# AI Planning Applications

## Plan Execution
## Practical Applications of AI Planners

# AI Planning Applications

- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

# AI Planning Applications

- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

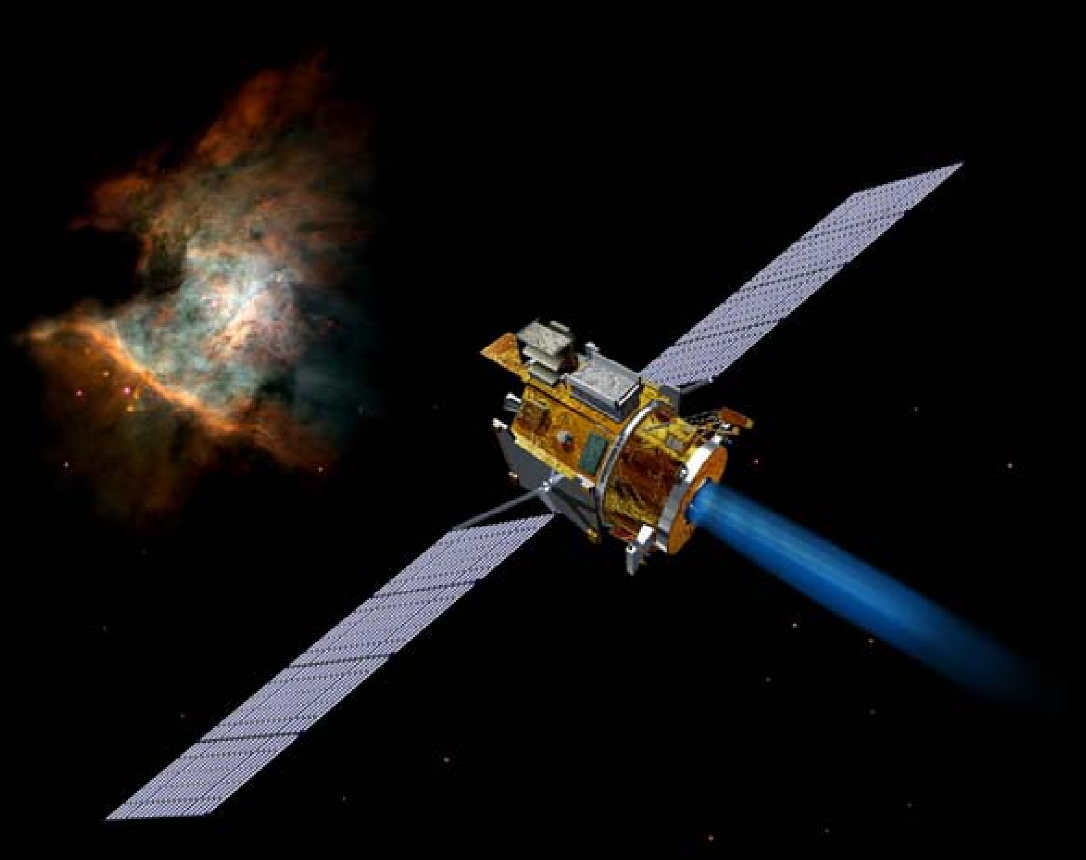
# Plan Execution & Monitoring

- STRIPS – PLANEX – Fikes, Hart & Nilsson 1971
- NOAH – Engineer’s Apprentice – Sacerdoti 1977
- Nonlin – Goal Structure Condition Monitoring – Tate 1984
- SIPE – Recovering from Execution Errors – Wilkins 1986
- PRS – Procedural Reasoning – Georgeff & Lansky 1986
- IPEM – Integrated Planning & Execution Monitoring – Ambros-Ingerson & Steel 1987
- RAP – Reactive Plan Executor – Firby 1987
- O-Plan – Repairing Plans – Drabble et al. 1997

# AI Planning Applications

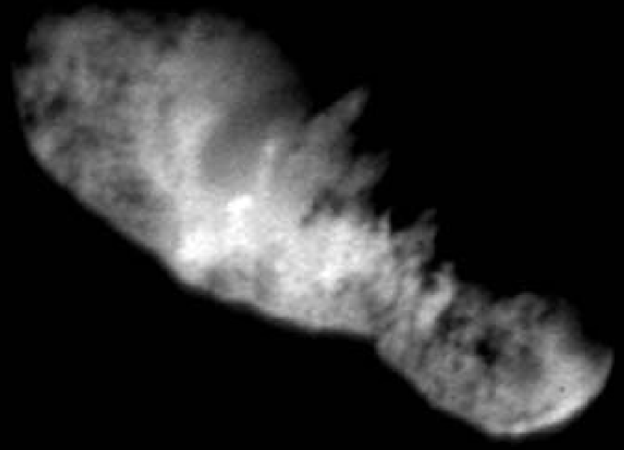
- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

# Deep Space 1 – 1998-2001



Photos: NASA

# DS 1 at Comet Borrelly



Photos: NASA

# DS1 Domain Requirements

Achieve diverse goals on real spacecraft:

- **High Reliability**
  - single point failures
  - multiple sequential failures
- **Tight resource constraints**
  - resource contention
  - conflicting goals
- **Hard-time deadlines**
- **Limited Observability**
- **Concurrent Activity**

# DS1 Remote Agent Approach

- **Constraint-based planning and scheduling**
  - supports goal achievement, resource constraints, deadlines, concurrency
- **Robust multi-threaded execution**
  - supports reliability, concurrency, deadlines
- **Model-based fault diagnosis and reconfiguration**
  - supports limited observability, reliability, concurrency
- **Real-time control and monitoring**

# DS1 Levels of Autonomy

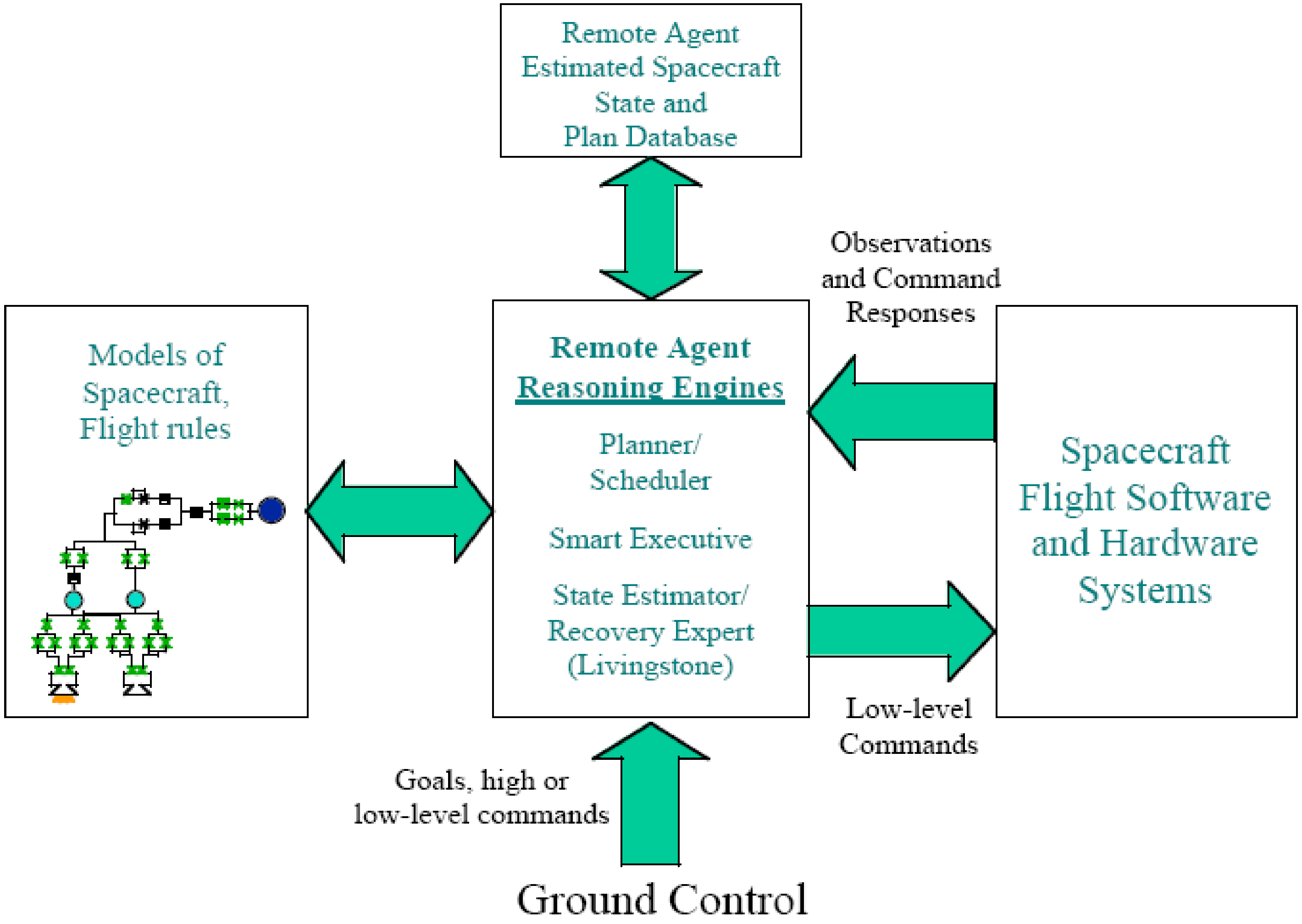
Listed from least to most autonomous mode:

1. single low-level real-time command execution
2. time-stamped command sequence execution
3. single goal achievement with auto-recovery
4. model-based state estimation & error detection
5. scripted plan with dynamic task decomposition
6. on-board back-to-back plan generation, execution, & plan recovery

# DS 1 Levels of Autonomy

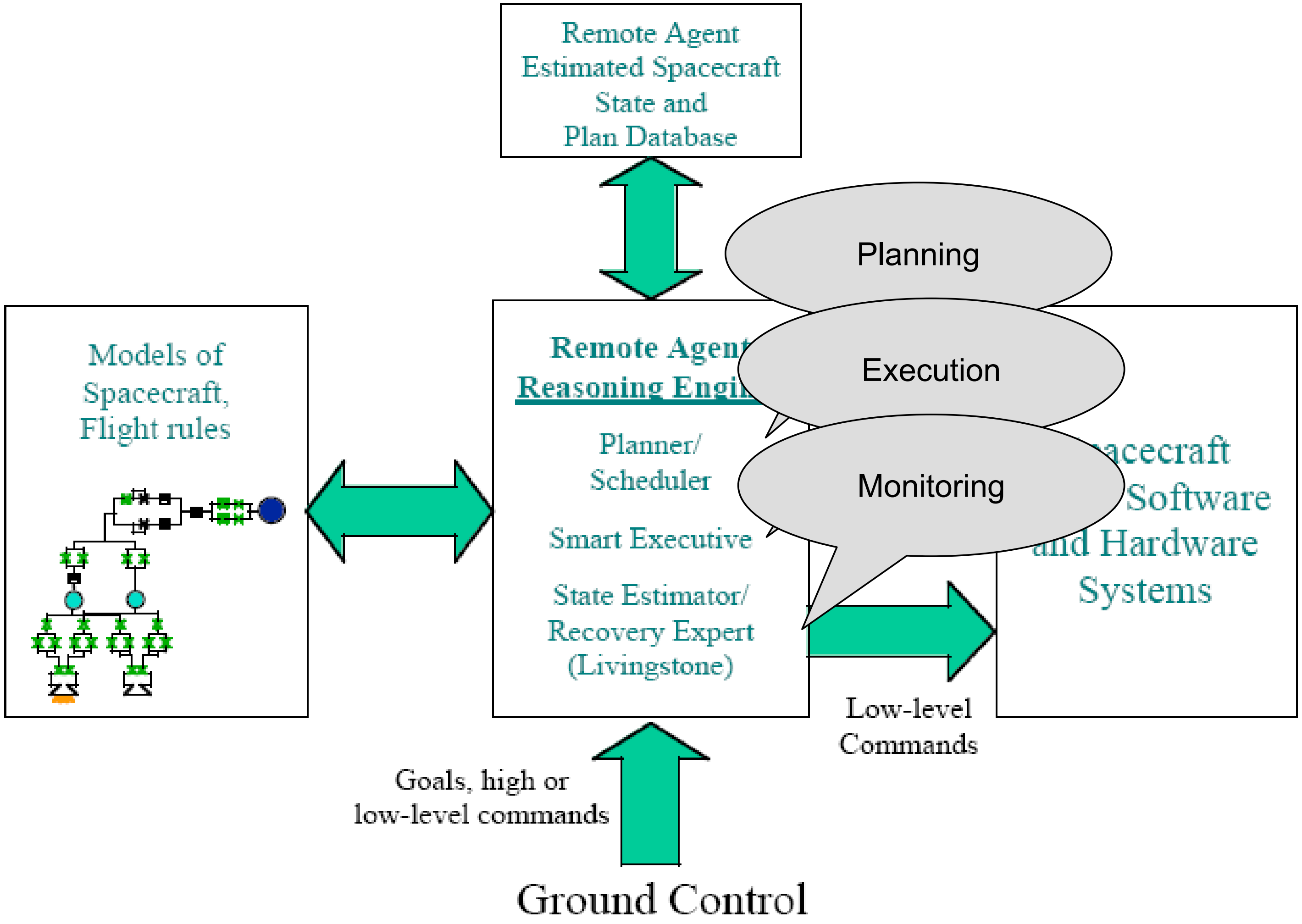
| Level | Ground System | On-board Planner/Scheduler | On-board EXEC |
|---|---|---|---|
| 1 | Prepare real-time commands | None | None (executed without using EXEC) |
| 2 | Prepare sequence | None | Execute sequence |
| 3 | Prepare plan, upload to EXEC as script | None | Execute plan in “Scripted Mode” |
| 4 | Prepare plan, upload to Planner as goals | Confirm and pass through to Planner | Execute plan in “Planner Mode” |
| 5 | Prepare plan, including some unexpanded goals | Complete the plan | Execute plan |
| 6 | Define goals | Prepare plans | Execute plan |

# DS 1 Systems



Remote Agent Estimated Spacecraft State and Plan Database
Observations and Command Responses
Models of Spacecraft, Flight rules
Remote Agent Reasoning Engines
Planner/ Scheduler
Smart Executive
State Estimator/ Recovery Expert (Livingstone)
Spacecraft Flight Software and Hardware Systems
Low-level Commands
Goals, high or low-level commands
Ground Control

# DS 1 Systems



Remote Agent
Estimated Spacecraft
State and
Plan Database
Planning
Execution
Monitoring
Models of
Spacecraft,
Flight rules
Remote Agent
Reasoning Engi
Planner/
Scheduler
Smart Executive
State Estimator/
Recovery Expert
(Livingstone)
acecraft
Software
and Hardware
Systems
Low-level
Commands
Goals, high or
low-level commands
Ground Control

# DS1 RAX Functionality

***Planner Scheduler/Mission Manager (PS/MM)***

- generate plans on-board the spacecraft
- reject low-priority unachievable goals
- re-plan following a simulated failure
- enable modification of mission goals from ground

***Executor (EXEC)***

- provide a low-level commanding interface
- initiate on-board planning
- execute plans generated both on-board and on the ground
- recognize and respond to plan failure
- maintain required properties in the face of failures

***Mode Identification and Recovery (MIR)***

- confirm executive command execution
- demonstrate model-based failure detection, isolation, and recovery
- demonstrate ability to update on-board state via ground commands

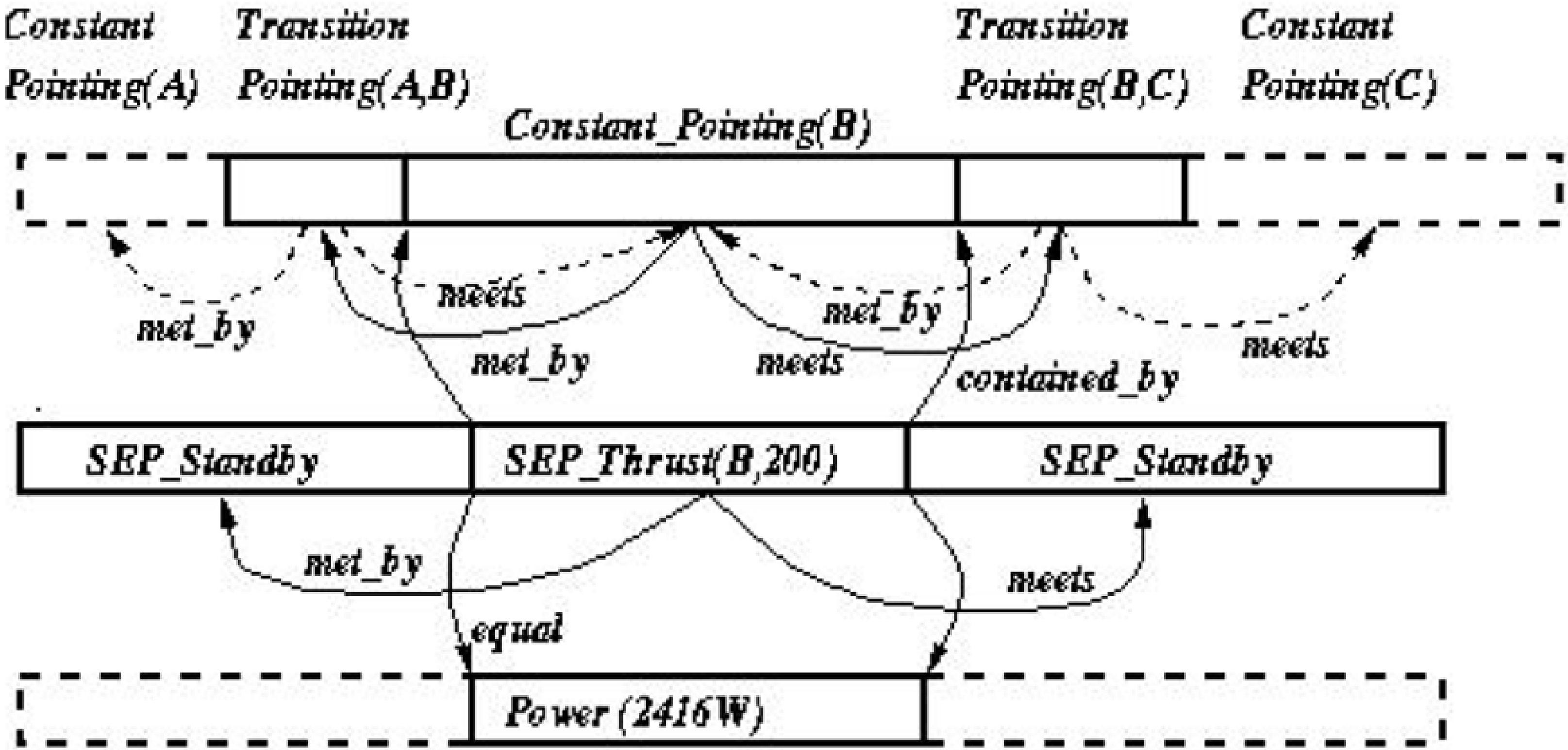
# DS1 Diversity of Goals

- **Final state goals**
  - “Turn off the camera once you are done using it”
- **Scheduled goals**
  - “Communicate to Earth at pre-specified times”
- **Periodic goals**
  - “Take asteroid pictures for navigation every 2 days for 2 hours”
- **Information-seeking goals**
  - “Ask the on-board navigation system for the thrusting profile”
- **Continuous accumulation goals**
  - “Accumulate thrust with a 90% duty cycle”
- **Default goals**
  - “When you have nothing else to do, point HGA to Earth”

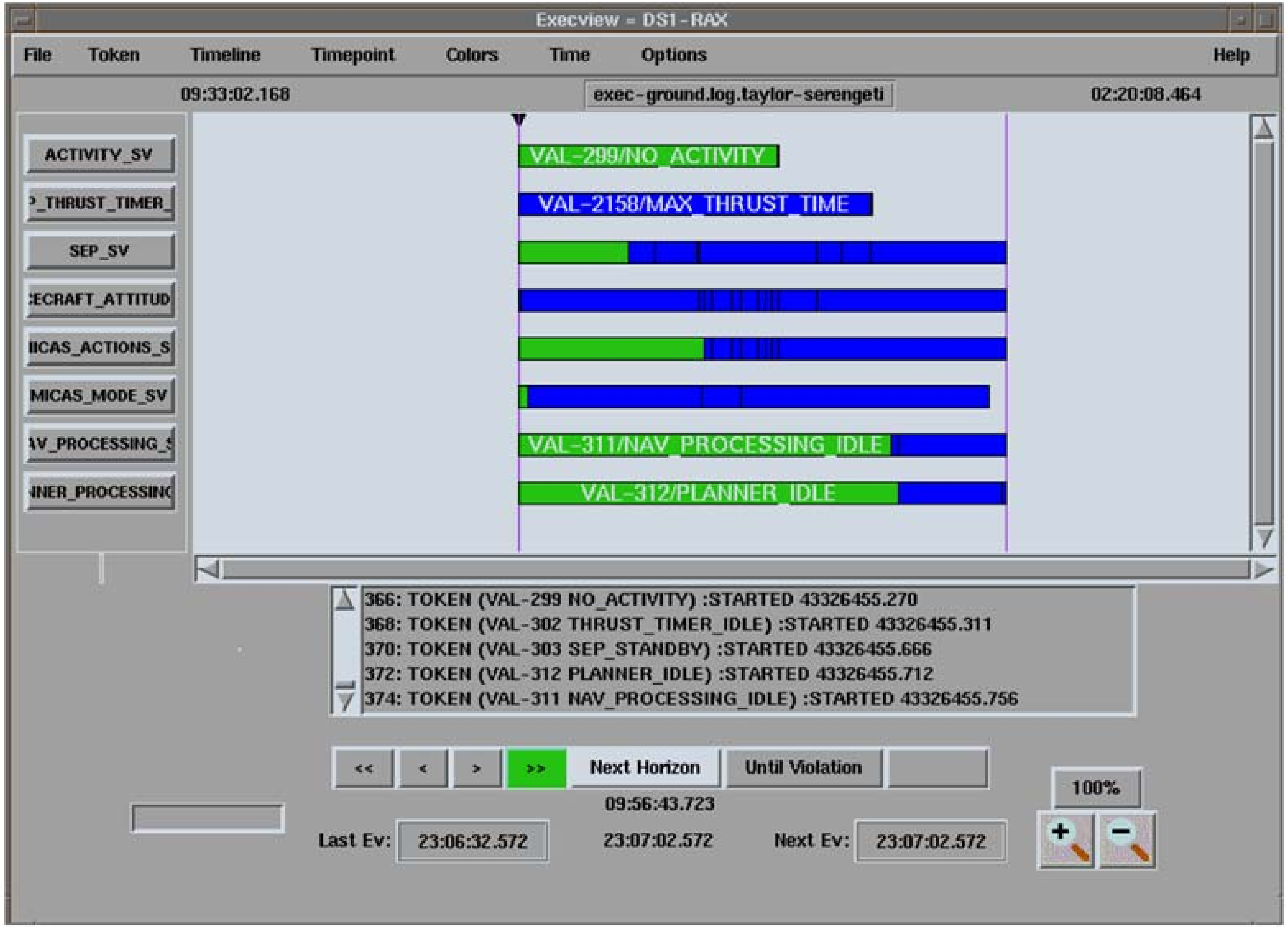
# DS1 Diversity of Constraints

- **State/action constraints**
  - “To take a picture, the camera must be on.”
- **Finite resources**
  - power
- **True parallelism**
  - the ACS loops must work in parallel with the IPS controller
- **Functional dependencies**
  - “The duration of a turn depends on its source and destination.”
- **Continuously varying parameters**
  - amount of accumulated thrust
- **Other software modules as specialized planners**
  - on-board navigator

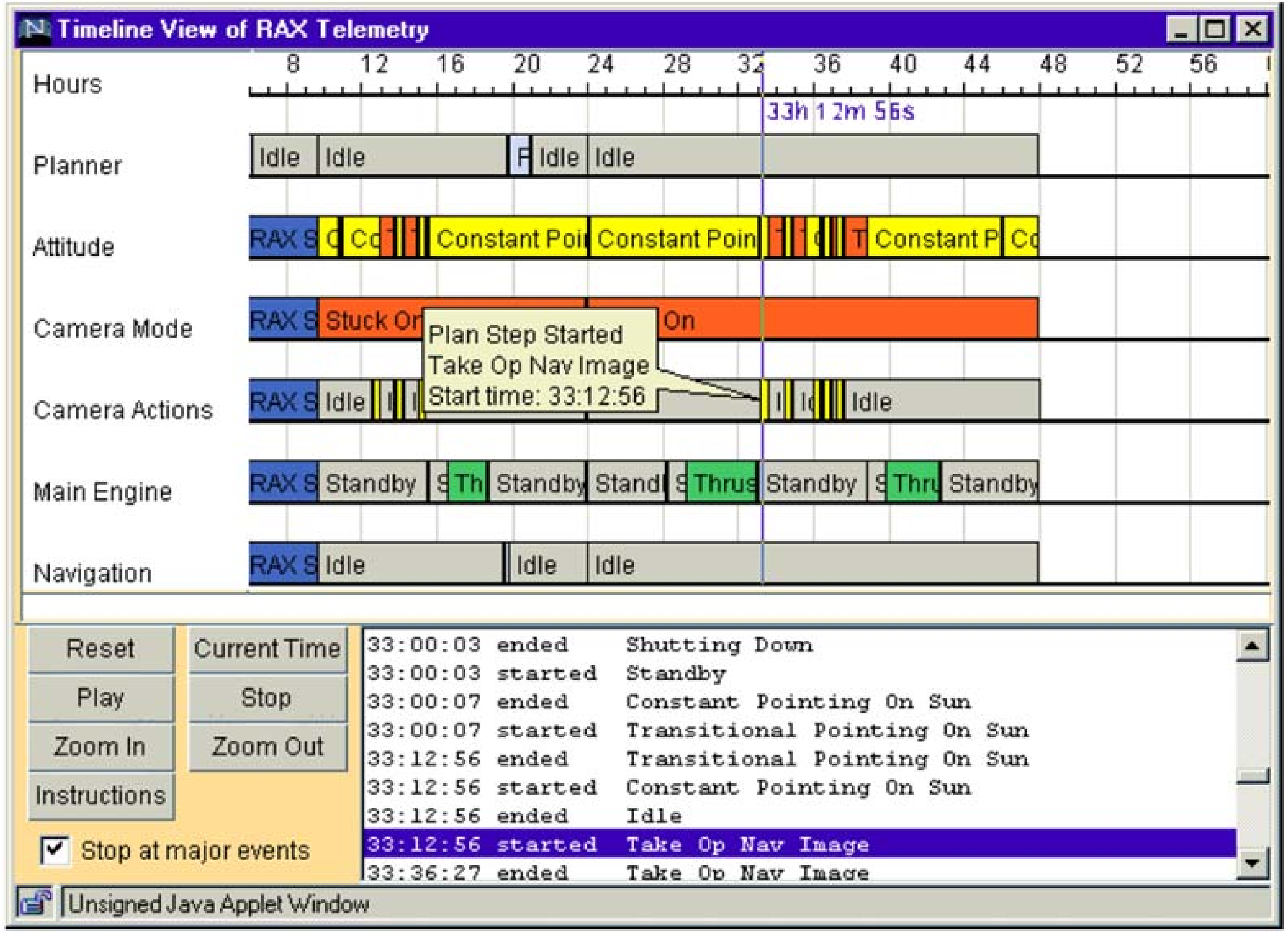
# DS1 Plan Fragment



# DS1 Remote Agent EXEC Status Tool



# DS1 Remote Agent Ground Tools



# DS1 RAX – Flight Experiments 17$^{th}$ – 21$^{st}$ May 1999

- RAX was activated and controlled the spacecraft autonomously. Some issues and alarms did arise:
- Divergence of model predicted values of state of Ion Propulsion System (IPS) and actual observed values – due to infrequency of real time monitor updates.
- EXEC deadlocked in use. Problem diagnosed and fix designed but not uploaded to DS1 for fears of safety of flight systems.
- Condition had not appeared in thousands of ground tests indicating needs for formal verification methods for this type of safety/mission critical software.
- Following later experiments, RAX was deemed to have achieved its aims and objectives.

# DS1 RAX Experiment 2 Day Scenario

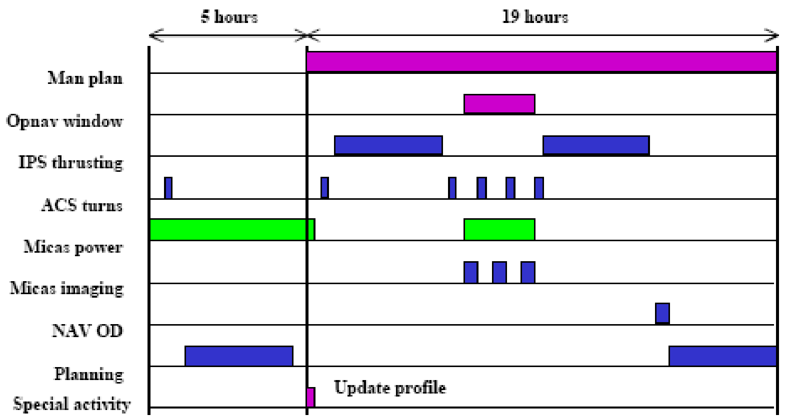



Figure 1415: First day of the 2-day scenario

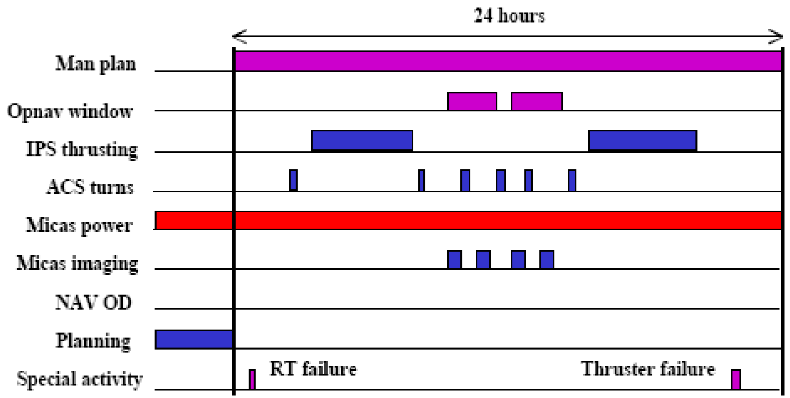



Figure 1617: Second day of the 2-day scenario

# DS1 RAX Summary
# Objectives and Capabilities

## *Validation Objectives*

- ✓ •Initiate and generate flexible plans on-board
- ✓ •Reject low-priority, unachievable goals
- ✓ •Execute plans generated both on-board and from Ground
- ✓ •Confirm execution of commands
- ✓ •Demonstrate model-based failure detection and recovery
- ✓ •Maintain required spacecraft states in the face of failures
- ✓ •Re-plan following a failure
- ✓ •Generate back-to-back plans
- ✓ •Modify mission goals from Ground
- ✓ •Execute low-level commands from Ground
- ✓ •Update estimated spacecraft state database from Ground

## *Capabilities*

•Robust Goal-based commanding

- Planner expands high-level goals into flexible plans
- Executive decomposes plans into low-level spacecraft commands and monitors that the states commanded to are achieved and maintained

•Fail-operational model-based fault recovery

- Livingstone identifies faults and suggests recoveries that the Executive uses to continue plan execution
- If necessary, Executive requests the Planner to generate a new plan in light of failure

# DS1 RAX Features

- AI planner outer level with re-planning capability
- Detailed constraint handling (e.g. time and resources)
- Integration with system diagnostics and analysis
- Integration of plan execution and monitoring
- Rich knowledge modelling languages
- Comprehensive user interfaces

# AI Planning Applications

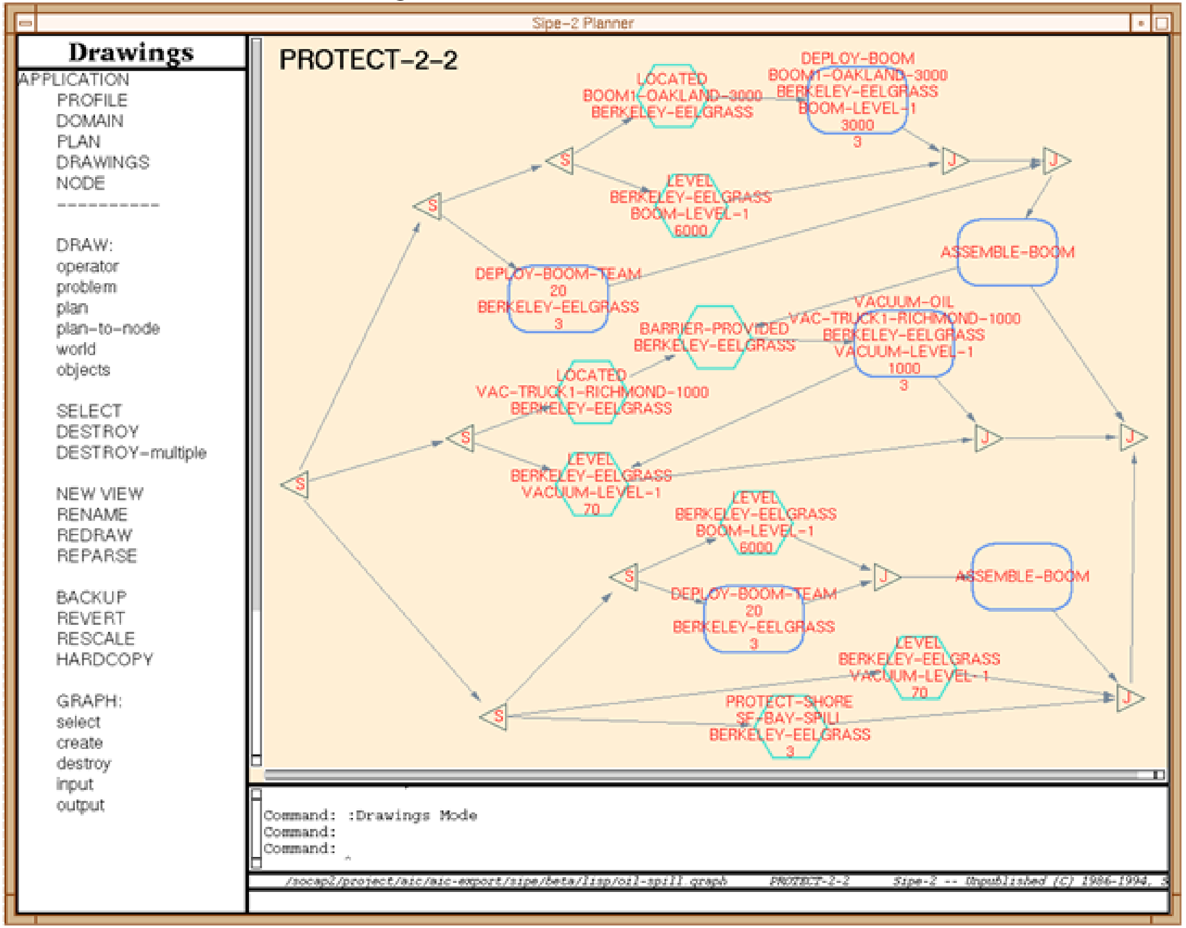
- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

# Practical AI Planners

| *Planner* | *Reference* | *Applications* |
|---|---|---|
| STRIPS | Fikes & Nilsson 1971 | Mobile Robot Control, etc. |
| HACKER | Sussman 1973 | Simple Program Generation |
| NOAH | Sacerdoti 1977 | Mechanical Engineers Apprentice Supervision |
| NONLIN | Tate 1977 | Electricity Turbine Overhaul, etc. |
| NASL | McDermott 1978 | Electronic Circuit Design |
| OPM | Hayes-Roth & Hayes-Roth 1979 | Journey Planning |
| ISIS-II | Fox et. al. 1981 | Job Shop Scheduling (Turbine Production) |
| MOLGEN | Stefik 1981 | Experiment Planning in Molecular Genetics |
| DEVISER | Vere 1983 | Spacecraft Mission Planning |
| FORBIN | Miller et al. 1985 | Factory Control |
| SIPE-2 | Wilkins 1988 | Oil Spill Response, Military Planning, etc. |
| O-Plan | Currie & Tate 1991 | Search and Rescue, Spacecraft Operations, etc. |
| SHOP/SHOP-2 | Nau et al. 1999 | Evacuation Planning, Forest Fires, Bridge Baron, etc. |
| I-X/I-Plan | Tate et al. 2000 | Emergency Response, etc. |

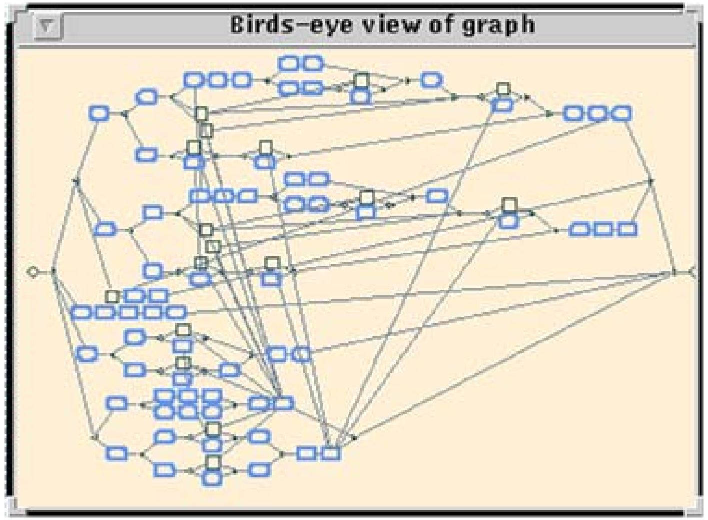
# Practical Applications of AI Planners

## SIPE-2: System for Interactive Planning and Execution





David Wilkins, AI Center
SRI International

# Practical Applications of AI Planners

## SIPE-2 Features

- Supports interactive planning, allowing humans and the system to cooperate in mixed-initiative planning
- Efficiently reasons about actions to generate a novel sequence of actions that responds precisely to the situation at hand
- Supports the giving of advice to the planner
- Plans hierarchically at different levels of abstraction
- Replans during execution
- Generates parallel plans (allowing multiple agents)
- Posts constraints and reasons about resources
- Interacts with users through a graphical interface

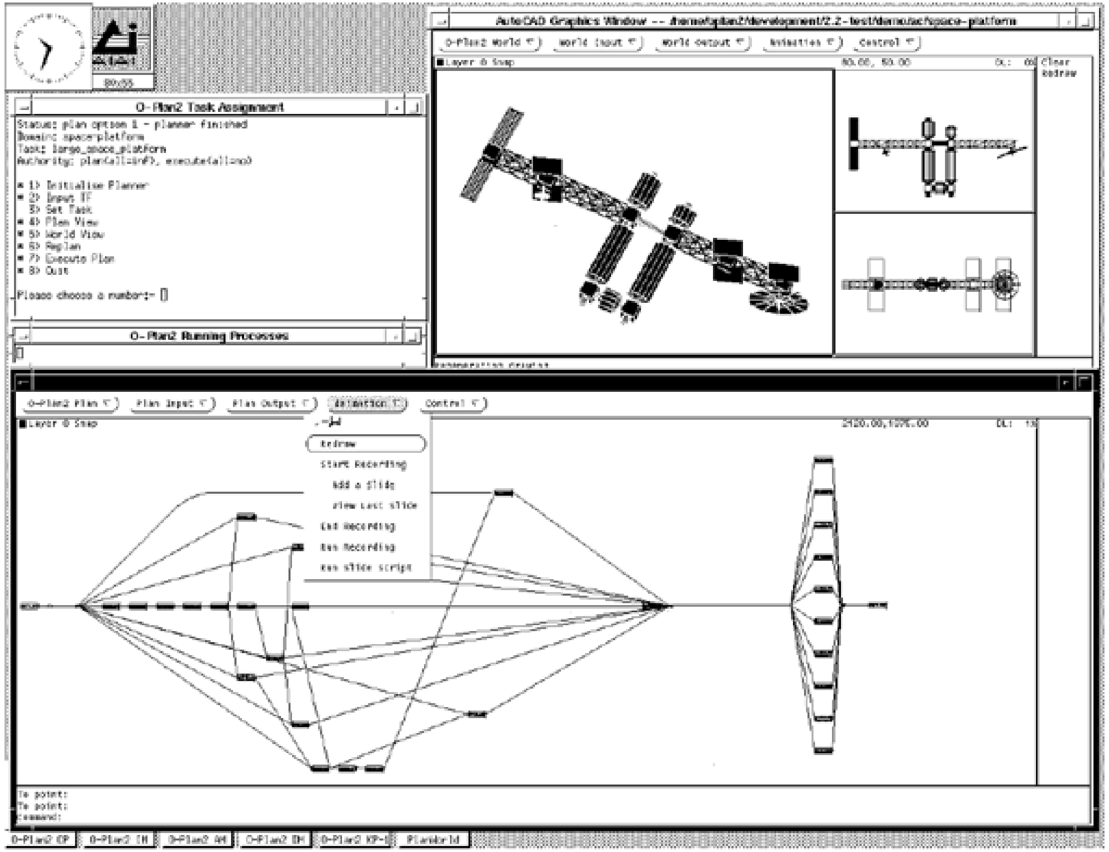
# Practical Applications of AI Planners

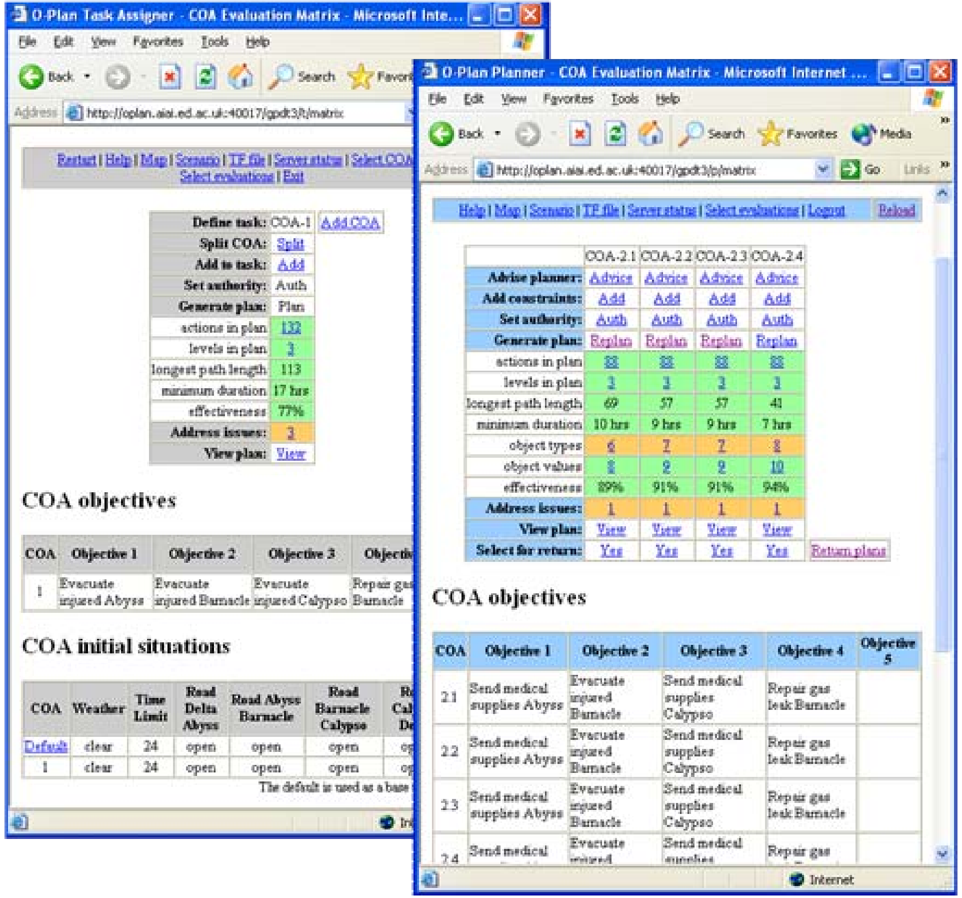
## SIPE-2 Applications

- Air campaign planning
- Military operations planning
- Oil spill response
- Production line scheduling
- Construction planning
- Planning the actions of a mobile robot
- A range of toy problems and puzzles, such as Missionaries and Cannibals

# Practical Applications of AI Planners

## O-Plan – Open Planning Architecture



# Practical Applications of AI Planners

## O-Plan Features

- Domain knowledge elicitation
- Rich plan representation and use
- Hierarchical Task Network Planning
- Detailed constraint management
- Goal structure-based plan monitoring
- Dynamic issue handling
- Plan repair and re-planning in low and high tempo situations
- Interfaces for users with different roles
- Management of planning and execution workflow

# Practical Applications of AI Planners

## O-Plan Applications

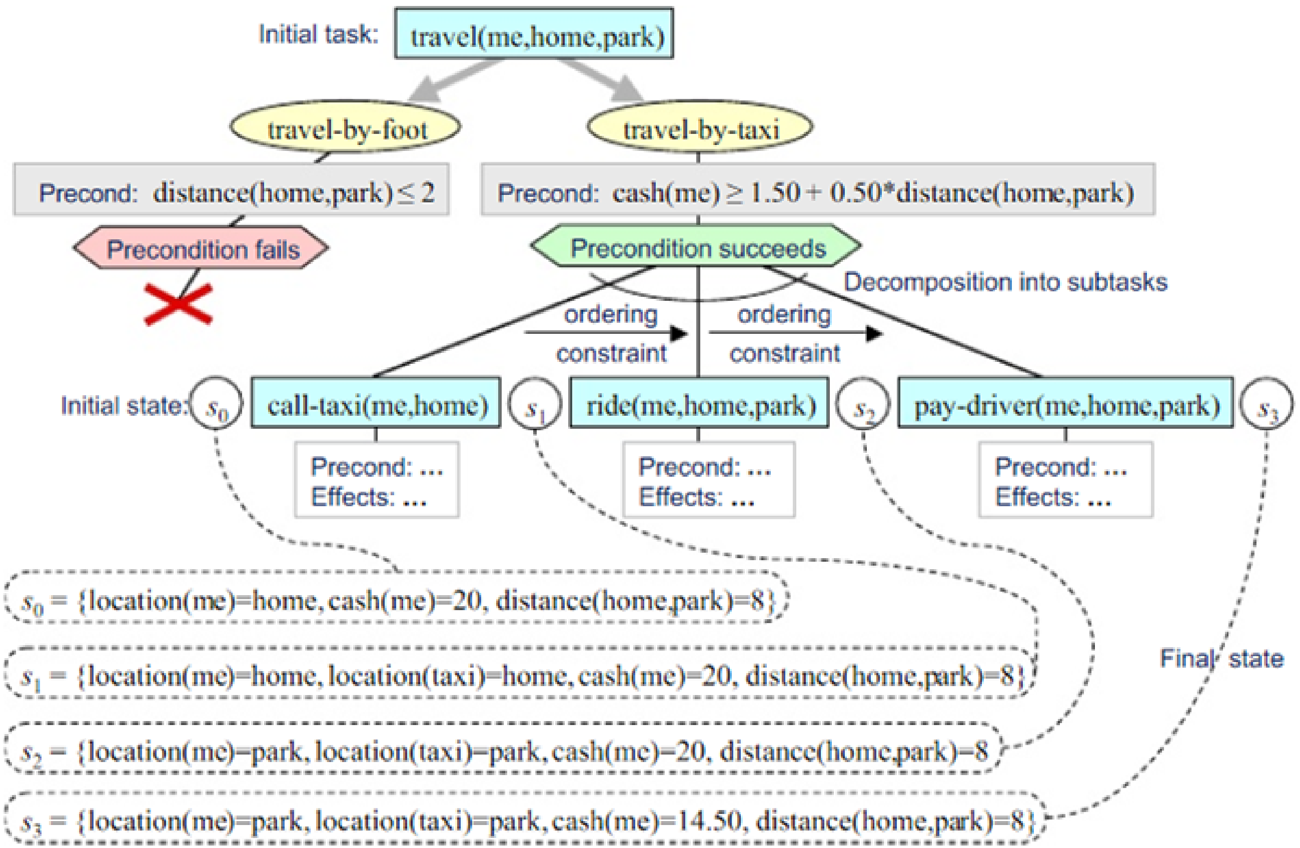
O-Plan has been used in a variety of realistic applications:

- Construction Planning (Currie and Tate, 1991 and others)
- Search & Rescue Coordination (Kingston et al., 1996)
- Spacecraft Mission Planning (Drabble et al., 1997)
- Engineering Tasks (Tate, 1997)
- US Army Hostage Rescue (Tate et al., 2000a)
- Noncombatant Evacuation Operations (Tate, et al., 2000b)
- Biological Pathway Discovery (Khan et al., 2003)
- Unmanned Autonomous Vehicle Command and Control
- Web Services Composition and Workflow Management

- O-Plan’s design was also used as the basis for Optimum-AIV (Arup et al., 1994), a deployed system used for assembly, integration and verification in preparation of the payload bay for flights of the European Space Agency Ariane IV launcher.

# Practical Applications of AI Planners

## SHOP/SHOP2 – Simple Hierarchical Ordered Planner



# Practical Applications of AI Planners
## SHOP2 Features

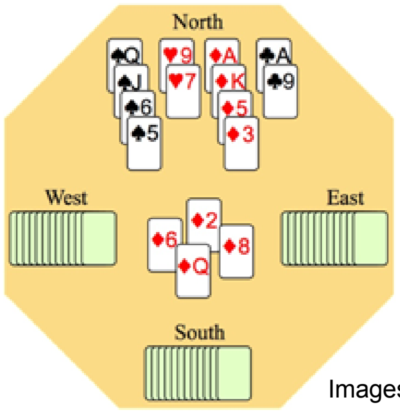
- Hierarchical Task Network Planner
- Selects activities to include in the plan in the order they will be executed
- But can interleave multiple goal and sub-goal achievement
- Axiomatic inference, quantifiers and conditional effects
- Temporal constraints
- Mixed symbolic/numeric computations
- Calls to external programs to constrain the plan
- Rich domain model and operator choice advice

# Practical Applications of AI Planners

## SHOP/SHOP2 Applications

- Evacuation Planning
- Evaluating Terrorist Threats
- Fighting Forest Fires
- Controlling Multiple UAVs
- Software Systems Integration
- Automated Composition of Web Services
- Business Workflow Management
- Project Planning

- Related Domain-Specific Planning Apps.
  – Manufacturing Automation
  – Bridge Baron (Commercial Product)




Images: Dana Nau

# AI Planning Applications

- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

# Common Features for Practical AI Planners

- Outer HTN “human-relatable” approach
- Underlying detailed constraint handling (e.g. time and resources)
- Integration with simulation and analysis
- Integration with plan execution and monitoring
- Rich knowledge modelling languages
- Comprehensive user interfaces

# AI Planning Applications - Summary

- Planning for Execution
- Deep Space 1 and Remote Agent Experiment
- Practical Applications of AI Planners
- Common Themes

# Readings

## Deep Space 1 Papers

- Muscettola, N., Nayak, P.P., Pell, B.,Williams, B.C. (1998) "*Remote Agent: to boldly go where no AI system has gone before*", in Special issue on "Artificial Intelligence 40 Years Later", Artificial Intelligence, Vol. 103, Nos. 1/2, pp. 5-48, August, 1998, Elsevier.

- Ghallab, M., Nau, D. and Traverso, P., *Automated Planning – Theory and Practice*, chapter 19, Elsevier/Morgan Kaufmann, 2004.

## Other Practical Planners

- Tate, A. and Dalton, J. (2003) O-Plan: a Common Lisp Planning Web Service, invited paper, in Proceedings of the International Lisp Conference 2003, October 12-25, 2003, New York, NY, USA, October 12-15, 2003.

- Ghallab, M., Nau, D. and Traverso, P., *Automated Planning – Theory and Practice*, chapters 22 and 23. Elsevier/Morgan Kaufmann, 2004