State Analysis

A Control Architecture View of Systems Engineering

Robert D. Rasmussen, Ph.D.

Jet Propulsion Laboratory

The information contained in this document has been designated by the California Institute of Technology (Caltech) as Technology and Software Publicly Available (TSPA). Copyright 2005. The copyrights and patents related to this technology are owned by Caltech. United States Government sponsorship acknowledged.

NOTICE: The Adaptable State Based Control System is patented (US 6,745,089) to the California Institute of Technology (Caltech). Should you be interested in implementing and using the system, methods associated with implementing the system, or in obtaining software tools or code to implement the system or method, you should obtain a license from the Office of Technology Transfer at the Caltech. The distribution of this training material does not constitute a license to practice the patent or to use any tools or software code.

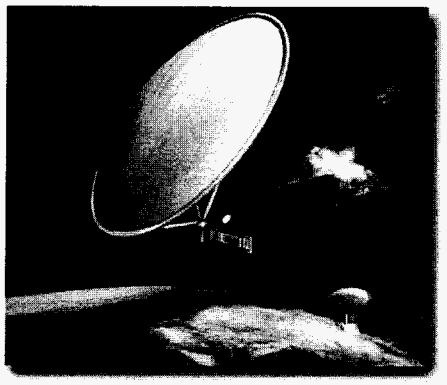
Topics

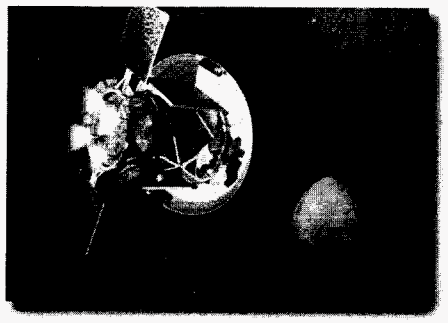
- Issues with growing complexity
- Limits of common practice
- Exploiting a control point of view
- A glimpse at the **State Analysis** process
- Synergy with model-based systems engineering
- Bridging the systems to software gap

Growing Complexity

- Elaborate and demanding missions
- Intricate interacting systems
- Full autonomy

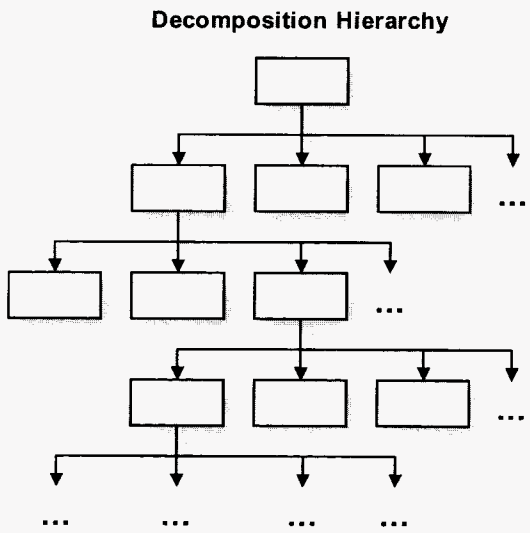
We need to improve our approach to addressing these issues





A Critical Look at
Functional Decomposition

- Concerned with division of delivery/design responsibility
  - What are the elements of a system? How are they composed?
  - What is the nature and purpose of the interfaces among system elements?
  - What function does the system serve?
  - Can this function be done with simpler functions working together?
  - How are these functions allocated among the elements of the system?
- The process is hierarchical
  - Successive levels of exposition
  - Encapsulation is a central concept
  - Interfaces support collaboration
  - Works best with weak coupling
- Embodies the classic "V" model
  - Partition (i.e., disintegrate)
  - Specify, analyze, build, …
  - Then reintegrate



Limits of Functional Decomposition

- The idealism of encapsulation, weak coupling, and controlled interfaces is often broken
  - Environmental coupling
  - Unintended side effects
  - Design compromises or errors
  - Operability oversights
  - Degradation and failure modes
  - Creative appropriation for other uses
- "Integration" is the process of dealing with such issues
  - A frequently ad hoc catch-all for whatever turns up
  - Systems engineers valued as problem solvers rather than problem avoiders
- Consequences:
  - Late software
  - Reduced capability
  - More complex mission operations
  - Lower reliability

A Control Point of View

- Complexity is mainly a control problem
  - Planning and executing elaborate scenarios
  - Coordinating broad and complicated interactions
- Control problems appear in a handful of recurring patterns, such as…
  - Interpreting varied and conflicting data
  - Managing resources and side effects
  - Distinguishing success from failure
  - Handling the unexpected

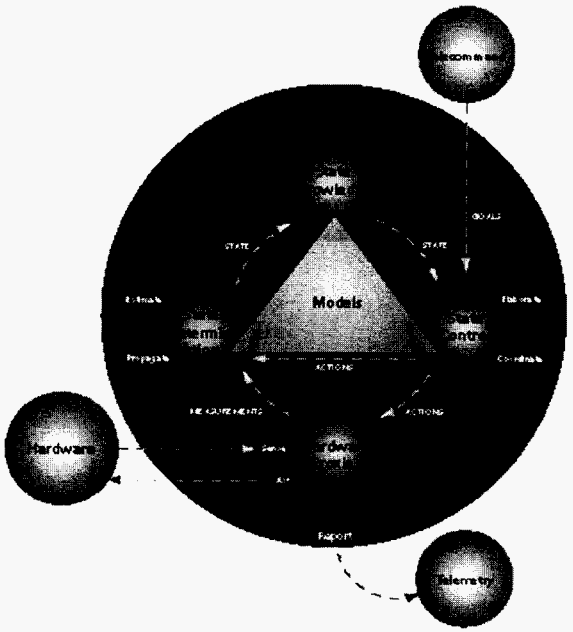
    and so on
- Control is only as good as your understanding of the system under control

How can we use this to our advantage?

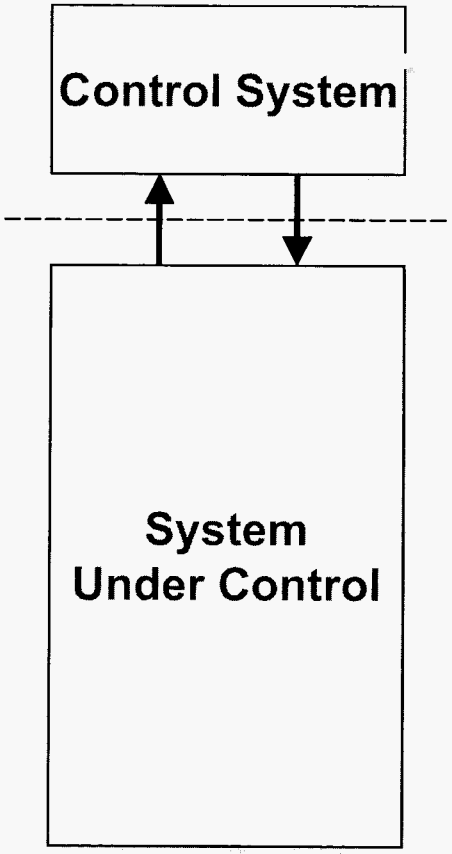
The Basic Ideas of State Analysis

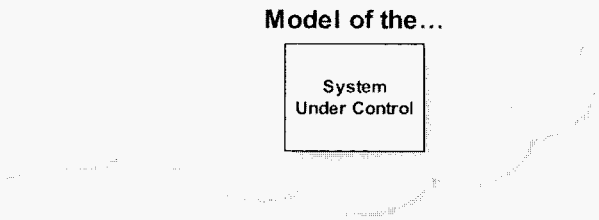
- Define a **principled control architecture** that addresses common control issues
- Build **functions from architecture** — not the other way around
- Make sure all **functions directly reflect models** of the system under control

- Essential properties of our control architecture:
  - **Regular structure** capturing a complete pattern of control
  - **State-based** to cast all issues in common terms used by models
  - **Goal-driven** to support objective reasoning about operational intent



The Importance of Models





- To understand the control system

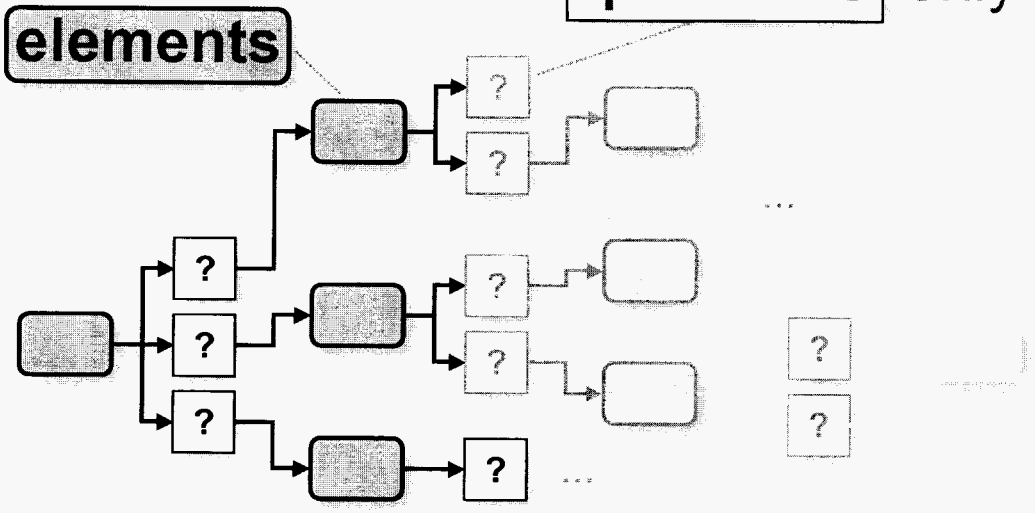
  … what it needs to do,
  what it needs to be…

  you need to carefully define the system under control

  and exploit your understanding of it in terms of **models**

The State Analysis Process

- Define a few basic types of **canonical elements**, enabled by adopting a regular architectural structure
- Pose **a standard set of questions** for each element type
- Express answers to these **questions** only in terms of other **elements**



- Each answer prompts additional questions, unfolding a step at a time until all the elements are identified

Some of the State Analysis Questions*

A closely integrated set of control concepts

States

- What **state variables** describe **knowledge** of the system under control?
- How do they represent variation in time, uncertainty, etc.?
- To what extent must they describe past and future?
- How are they stored and/or transported?

Constraints Networks

- What type of **state constraints** must the control system be able to enforce?
- How should state constraints be elaborated?
- What coordination information is needed for scheduling (using **time constraints**)?
- How is constraint execution regulated and monitored?

State-based Models

- How do states of the system under control behave and affect one another?
- How should constrained behavior be projected?
- How do commands affect states, nominally or otherwise?
- In what way do measurements depend on states?

Achievers

- How are models used by **estimators** to determine state knowledge from potentially inconsistent or uncertain evidence provided by measurements and commands, and by other state knowledge?
- How are models used by **controllers** to determine what commands to issue, given uncertain state knowledge?
- How is real time coordination accomplished?

Deployments

- How are control system functions deployed across the system?
- How can computing and communication attributes be used to assign responsibilities among **deployments**?

Measurements and Commands

- What are the measurements and commands at the interface to the system under control?
- How does the system under control manage them?
- How are they stored and/or transported?

** simplified !*

Following Leads — An Example

Standard Questions:

What do you want to achieve?
Move rover to rock

What's the state to be controlled?
Rover position relative to rock

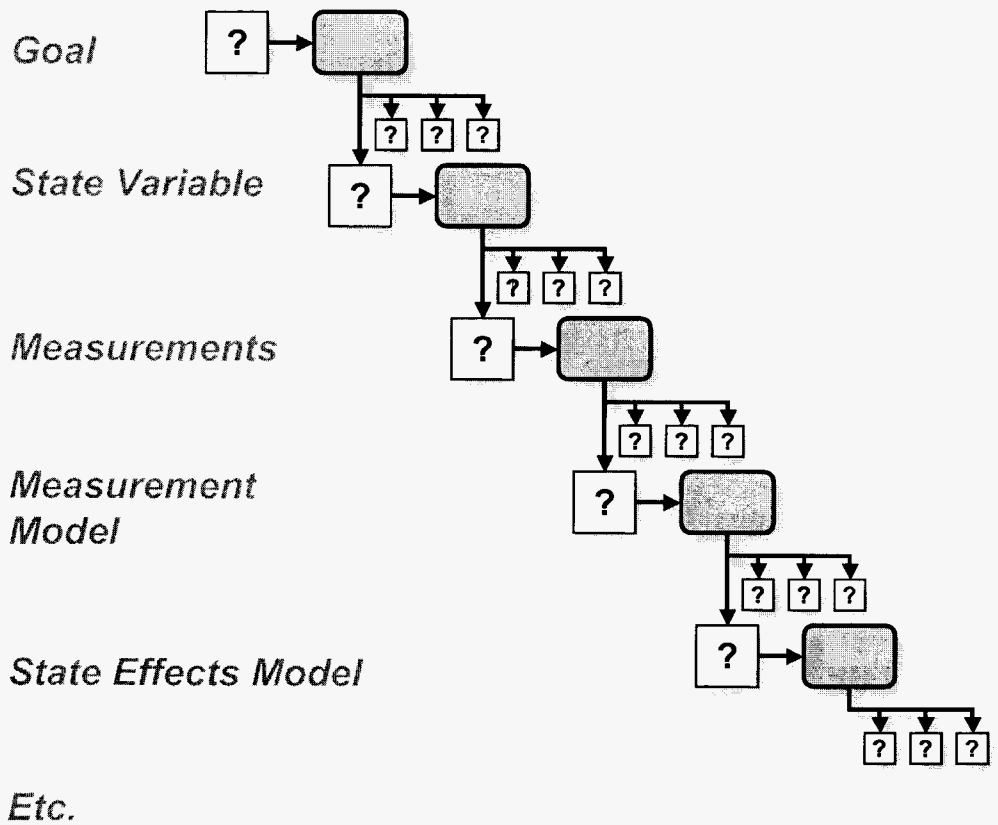
What evidence is there for that state?
IMU, wheel rotations, sun sensor, stereo camera

What does the stereo camera measure?
Distance to terrain features, light level, camera power (ON/OFF), camera health

How do you raise the light level?
Wait until the sun is up

Where is sun relative to horizon?
...

Common Elements:

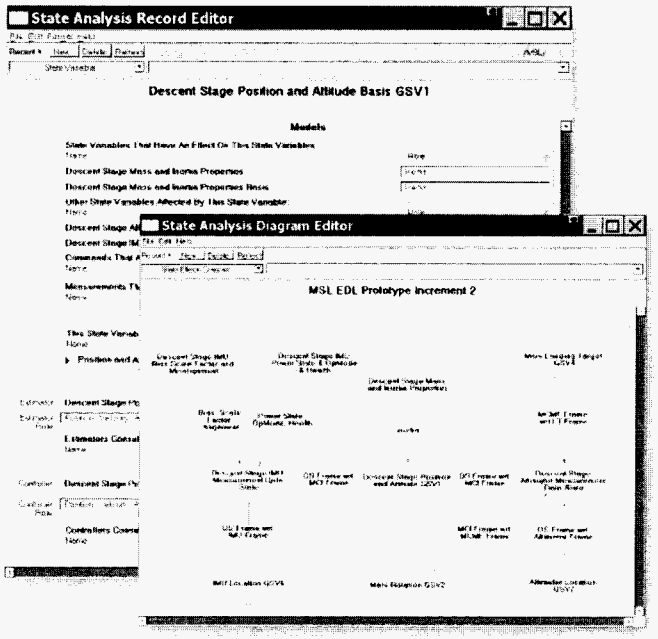


State Analysis
Produces Many Products

- Descriptions of state analysis elements become…
  - Behavior (functional) requirements on the system under control (hardware, data management, etc.)
  - Software component and algorithm requirements
  - End-to-end information flow requirements
  - Interface requirements among software components
  - Interface requirements between software and hardware
  - Operations requirements on system capabilities and constraints
  - Requirements on the deployment structure
  - Documentation of the assumed environment
  - Scenario descriptions

    … and so on

- These are presently captured in a "State Database" ➡
  - Structured to prompt the state analysis process
  - Shared, central, and globally accessible to promote consistency
  - Capable of generating a variety of reports
  - Accessible by a variety of tools, including multi-platform client



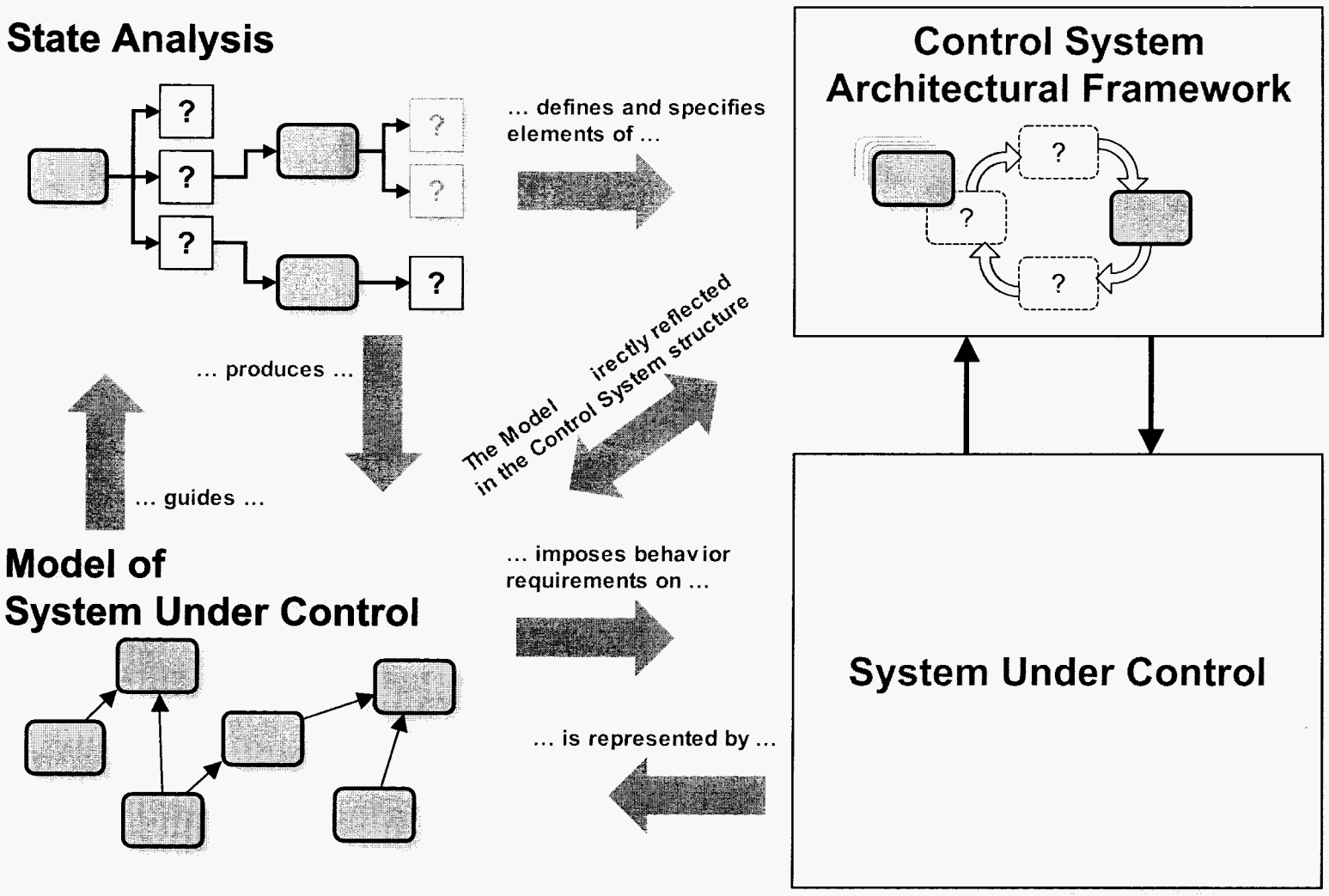
A Shorter Path from Systems Engineering to Software

- The general State Analysis control architecture can be captured in software frameworks
  - If so, every State Analysis element will have a direct counterpart in the software implementation
- Lots of advantages:
  - Early, …
    - To develop thorough, unambiguous **requirements**
    - To establish a clear **work breakdown**
  - During development, …
    - To guide the collection of development **metrics**
    - To foster **robust design**
    - To enable principled **coordination** of the system
    - As an **integration** tool at many levels
    - To facilitate the introduction of increased **autonomy**
  - During validation
    - To cross-check **coverage** and **consistency**
    - To improve **inspectability** and **testability**
  - During operations, …
    - As an aid to **understanding**
    - To improve the **performance** of the system
  - Afterward, …
    - To enhance the **reusability** of your product

Also a Natural Partner in Model-Based Engineering

- One modeling effort — many customers:
  - Control system design, as described here
  - Behavioral specification of the system under control
  - Simulation for analysis and testing
  - Model-based reasoning for estimation, planning, and execution
  - Etc.
- Integrated easily into a larger model-based engineering context
  - Major overlap in required models
  - Use of common tools is possible
- A powerful adjunct to traditional functional decomposition

A Map of the Process



Essential Characteristics of State Analysis

- Concerned with the process of control
  - How do the things being controlled really work?
  - What do we want to happen, and when does it need to happen? Can it be done, and if so, what actions are appropriate? How do we coordinate competing desires?
  - How do we know if and when we have succeeded? What sorts of things can interfere with success? What should we do if we fail?
  - What do we need to know about the system to do all this?
- The process is comparatively "flat"
  - Deals with local peer-to-peer issues
  - Lets local influences (designed or otherwise) determine relationship hierarchies
- Directed more toward continuous integration
  - Discover and deal with issues early
  - Resolve them within a regular structure
  - Work operations scenarios in detail as part of the design process

State Analysis Status

- In development for several years at JPL
  - Basic tools, training, and documentation in place
  - Full range of core supporting software frameworks near completion
  - Applied successfully to small pilot projects
- Looking for a big test to prove its merits