Technology Infusion into the Mars Science Lab Mission Planning and Execution System

Anthony Barrett, James Kurien, Mark Drummond, Michael Freed, Sven Grenander, George Stebbins

Goals/Objectives

- Enhance the performance and capability of the Mars Science Lab's core Mission Planning and Execution (MPE) system through the application of AI techniques developed within the Intelligent Systems program.

Schedule & Milestones

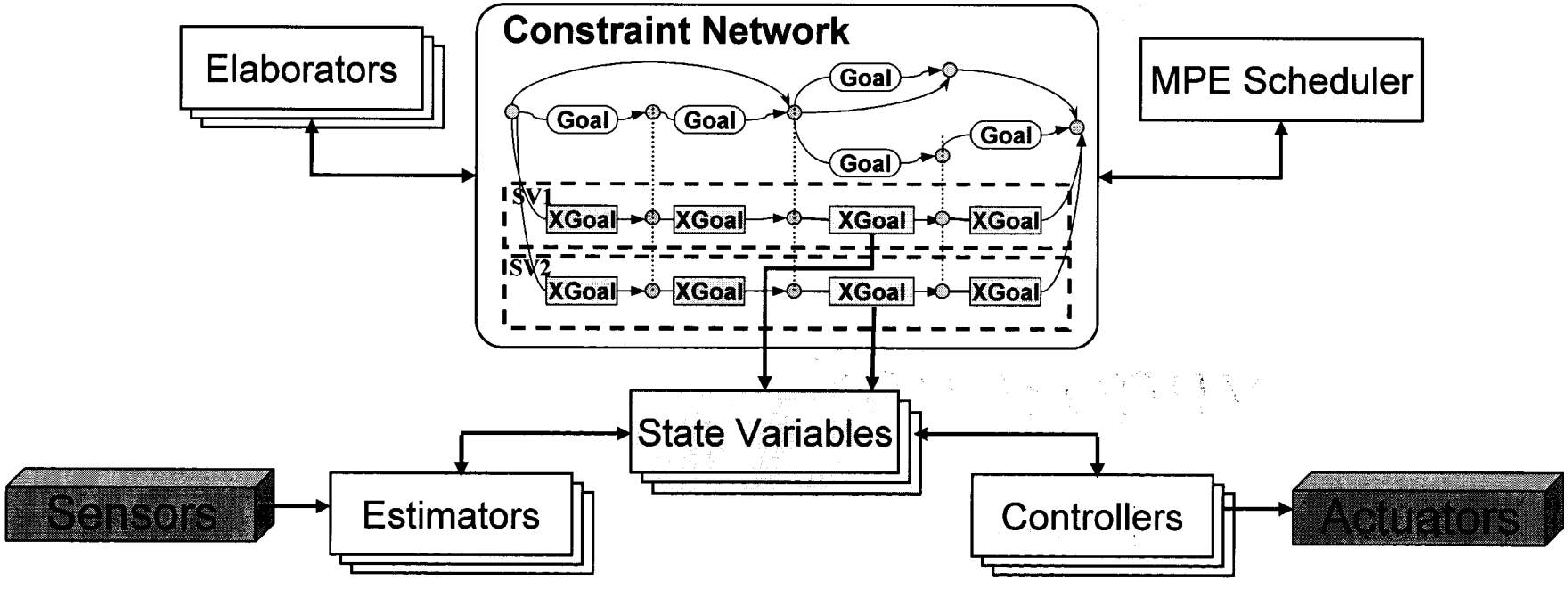
- FY 03
  - To analyze the MPE system and cleanly characterize its underlying algorithms.
  - Develop agreements with Mars Science Lab to enable the infusion into a flight project.
- FY 04
  - Improve performance with existing techniques in search control, anytime algorithms, fast temporal propagation
  - Improve goal net robustness with focused extensions from the world of flexible plan generation and execution.
- FY 05
  - Allow fail-operational modes with fault localization, redundancy management techniques
  - Participate in MSL Integrated Software Demonstration

Recent Results

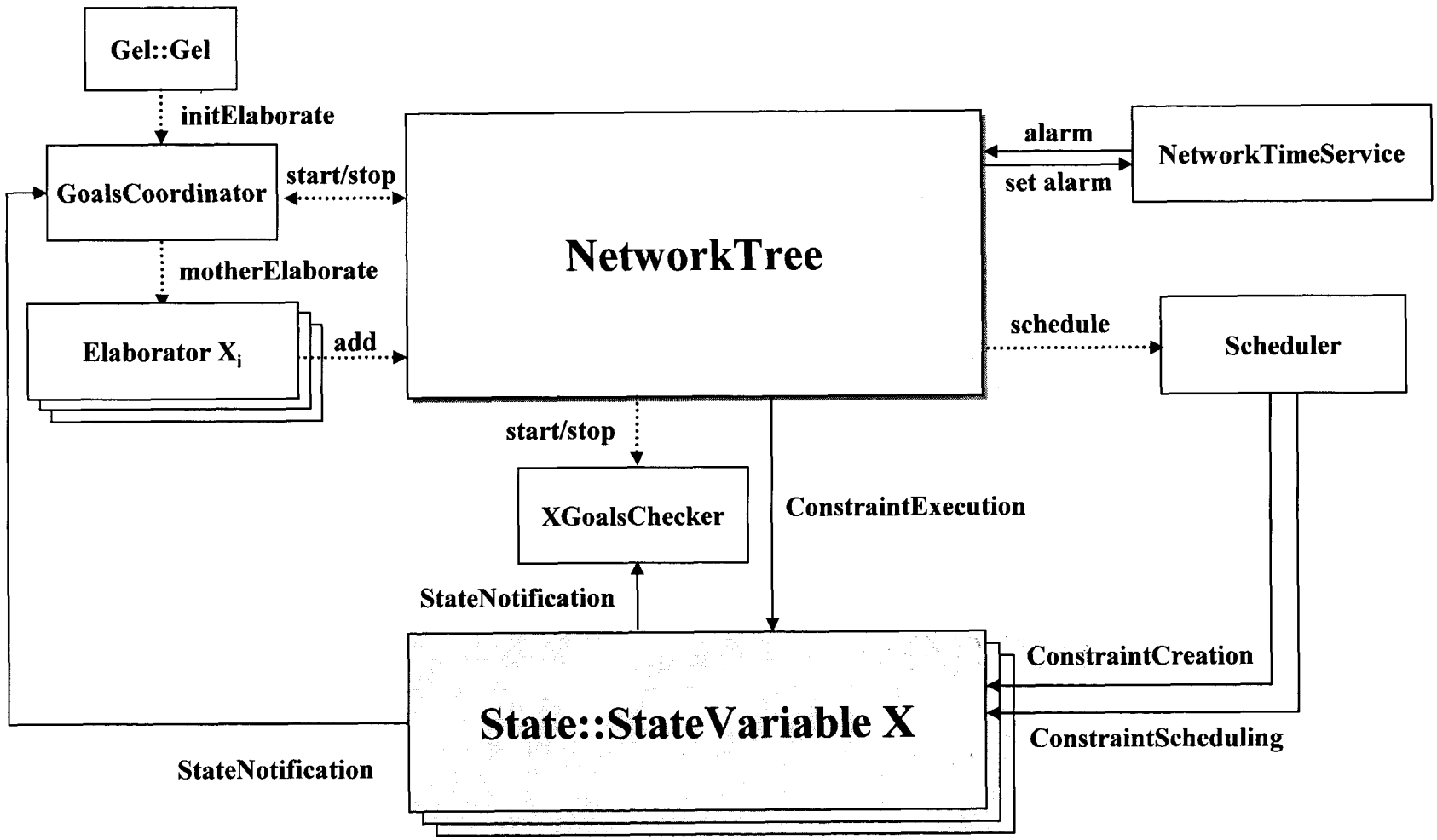
- Analysis of the MPE system to cleanly characterize its underlying algorithms.
- Defined the MPE/eMPE boundary
- Developed a technology infusion plan (currently in the signature process).

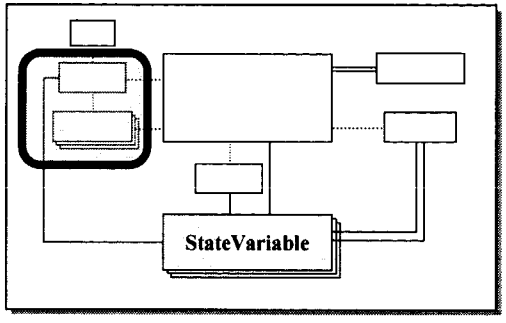
Underlying Philosophy

- Model a spacecraft as a collection of interacting state variables.
- Control the spacecraft through the time varying enforcement of constraints on state variables.



Actual MPE Components
(each has an execution thread)

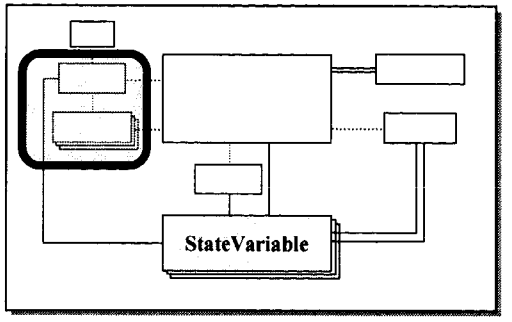




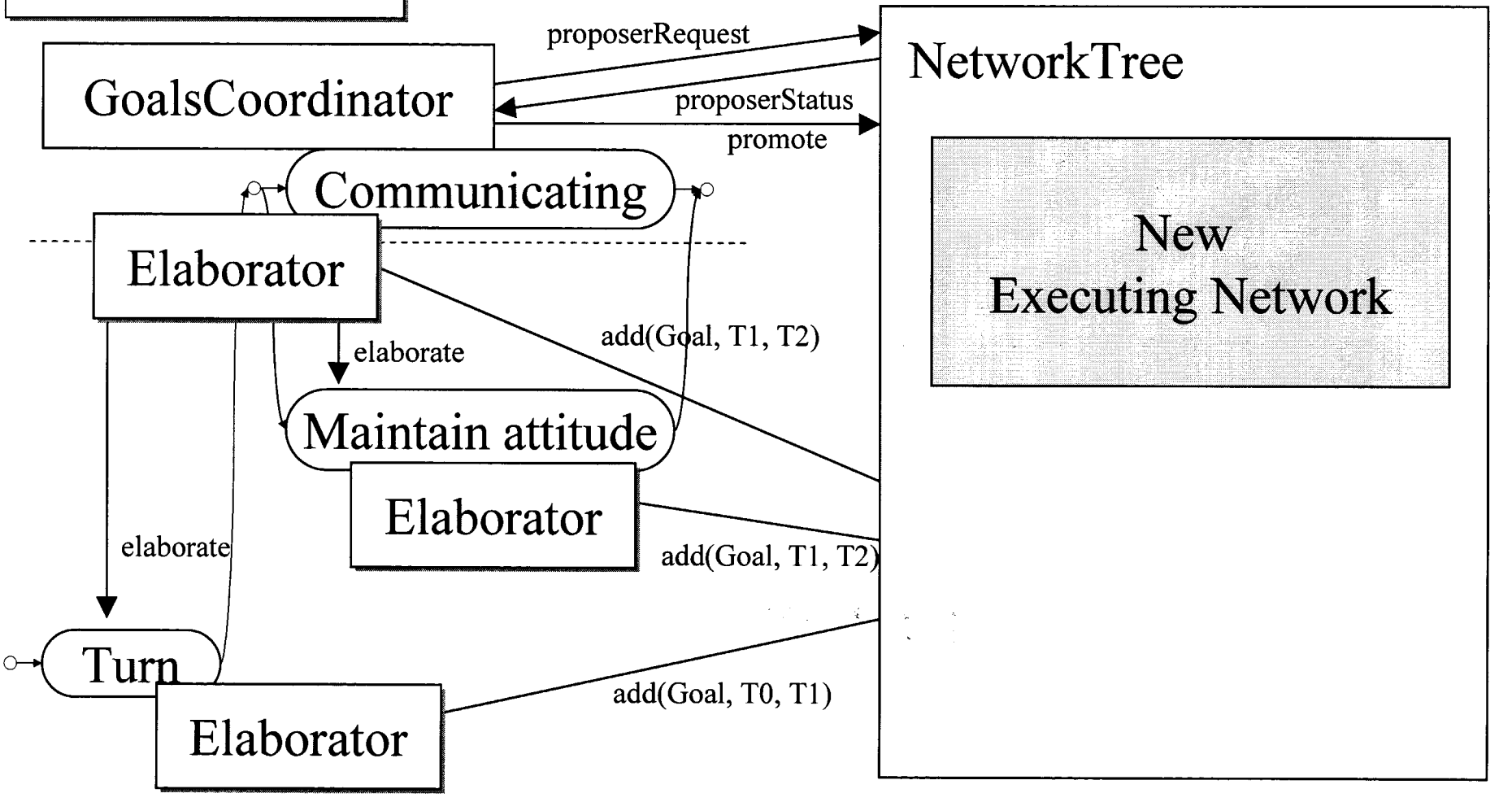
Elaboration

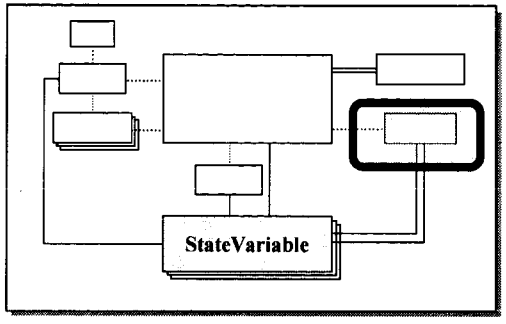
- Objective
  - To set up conditions where a state variable's controller can enforce a goal's constraint.
- Algorithm

Copy task network to proposed network for modification
While there are goals to elaborate do
  Choose a goal **G** to elaborate (from a heuristic ordering)
  Exhaustively choose elaboration tactic **E** for **G**
  Apply **E**, which possibly generates new goals to elaborate
    - Backtrack if application of **E** illegal



Elaboration



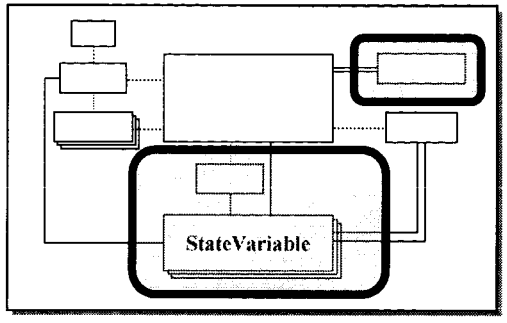


Goal Net Scheduling

- Objective
  - To merge goals into executable x-goals
- Algorithm (used during promotion)

For each state variable **SVAR** (from a heuristic ordering) do
  For each new goal **G** on **SVAR** (from a heuristic ordering) do
    Exhaustively choose **G**'s constrained start timepoint **S**
    Exhaustively choose **G**'s constrained end timepoint **E**
    Merge **G**@[**S**,**E**] into **SVAR**'s timeline – Backtrack if merge illegal
  Propagate the expected behavior of **SVAR** given the goals
    – Backtrack if the propagation illegal

Replace executing task network with proposed one if it scheduled

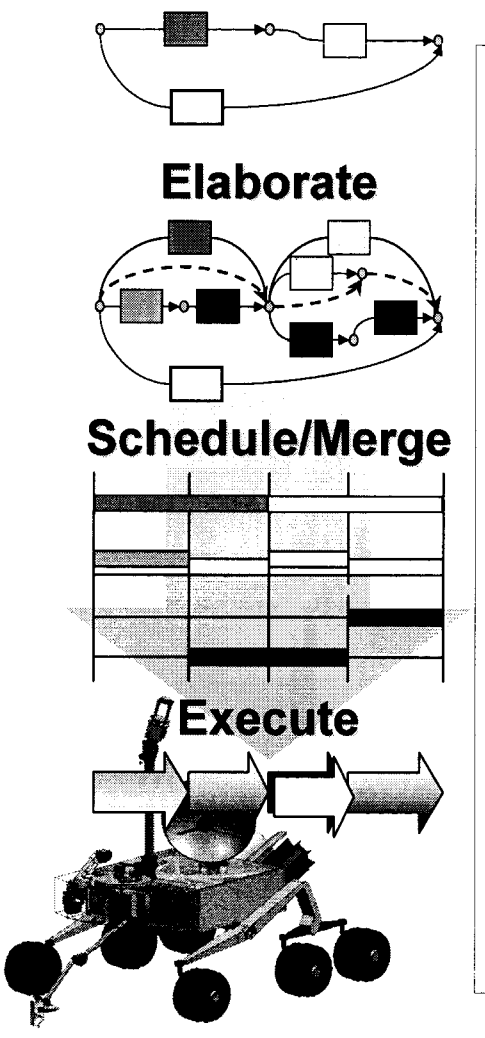


Goal Net Execution

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

For each unfired timepoint **TP** not temporally constrained into the future do
  If **TP**'s x-goals are ready to start or **TP** is about to time out then (fire)
    For each x-goal **G**@[**TP**,*] respectively do
      Send "start[**G**]" to **G**'s state variable's controller & estimator
      Inform **G**'s parent goals to start or stop monitoring **G**'s status

Putting It All Together: a high-level characterization of current MPE algorithm



- Elaboration algorithm

  Copy task network to proposed network for modification
  While there are goals to elaborate do
    Choose a goal **G** to elaborate (from a heuristic/random ordering)
    Exhaustively choose elaboration tactic **E** for **G**
    Apply **E**, which possibly generates new goals to elaborate
      – Backtrack if application of **E** illegal

- Scheduling algorithm (used during promotion)

  For each state variable **SVAR** (from a heuristic/random ordering) do
    For each goal **G** on **SVAR** (from a heuristic/random ordering) do
      Exhaustively choose **G**'s constrained start timepoint **S**
      Exhaustively choose **G**'s constrained end timepoint **E**
      Merge **G**@[**S**,**E**] into **SVAR**'s timeline – Backtrack if merge illegal
    Propagate the expected behavior of **SVAR** given the goals
      – Backtrack if propagation illegal
  Replace executing task network with proposed one if it scheduled

- Execution algorithm

  For each unfired timepoint **TP** not temporally constrained into the future do
    If **TP**'s Xgoals are ready to start or **TP** is about to time out then (fire)
      For each Xgoal **G**@[**TP**,*] respectively do
        Send "start[**G**]" to **G**'s state variable's controller & estimator
        Inform **G**'s parent goals to start or stop monitoring **G**'s status