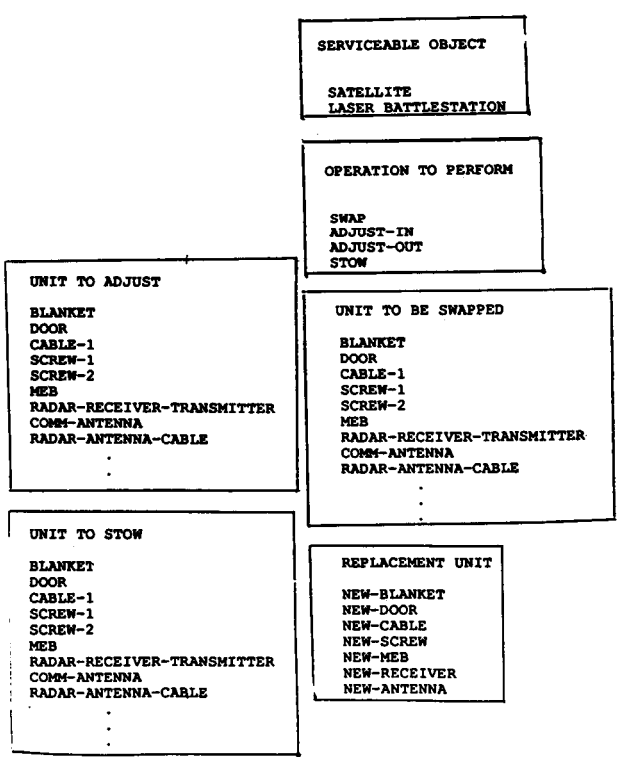

FIGURE - 8 USER INTERFACE MENUS