Mission Planning and Execution
Within the Mission Data System

Anthony Barrett, Russell Knight,
Richard Morris, Robert Rasmussen

Jet Propulsion Laboratory
California Institute of Technology

Motivation

- A framework for planning, scheduling, and execution.
- A control centered approach toward representing plans as opposed to an action centered approach.
- This is the current default system for flying onboard the MSL rover.
Outline

- Plan/problem representation
  - Partially ordered constraints on state variables.
- MPE component architecture
  - Numerous threads of execution
- Elaboration & Scheduling
  - Subsumes both PO and HTN planning
- Plan execution
  - Firing time-points & enforcing constraints

The Problem

- Givens
  - A model of how state variables affect each other
  - Tactics for elaborating constraints into supporting constraints.
  - A temporally constrained set of commanded constraints on state variables.
- Objectives
  - Elaborate the constraints into an executable network, where constraints can be incrementally passed to controllers for each state variable.
  - Execute the elaborated network.
Plan/Problem Representation

- A network of timepoints connected by temporal and state-variable constraints.

Definitions

- Timepoint
  - A flexible point in time

- Temporal Constraint
  - A timing relation between two Timepoints

- Goal
  - A State Constraint over a time interval demarcated by two Timepoints

- State Constraint
  - A representation of a set of values, used to specify a required (allowable) state

- Executable goals (xgoal)
  - A merged state constraint that can be passed to a state variable controller
Example Problem (Givens)

- **State variable model**
  - Heater Mode → Temperature → Transmitter

- **Constraint support tactics**
  - Transition Temp 10°-20° C
  - Maintain Temp 10°-20° C
  - Heater On

- **Commanded constraints**
  - Transition Temp X°-Y° C
  - Maintain Temp X°-Y° C
  - Heater On

Example Problem (Executable Network)

Example Problem (Executable Network)
MPE Component Architecture

Actual MPE Components
(each has an execution thread)
Objective
➢ To set up conditions where a state variable’s controller can enforce a goal’s constraint.

Algorithm
Copy executing task net to proposed network for modification
While there are goals to elaborate do
  Choose a goal $G_{[S,E]}$ to elaborate (heuristically ordered)
  Add goal to proposed network
  Exhaustively choose elaboration tactic $M$ for $G$
  Apply $M$, which possibly generates new goals to elaborate
    - Backup (and remove goals) if application of $M$ is illegal
Backtracking

- When an elaborator fails to elaborate
  - Scheduling/propagation fails
  - Cannot find a Tactic that works
- Reports failure to its parent elaborator
- Parent can tell a sibling of the failed Elaborator to re-elaborate
  - Doing things a different way could clear up conflicts with the sibling’s elaboration
- Then tell the failed Elaborator to “try again”
- Can do this for all supporting goals

Goal Net Scheduling

- Objective
  ➢ To merge goals into executable x-goals

- Algorithm (used during promotion)
  For each state variable \textit{SVAR} (from a heuristic ordering) do
  For each new goal \textit{G@[S,E]} on \textit{SVAR} (heuristically ordered) do
    Exhaustively choose how to constrain \textit{G}’s start timepoint \textit{S}
    Exhaustively choose how to constrain \textit{G}’s end timepoint \textit{E}
    Merge \textit{G@[S,E]} into \textit{SVAR}’s timeline – Backup if merge illegal
    Propagate the expected behavior of \textit{SVAR} given the new merges
    - Backup if the propagation is illegal
  Replace executing task net with proposed one if it scheduled
Computing X-Goals

Module Temperature

Heater Mode & Health

0 to =

Module Temperature: transition to temp between X and Y degrees C

Heater Switch & Health: healthy and switching between on and off

0 to =

Module Temperature: between X and Y degrees C

Heater Switch & Health: healthy and switching between on and off

6/20/04

IWPSS 2004

15
Computing X-Goals

Module Temperature
between X and Y degrees C

Heater Switch & Health: healthy and switching between on and off

Heater Switch & Health: healthy and switching between on and off

Module Temperature: transition to temp between X and Y degrees C

Unconstrained
Computing X-Goals

Module Temperature

Heater Mode & Health

Temperature

Heater Mode & Health

deg C

deg C

deg C

deg C

On and Off

On and Off

On and Off

On and Off

Unconstrained

6/20/04

IWPSS 2004

19
Computing X-Goals

Objective

To feed x-goal constraints to state variable controllers as temporal constraints require and circumstances permit.

Algorithm

For each unfired timepoint TP not constrained into the future do

If TP's XGoals can start or TP is about to time out then (fire)

For each Goal G@[* , TP] do Stop monitoring G

For each XGoal XG@[TP , *] do

Change imposed constraint on XG's state variable to XG

For each Goal G@[TP , *] do Start monitoring G
When A fires, estimator and controller get new instructions (a state constraint with timing information).

100 Seconds Later, Start Conditions are Monitored

B will fire when all outgoing XGoals (there is one in this example) are ready.
Timepoint B Fires

Mission Data System

Transition Temp 10-20

Maintain Temp 10-20

New state constraint replaces previous one in controller and estimator

XGoalsChecker

State Constraint and Timing

Temperature State Variable

Temperature Estimator

Temperature Controller

B will fire when temperature gets in range

Failure Response

GoalsCoordinator monitors executing goals

Upon detecting that an active goal is not being satisfied the GoalsCoordinator can:

1. Reschedule to delay the goal and its supporting goals;
2. Reschedule to delay the goal that the failed goal supports, progressively moving farther up;
3. Re-elaborate the failed goal and reschedule;
4. Re-elaborate the goal that the failed goal supports and reschedule, progressively moving farther up;
5. Remove the set of operator-requested constraints that resulted in creating the failed goal; and
6. Remove all operator requests and enter a safe state
Putting It All Together:
a high-level characterization of current MPE algorithm

- Elaboration algorithm
  - Copy executing task net to proposed network for modification
  - While there are goals to elaborate do
  - Choose a goal \( G \in [S,E] \) to elaborate (heuristically ordered)
  - Add goal to proposed network
  - Exhaustively choose elaboration tactic \( M \) for \( G \)
  - Apply \( M \), which possibly generates new goals to elaborate
    - Backup (and remove goals) if application of \( M \) is illegal

- Scheduling algorithm (used during promotion)
  - For each state variable \( SVAR \) (from a heuristic ordering) do
  - For each new goal \( G \in [S,E] \) on \( SVAR \) (heuristically ordered) do
  - Exhaustively choose how to constrain \( G \)'s start timepoint \( S \)
  - Exhaustively choose how to constrain \( G \)'s end timepoint \( E \)
  - Merge \( G \in [S,E] \) into \( SVAR \)'s timeline - Backup if merge illegal
  - Propagate the expected behavior of \( SVAR \) given the new merges
    - Backup if the propagation is illegal
  - Replace executing task net with proposed one if it scheduled

- Execution algorithm
  - For each unfired timepoint \( TP \) not constrained into the future do
    - If \( TP \)'s \( X \)Goals can start or \( TP \) is about to time out then (fire)
    - For each Goal \( G \in [*,TP] \) do Stop monitoring \( G \)
    - For each \( X \)Goal \( XG \in [*,TP] \) do
      - Change imposed constraint on \( XG \)'s state variable to \( XG \)
    - For each Goal \( G \in [TP,*] \) do Start monitoring \( G \)

Concluding Observations

- The focus is on controlling to enforce constraints on state variables.
  - As opposed to the operator focus in AI planning.
  - Facilitates a clear way to merge concurrent action.
- The use of components and elaboration tasks enables a surprising amount of flexibility.
  - GEL can be used, but other approaches to elaboration are possible (in the same implementation).
  - The scheduler component can easily be replaced.
- Lifted time is the default, but it is not necessary.