EECS 496: Sequential Decision Making

Soumya Ray

sray@case.edu

Office: Olin 516

Office hours : T 4-5:30 or by appointment

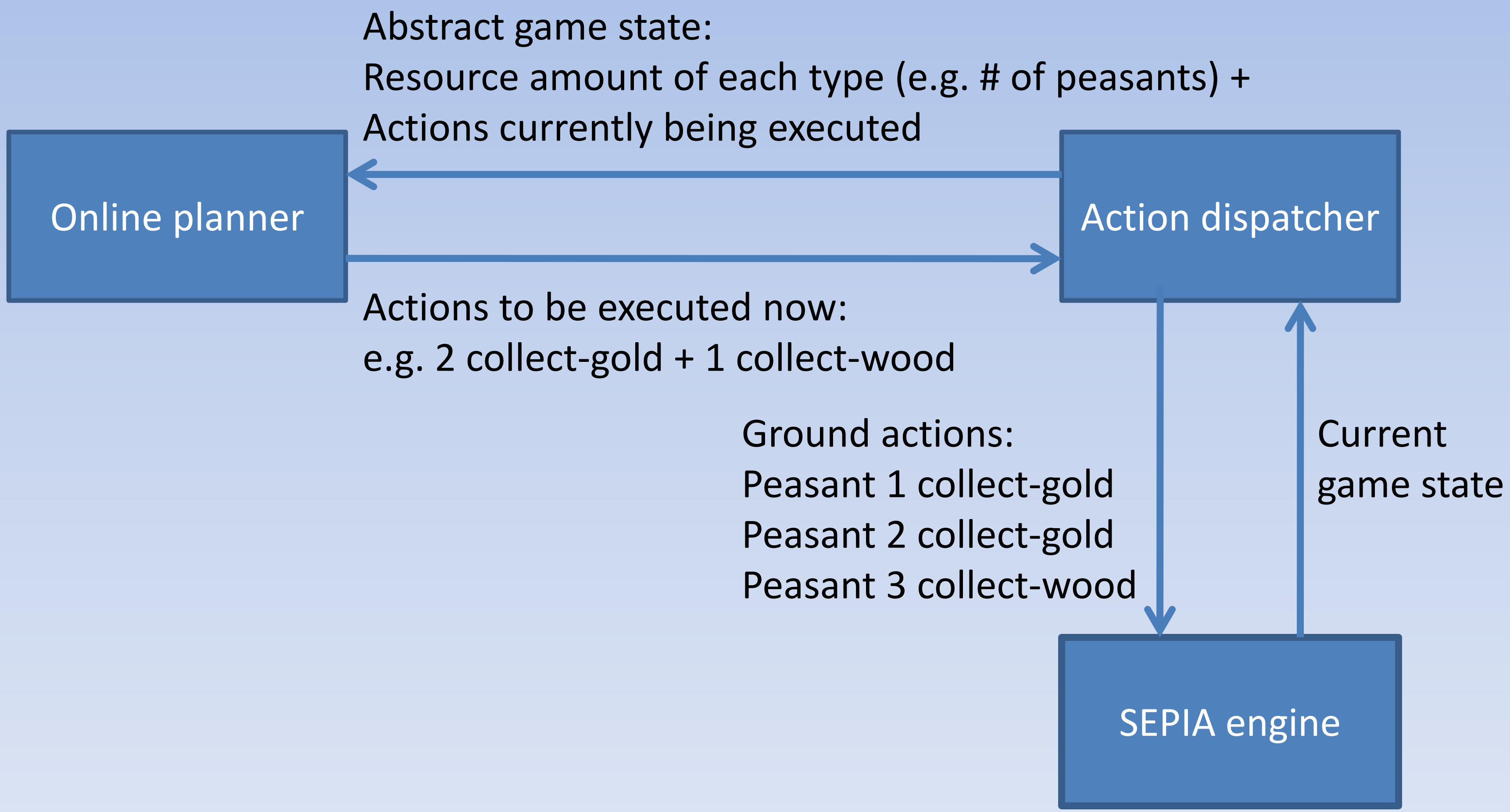
Recap

- When actions are durative, we need to ____ them after (or during) planning.
- The critical path in a POP is the path that _____.
- Shortening non-critical paths does/does not affect the makespan.
- In the critical path method, we annotate each action with an ___ ___ time and a ___ ___ time. For actions on the critical path, these two are ___.
- We use two ___ ___ in this method.
- The forward pass computes the ___ ___times. We first ____ ____ the actions. Then START gets a value ___, while for every other action, the value is the ____ of the [____ + ____] of its _____.
- The backward pass computes the ___ ___times. We first ____ ____ ____ the actions. Then END gets a value ___, while for every other action, the value is the ____ of the [____ + ____] of its _____.
- In general, _____ after planning is _____. However it is often _____ and produces ______ solutions.
- Resources are artifacts that ____ ____.
- Resources can be _____ or _______.
- When an action uses the first kind, it ____. When an action uses the second kind, it _____.
- When planning with resources, we generally use ____ _____. What are the steps in the planning loop?

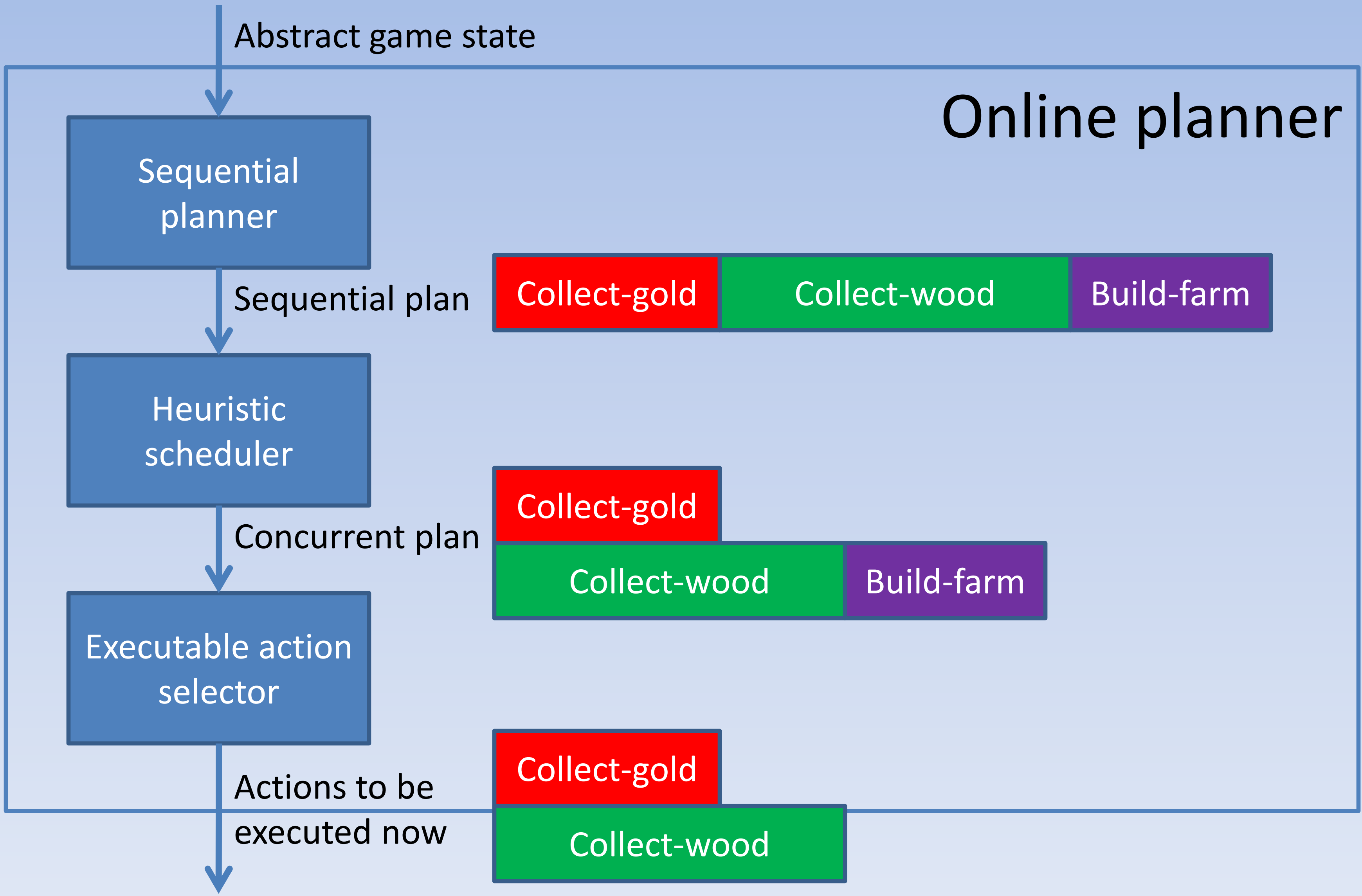
Today

- Nonclassical Planning: Actions with Durations and Resources
- Applications of planning

Online Planning for RTS resource collection



Online Planner Architecture



However, …

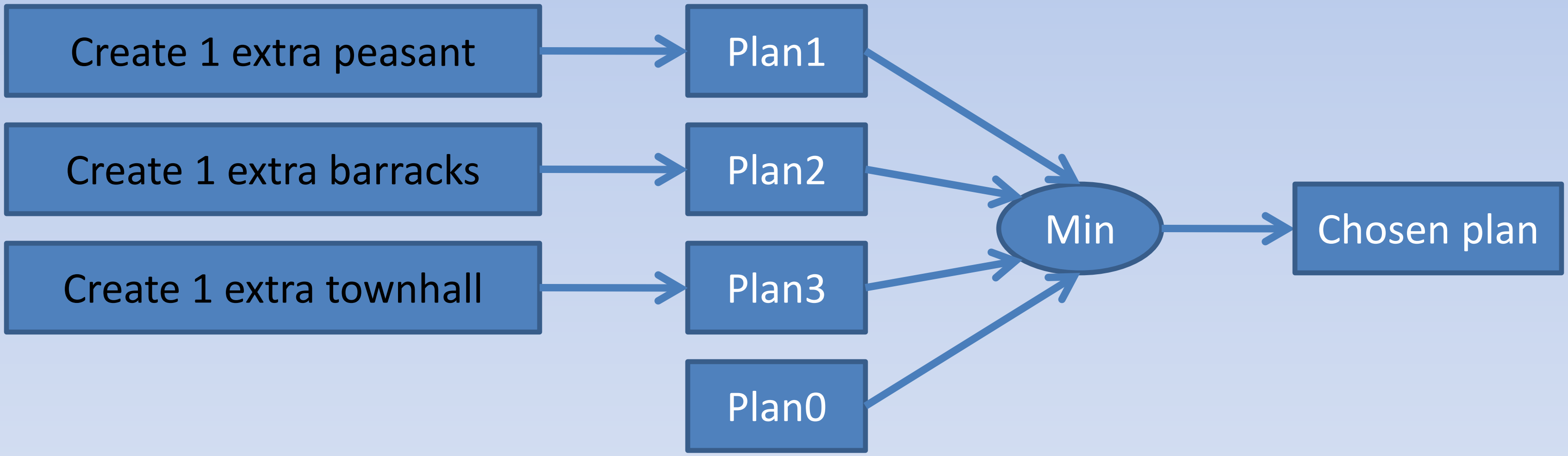
- The above planner only creates the necessary resources to achieve the goal

Collect-gold	Collect-wood	Build-farm

- If there is currently one peasant, the planner will not create any additional peasants
  - No concurrency possible
  - Creating more peasants may decrease makespan
- A good planner must create a close-to-optimal number of renewable resources, e.g. peasants

Creating Renewable Resources

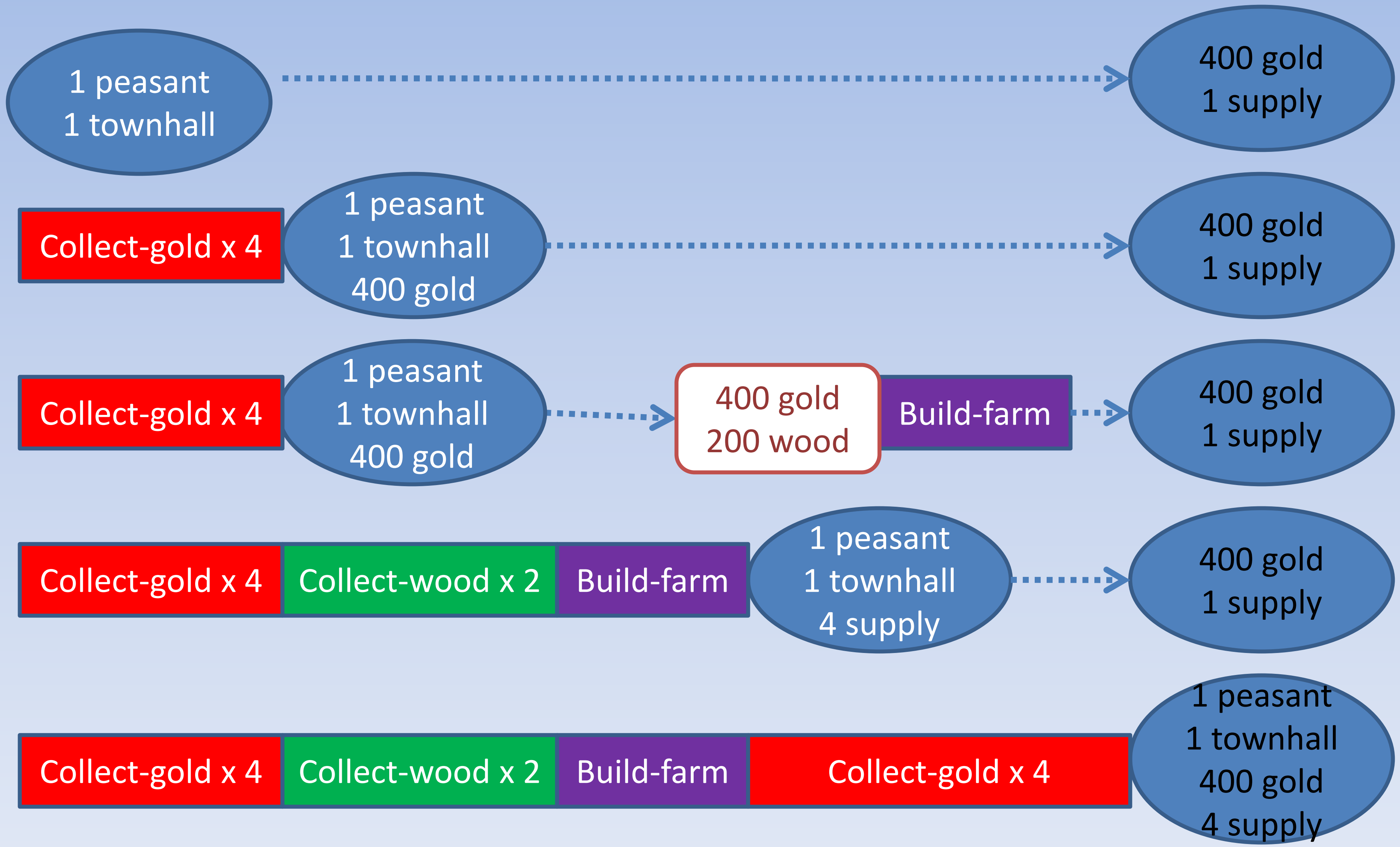
- Find plans that satisfy intermediate goals



Sequential Planner

- Means-Ends Analysis (MEA) is used to find a sequential plan to achieve the goal
- MEA iteratively constructs a plan to satisfy each resource goal
  - Recursively finds a sub-plan which satisfies the preconditions of an action which produces the resource

Means-Ends Analysis



Heuristic Scheduler

- Reschedule actions from the sequential plan to allow concurrency and decrease makespan
- Each action is moved repeatedly to an earlier time, until its preconditions no longer hold
- Rescheduling can be done in quadratic time in the number of actions in the plan
  - Not optimal, but suitable for online planning

Main Loop

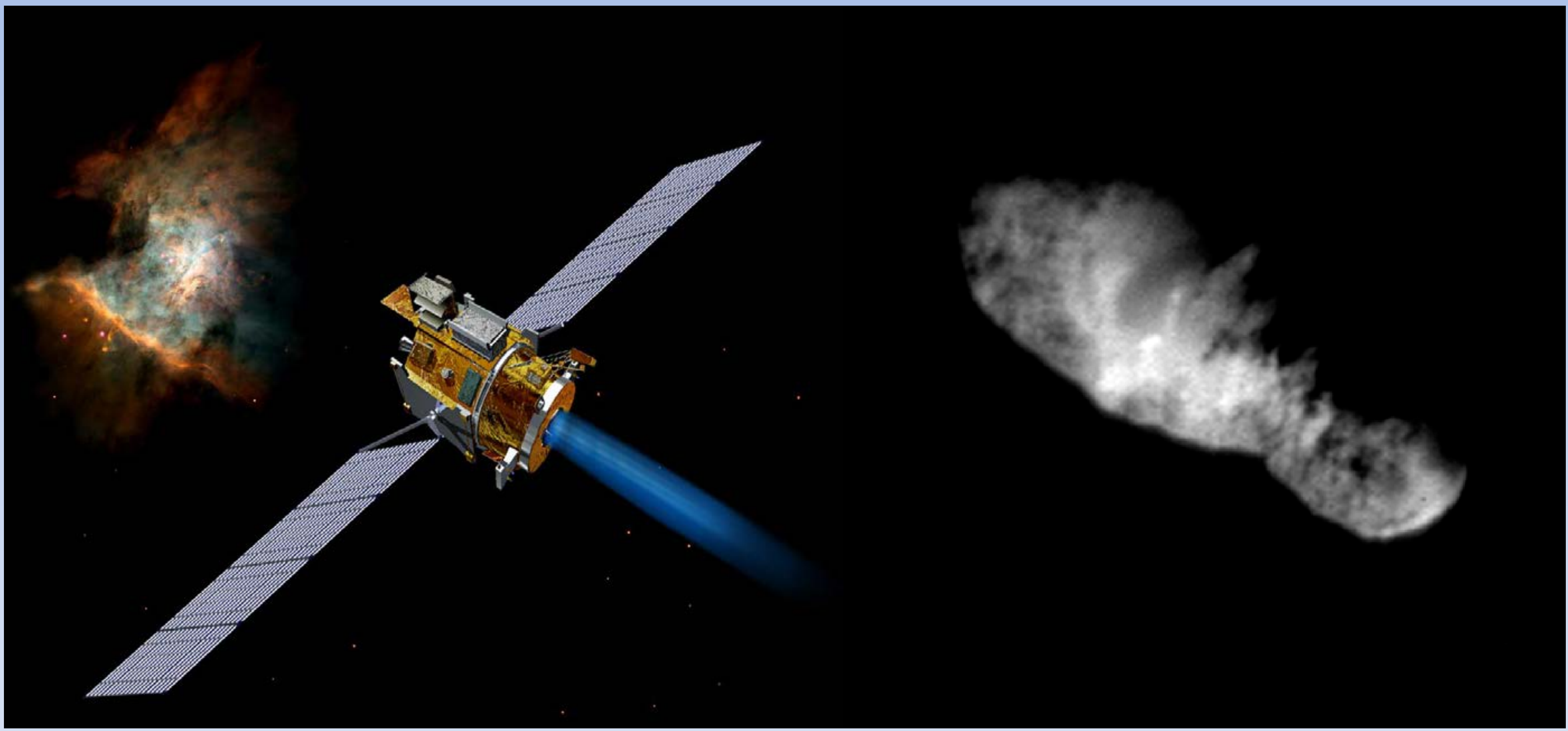
- Every few game ticks
  - Get current state
  - For each new subgoal of creating renewables
    - Create a plan that achieves the goal via creating the renewables subgoal
    - If it has a (scheduled) makespan less than the current plan, replace current plan
  - Start executing currently scheduled actions of new plan

Limitations

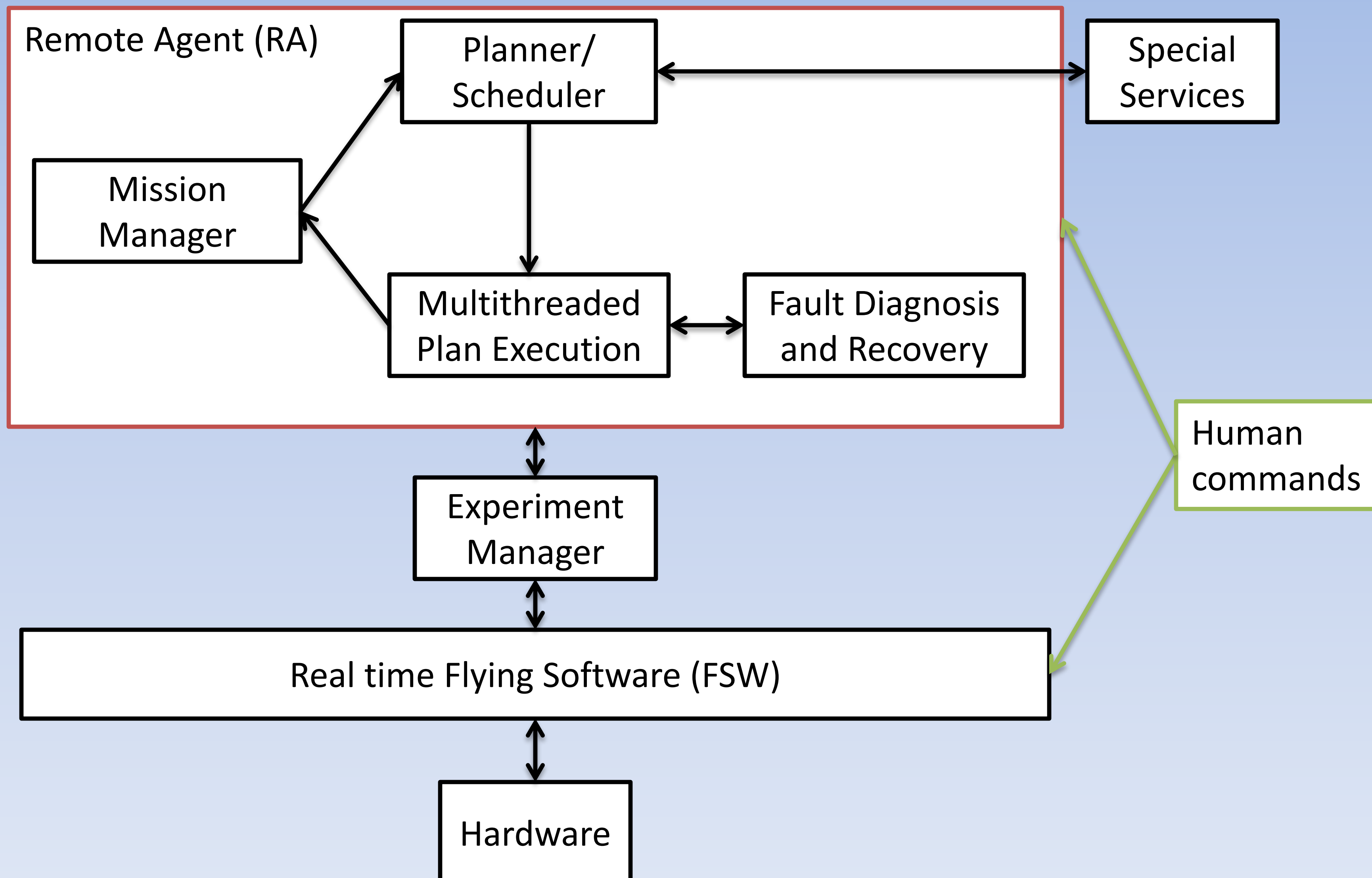
- Actions have constant durations and effects
  - Durations vary for actions such as collect-wood
  - Effects can change due to new technology
- Unable to infer object identity
  - May be necessary for some actions
- No spatial reasoning
  - Necessary for building placement
- Actions cannot be stopped
  - Necessary if goals or environment change

Application: Deep Space 1

(Active mission Oct 24 1998-Dec 18 2001)



Architecture



Architecture

- Mission manager selects goals for the next horizon (two days)

- Planner/Scheduler plans and schedules actions based on internal high level spacecraft model
  - Might require special services to e.g. determine location in space relative to “beacons”
  - Might create a sub-planning problem

Architecture

- Execution module sends commands to flight systems
  - Translates (relatively) high level plan from planner to actions for flight control software
  - If there is a system failure, attempts recovery
  - Performs conditional execution of plan if needed
  - Manages time and resources

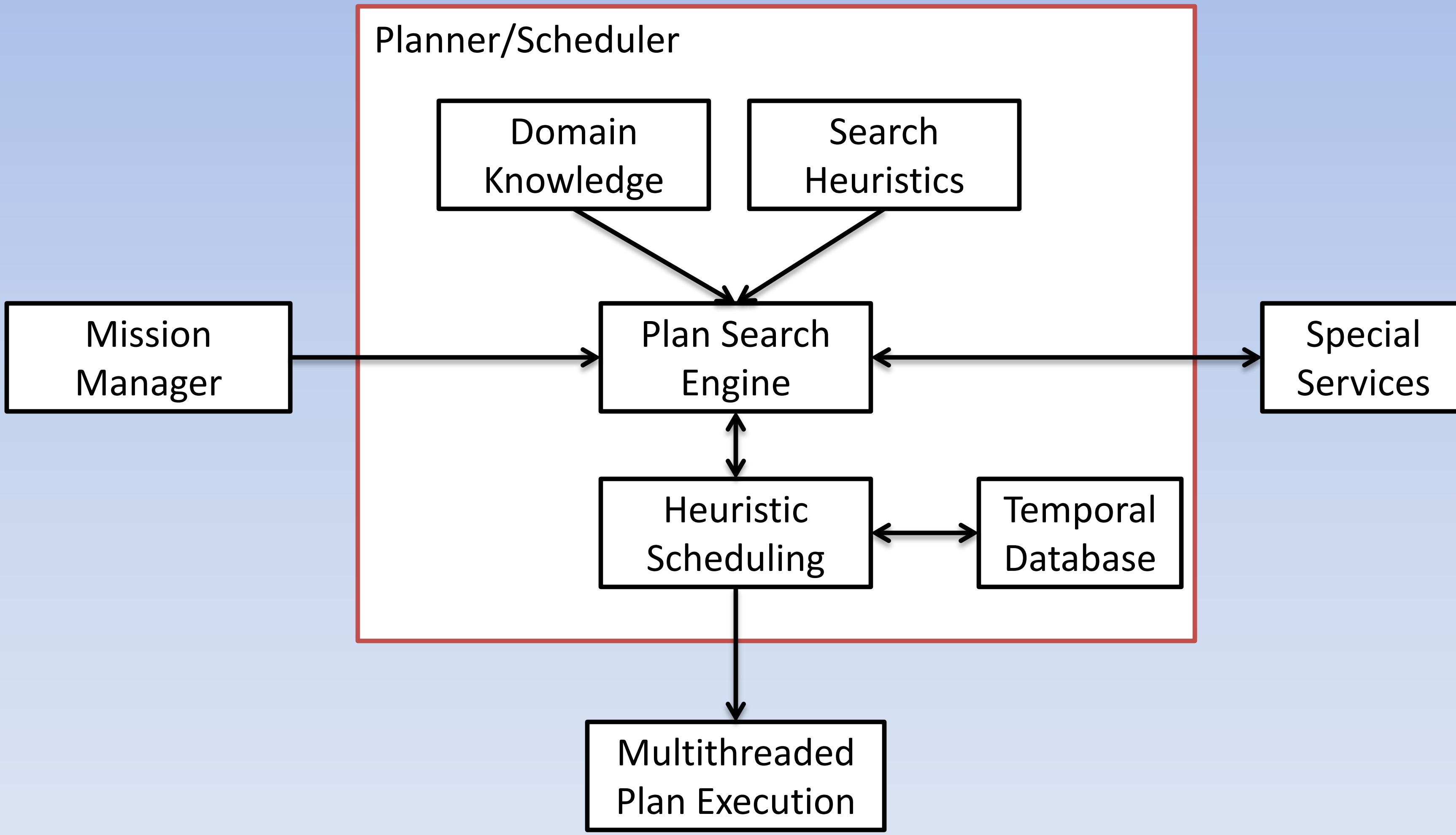
Architecture

- Failure Diagnosis and Recovery
  - As commands executed, tracks status and compares to expected based on low level component models
  - Identifies when failures have occurred in components and notifies execution module
  - Also contains procedures for possibly fixing failures

Dealing with Uncertainty

- Can deal with uncertain events through
  - Model based plan repair
  - Replanning

Planner/Scheduler



Planner/Scheduler

- State space heuristic search
  – Heuristics involved detailed domain knowledge (not just relaxation/subplans)

- Uses Domain Definition Language (very similar to PDDL)

Scenarios

- System was tested in flight using several scenarios. In most complex case, system had to operate autonomously for two days, in the presence of a simulated failure, align spacecraft with comet, and take pictures.
  - It worked!
  - With a couple of hiccups…

Hiccup 1

- Planner backtracked more than expected
  - Traced to different information in the autonomous nav system
  - Still produced valid plans, showed system robust to changes

Hiccup 2

- At some point, the plan required a thruster operation to commence. It did not.
  - Traced to a race condition between two EXEC threads, leading to resource deadlock
  - A critical section had not been identified
  - Glitch was not activated during thousands of simulations on the ground, but was activated the first time in space
    - Software bug!

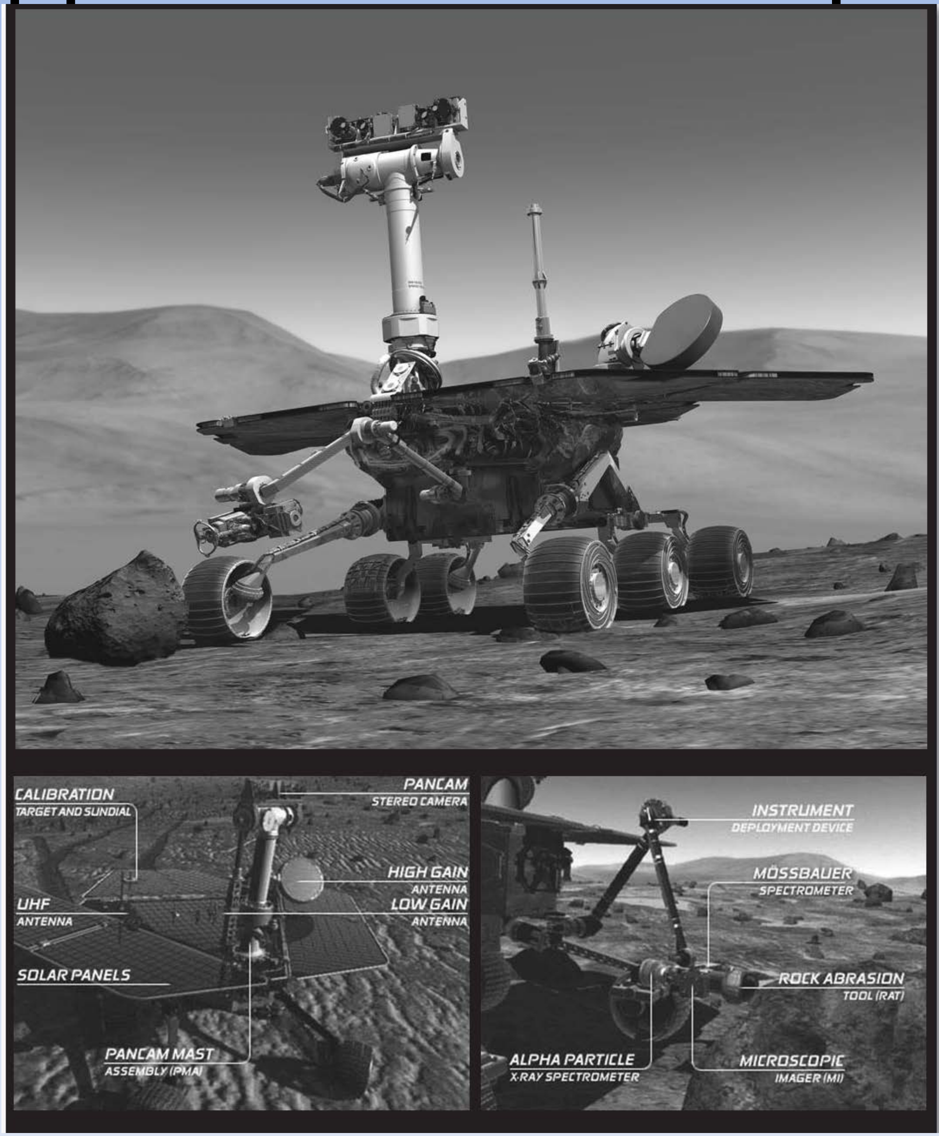
Hiccup 3

- At some point, the link between RA and FSW broke leading to packet loss. This caused RA to estimate the wrong status of the spacecraft.
  - Fortunately, this quickly resolved itself, and resulted in no effect.

Lessons

- Overall, the experiment was deemed successful and showed that automated planning could be used to support human decision making in complex environments

Application: Mars Exploration Rovers



Active mission:
Spirit: 1/2004-3/2011
Opportunity: 1/2004-6/2018

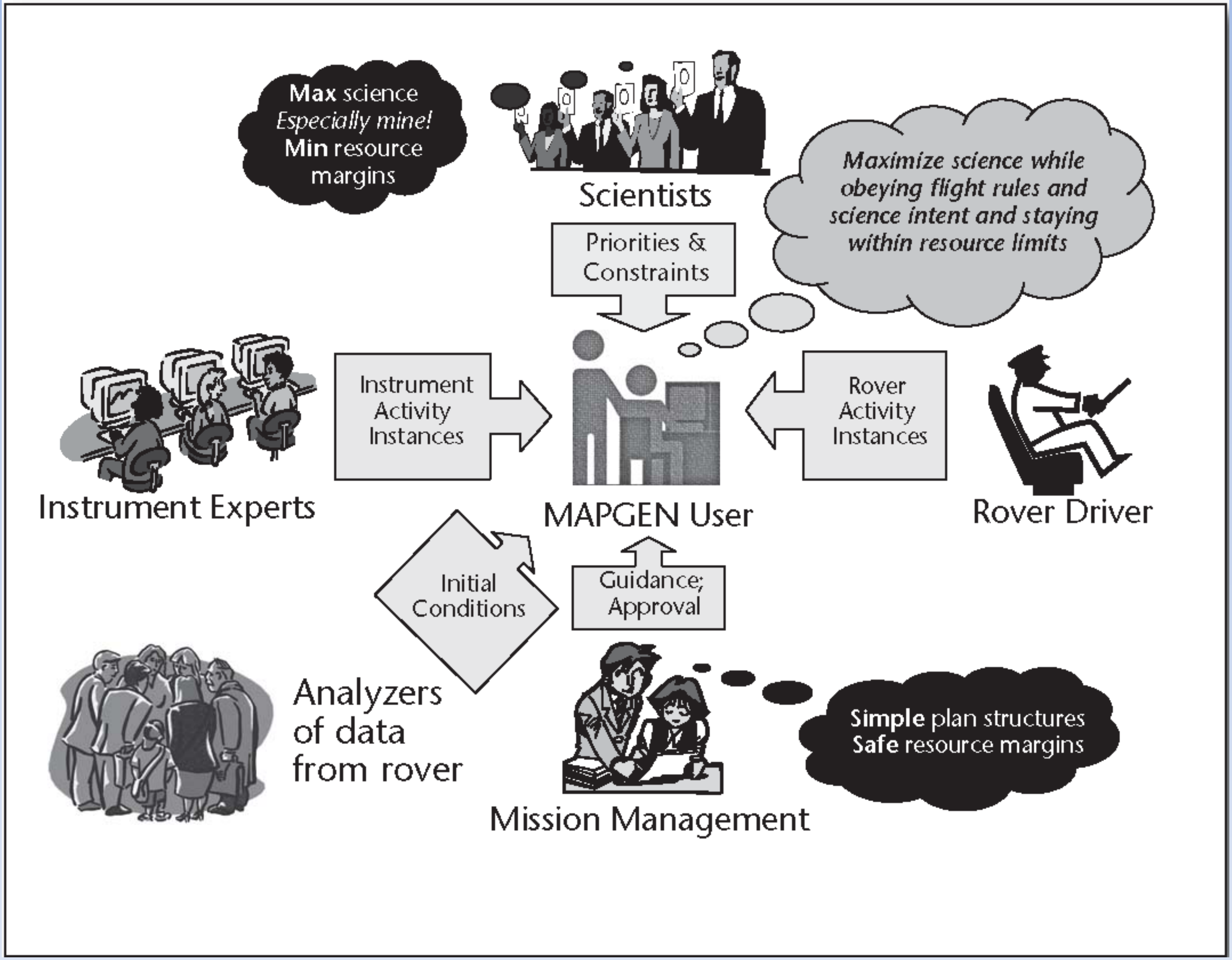
Planning in the MERs

- No on-board planner (power constraints, much more uncertain environment)
- Rather, a ground (JPL)-based tool, MAPGEN, was used *collaboratively* by a human mission planner and an automated planner to plan offline for the next sol's activity
  - Then uploaded to MER for execution
  - Done *every day* for ongoing mission

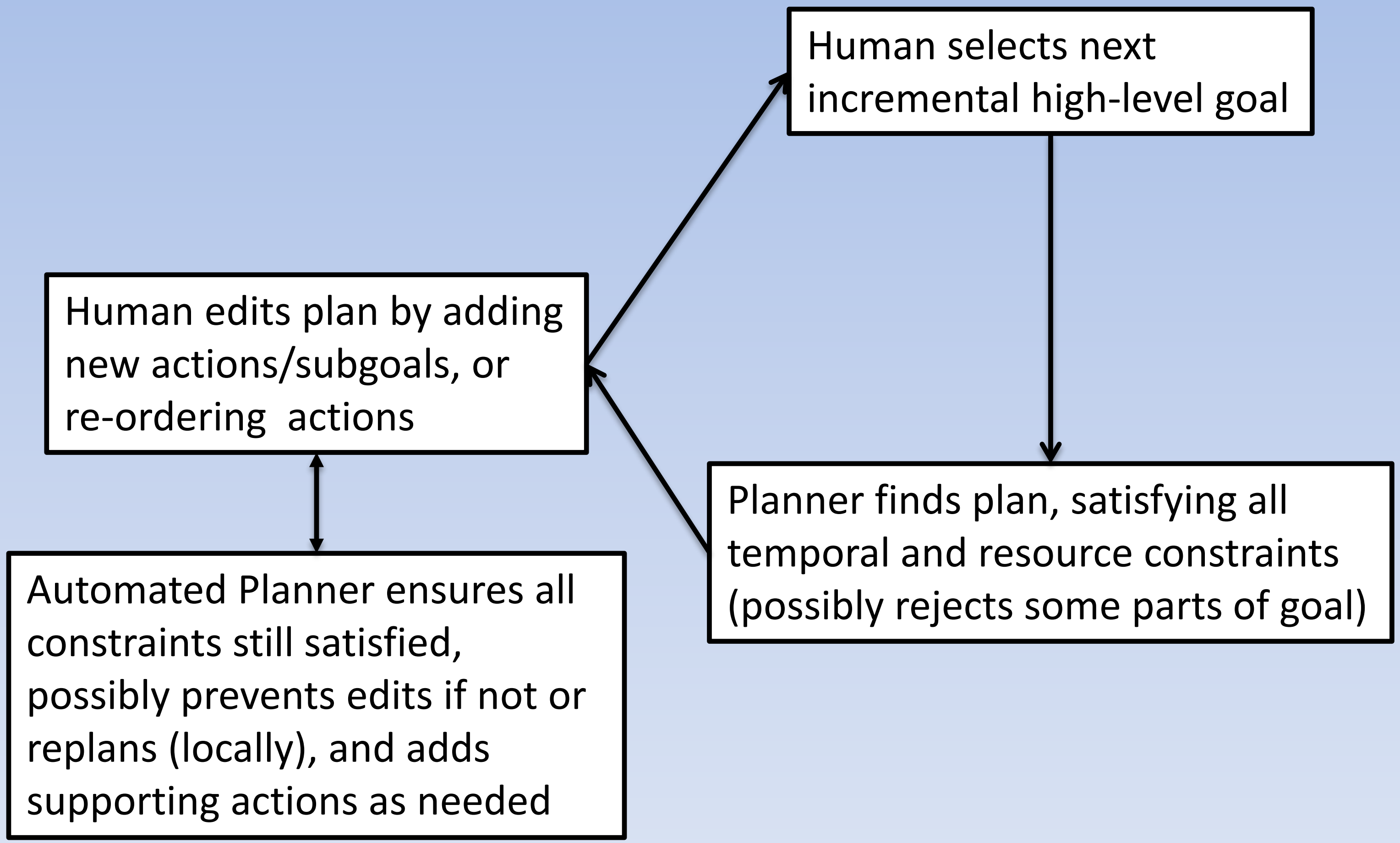
Mixed Initiative Planning

- Collaborative planning like this is called "mixed-initiative" planning
  - Sometimes the human has control, sometimes the automated system

Mixed Initiative Planning



MIP Loop


Human selects next incremental high-level goal
Planner finds plan, satisfying all temporal and resource constraints (possibly rejects some parts of goal)
Human edits plan by adding new actions/subgoals, or re-ordering actions
Automated Planner ensures all constraints still satisfied, possibly prevents edits if not or replans (locally), and adds supporting actions as needed

Automated Planner

- The planner in MAPGEN is an updated version of the DS1 planner
  - Updated to support MIP, supports different levels of autonomy
  - Handles temporal and resource constraints

- Updated version of this planner, M-SLICE, used for MSL

Summary of Part 2

- STRIPS planning
- POP
- Graphplan
- SATPlan
- Planning with time and resources
- Some examples of real-world planning systems