University of Maryland

A SIMULATION APPROACH TO RISK-BASED DESIGN OF COMPLEX DYNAMIC SYSTEMS USED IN SPACE MISSIONS

M. Azarkhail
M. Modarres

Center for Risk and Reliability
University of Maryland

COPYRIGHT © 2007, M. Modarres

Presentation Outline

- Propulsion System Description
  - Thruster Assembly Design
  - Propulsion System Design
  - External Leakages
- Mission Time Profile
- Simulation Overview
- Simulation Flowchart
- Agent-Oriented Approach
  - Agent Definition
  - Why Distributing Intelligence?
  - Example
  - Hierarchy of Agents
- CCF Considerations
- Results
- Conclusions

Propulsion System Description

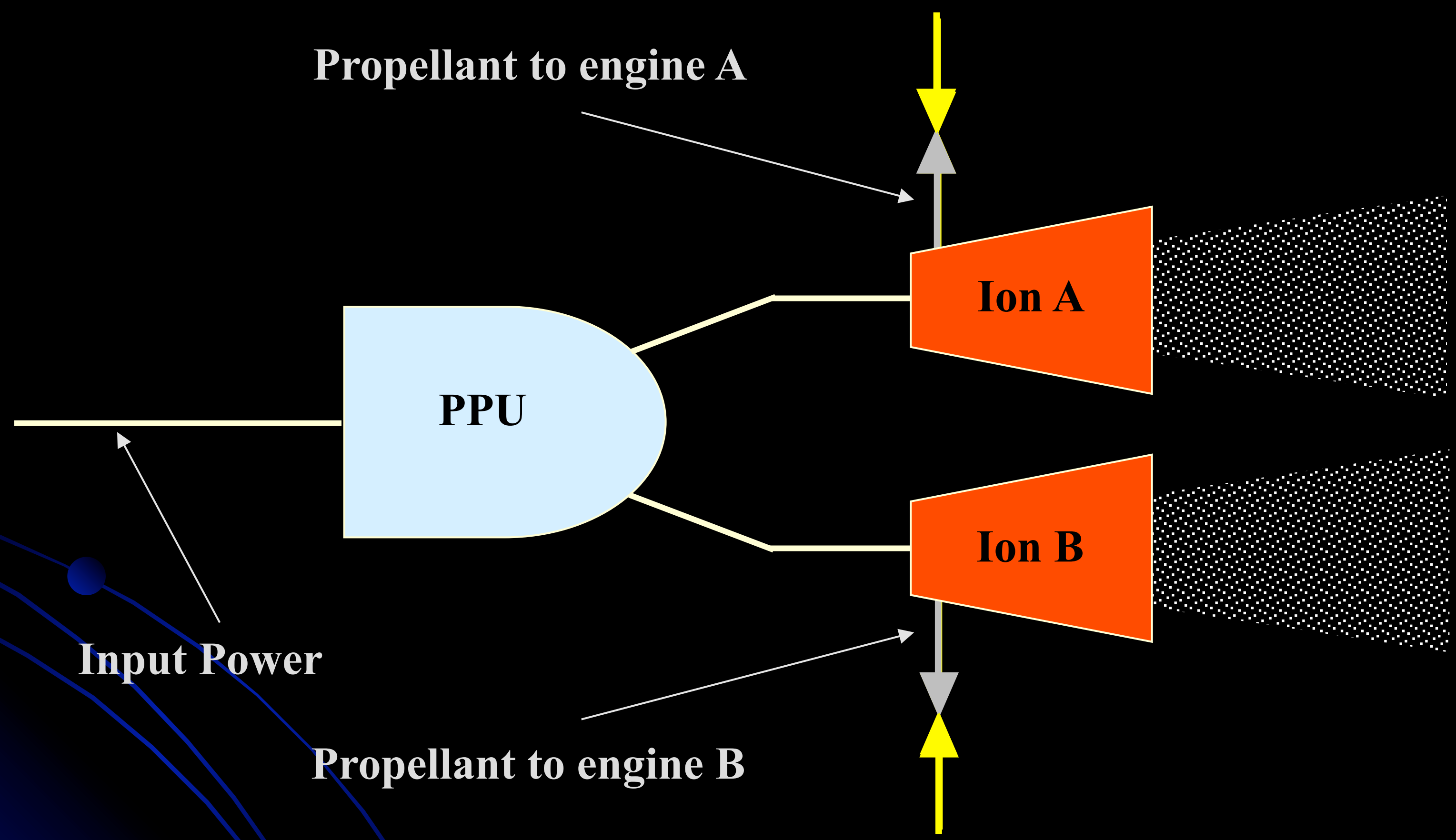
1. The system is needed for science missions at the outer solar system.
2. The system consists of five thruster assemblies and one propellant supply.
3. Each assembly has one propulsion power unit (PPU) and two ion engines (IE).
4. When an assembly is operating, the PPU provides power to just one ion engine.
5. The other engine remains on standby, unless failed.
6. In some phases the propulsion system only operates during part of the phase

Propulsion System Description (cont.)

7. Thruster needs 2 out of 4 assemblies during the first phase, and 3 out of 5 assemblies during the subsequent phases
8. The failed assembly is replaced by the lowest numbered standby assembly.
9. For example, in phase 1, in case of failure assembly 2 will be replaced by assembly number 3, and assembly number 4 becomes the available standby assembly
10. Mission fails if there are more than 2 failed assemblies

COPYRIGHT © 2007, M. Modarres

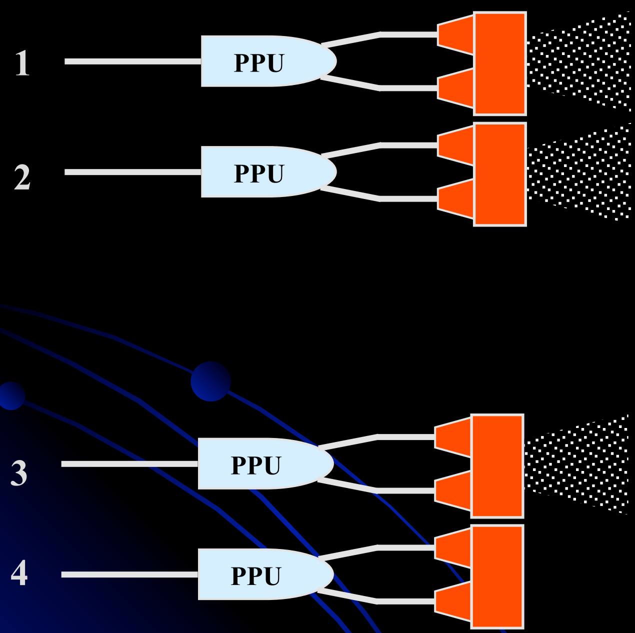
Thruster Assembly Design



COPYRIGHT © 2007, M. Modarres

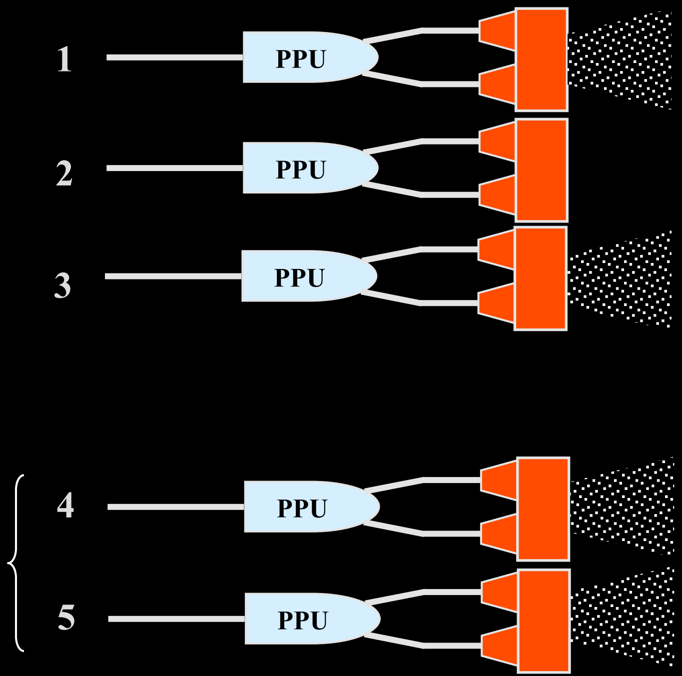
Propulsion System Design

Phase 1: 2 out of 4



Standby units

Phase 2 to 7: 3 out of 5

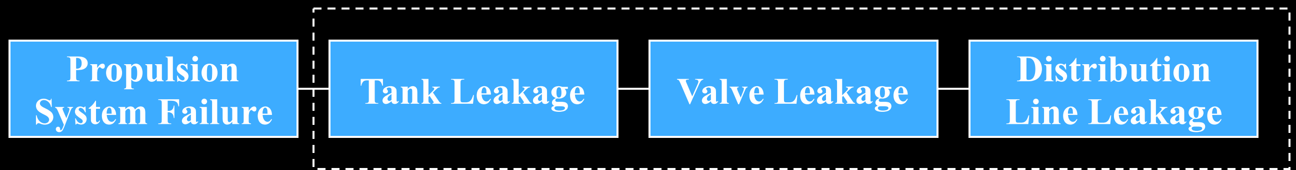


COPYRIGHT © 2007, M. Modarres

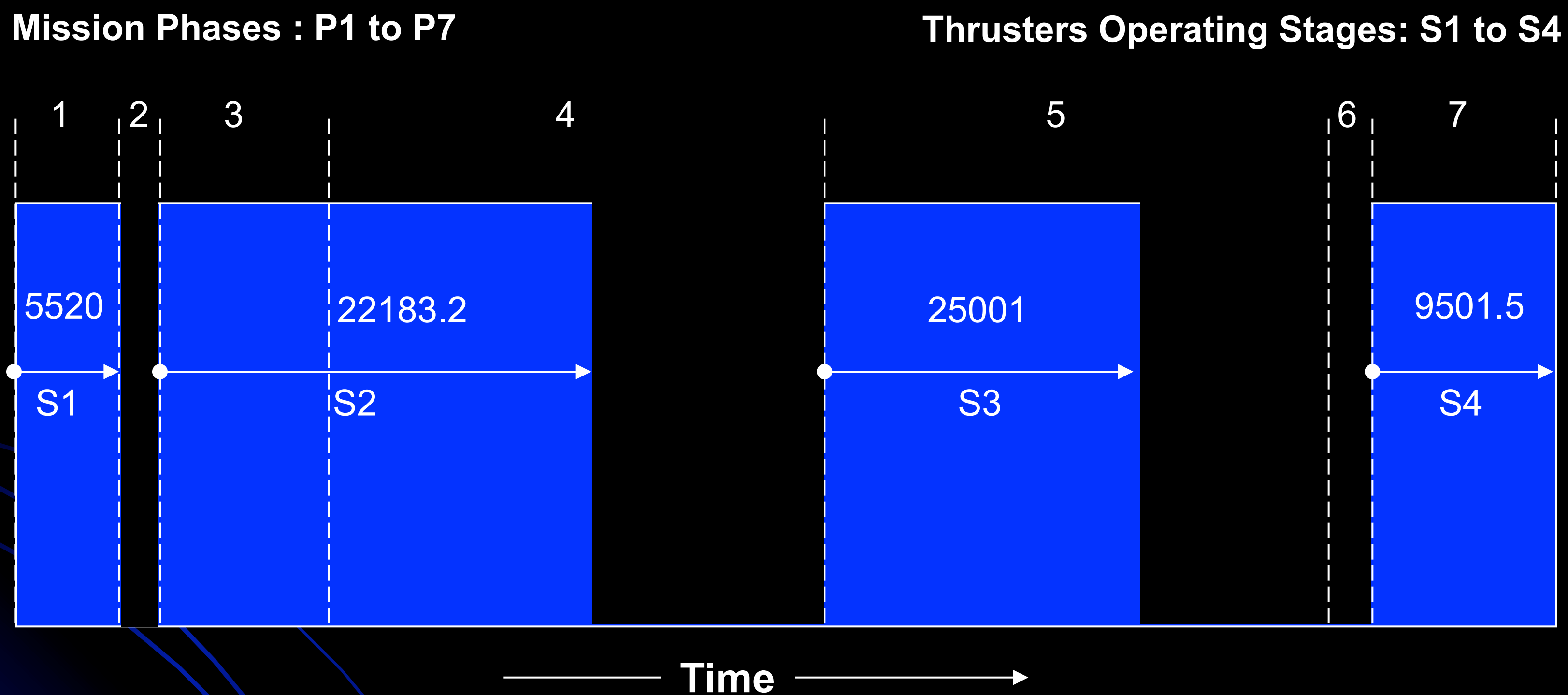
Center for Risk and Reliability
University of Maryland

External Leakage

Mission Failure:

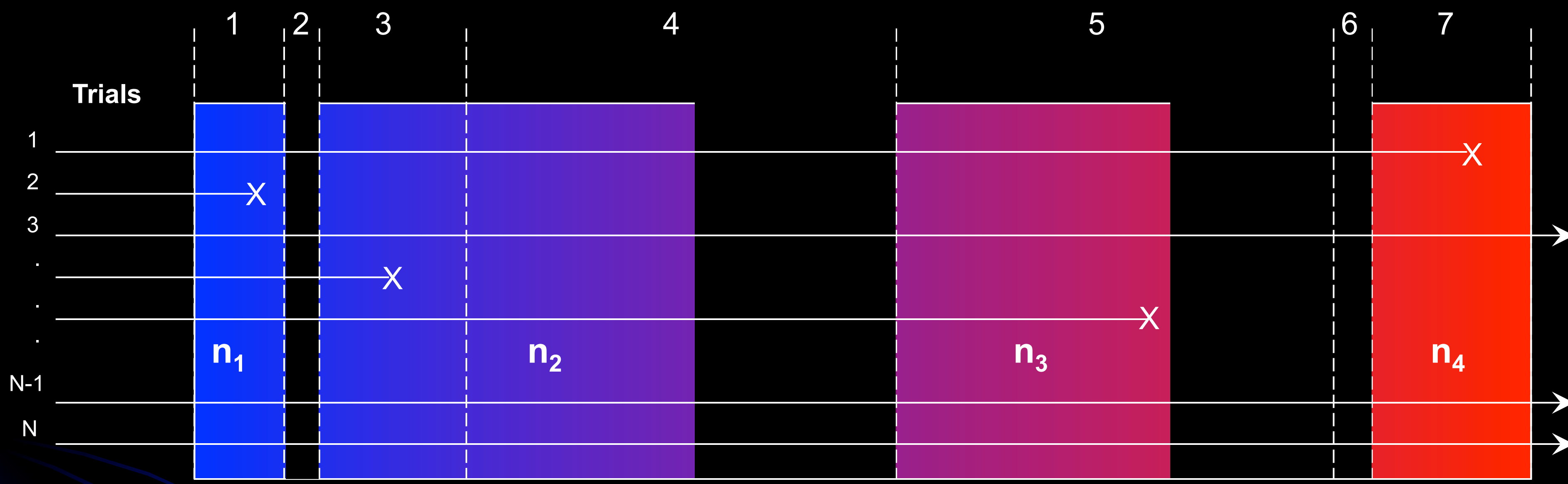


Mission Time Profile



Center for Risk and Reliability
University of Maryland

Simulation Overview



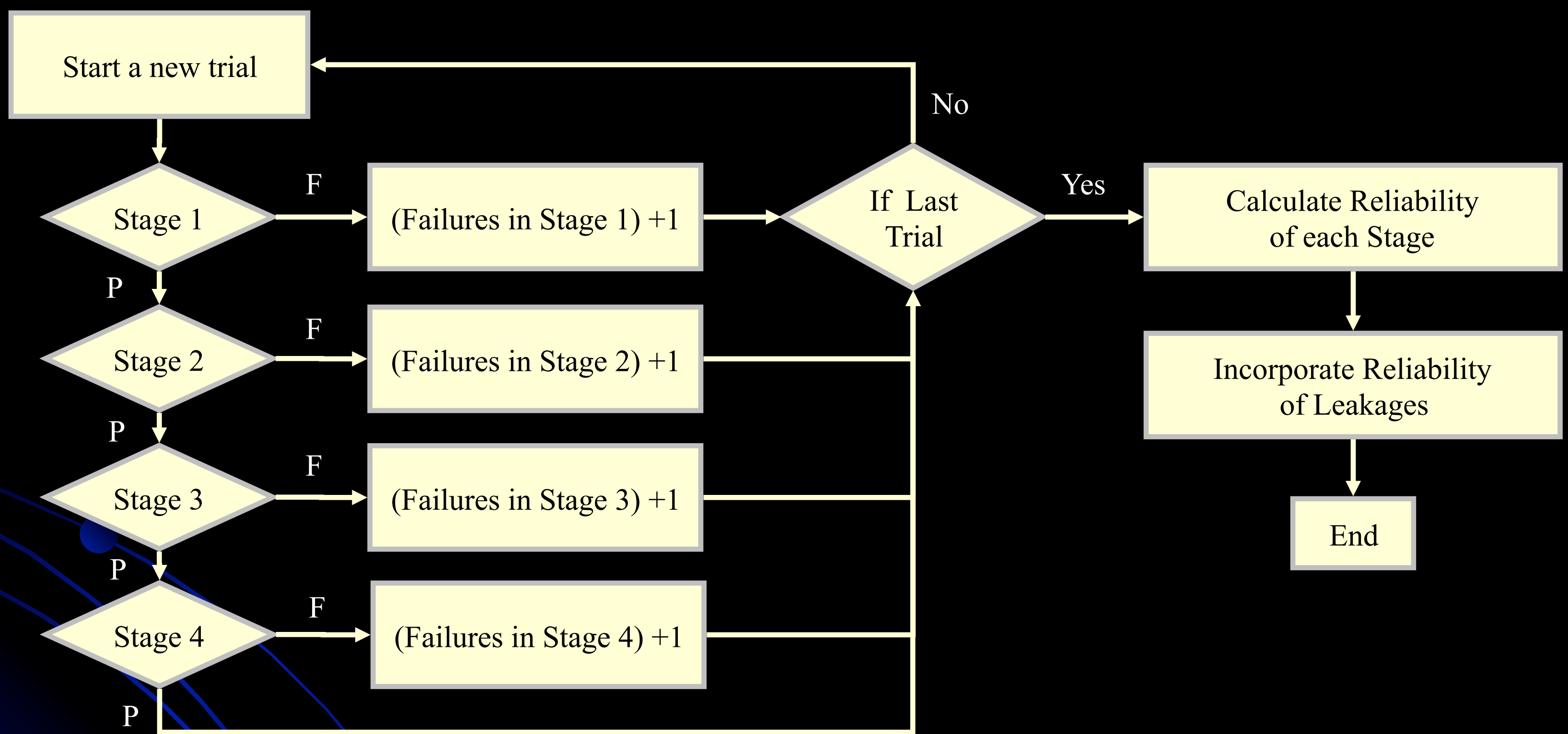
X : Failed Trials
→: Passed Trials
n_i: Number of Failures in Stage i
Ri: Reliability at Stage i

Time →

$$R_i = \frac{\text{Number of Survivals}}{\text{Total Number of Trials}} = \frac{N - \sum_{j=1}^{i} n_j}{N}$$

Center for Risk and Reliability
University of Maryland

Simulation Flow Chart



COPYRIGHT © 2007, M. Modarres

Center for Risk and Reliability
University of Maryland

Agent-Oriented Approach

- Every part is replaced by an intelligent piece of software
  - Contain all properties of the part (attributes)
  - Mimic the behaviors of the part (methods)
  - Able to communicate with other agents
- Inquiry is directed to an agent
- The inquiry is processed autonomously
- The process depends on agent status and boundary conditions
- The agent responds appropriately
  - The autonomous reaction is in form of either activity or information
  - The agent may contact, activate or request a task from other agents

COPYRIGHT © 2007, M. Modarres

Agent Definition

A piece of software capable of displaying

- Autonomy :
  - Capable of actions with no direct supervision
  - Have Some degree of control over its own actions ("self-activation")
- Reactivity :
  - Perceive their environment
  - Respond to the changes that occur in environment
- Pro-activity (goal orientation):
  - Act in a goal-directed manner
  - Take the initiative where appropriate
- Social activity (communication skills):
  - Interact when appropriate with other agents

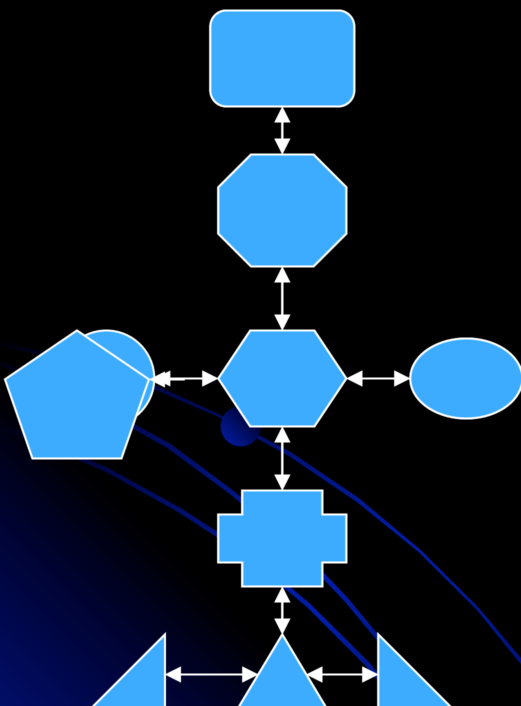
COPYRIGHT © 2007, M. Modarres

Simulation Model

- Dynamic vs. static
- Distributed vs. concentrated intelligence
- Simplicity vs. complexity
- Dependent vs. independent failures

COPYRIGHT © 2007, M. Modarres

Dynamic Vs. Static

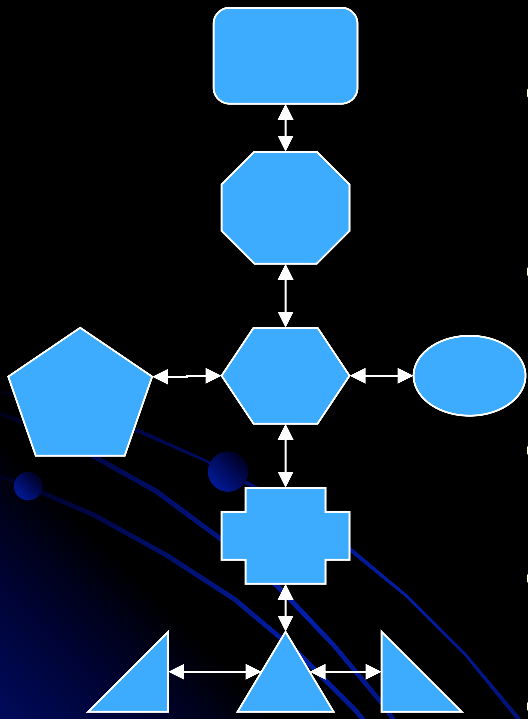


- Real time simulation instead of time snapshot model
  - Account for change in component properties
  - Account for changes in component behaviors
  - Account for changes in system configuration

COPYRIGHT © 2007, M. Modarres

Why Distributing Intelligence?

System



- Components of system are physically distributed
- Components and subsystems are heterogeneous in functionalities
- Hierarchical representation of the system force a distributed view
- Complexity of system forces local viewpoint
- Simulation model should be adaptable to changes
- Failure modes are autonomous and have their own persistent thread of control

Modeling procedure is a journey from

<u>Distributed system</u>

to

<u>Distributed intelligence</u>

COPYRIGHT © 2007, M. Modarres

Center for Risk and Reliability
University of Maryland

Simplicity vs. Complexity

- The failure logic of the system is simple
- Complexity comes from plurality of scenarios
- Group of components act the same therefore:
  - Individuals can be instances of the same class
  - Individuals inherit the properties and methods from their parents
- Rules are the same for individuals of a group, yet each component is autonomous in action
- The failure logic can be modeled in a higher level without getting involved in detailed scenarios

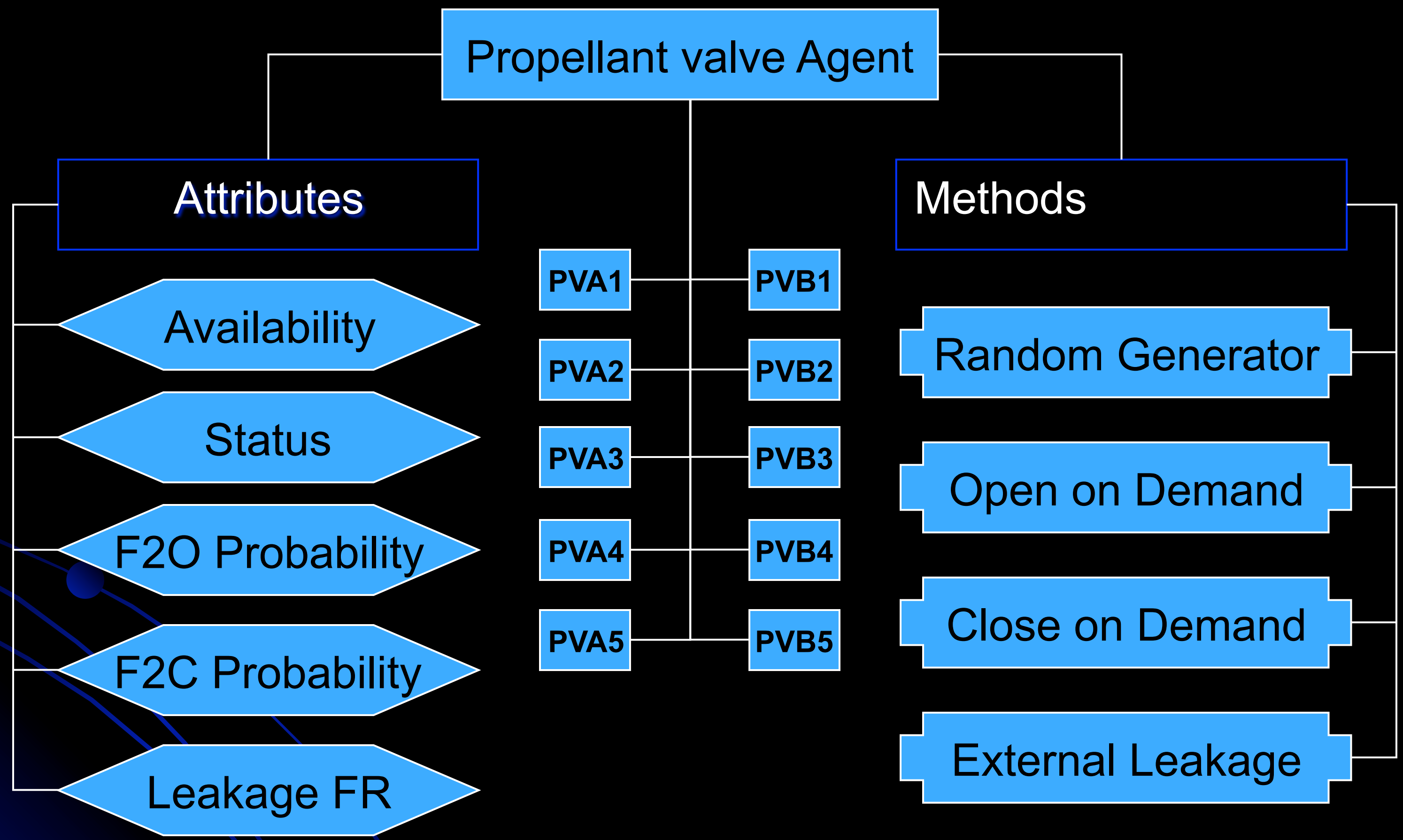
COPYRIGHT © 2007, M. Modarres

Dependent vs. independent

- Dependencies are traditionally added to the system model as extra independent events
- Direct simulation approach:
  - Make modeler able to assess the behavior of each object having the status of others (i.e. conditional probabilities)
  - Dependencies are modeled through communication of objects

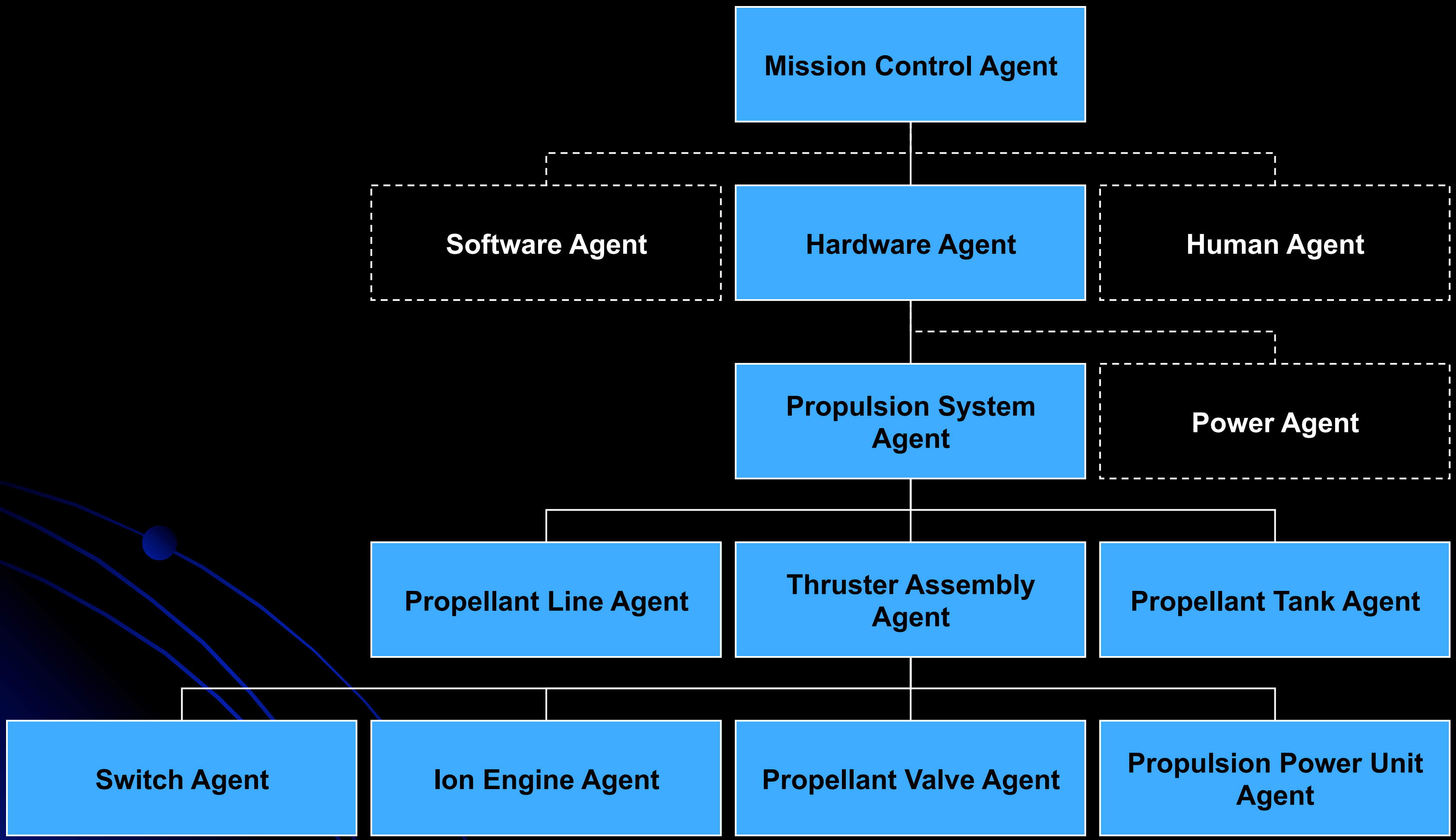
COPYRIGHT © 2007, M. Modarres

Example – Propellant Valve Agent



COPYRIGHT © 2007, M. Modarres

Hierarchy of Agents



Center for Risk and Reliability
University of Maryland

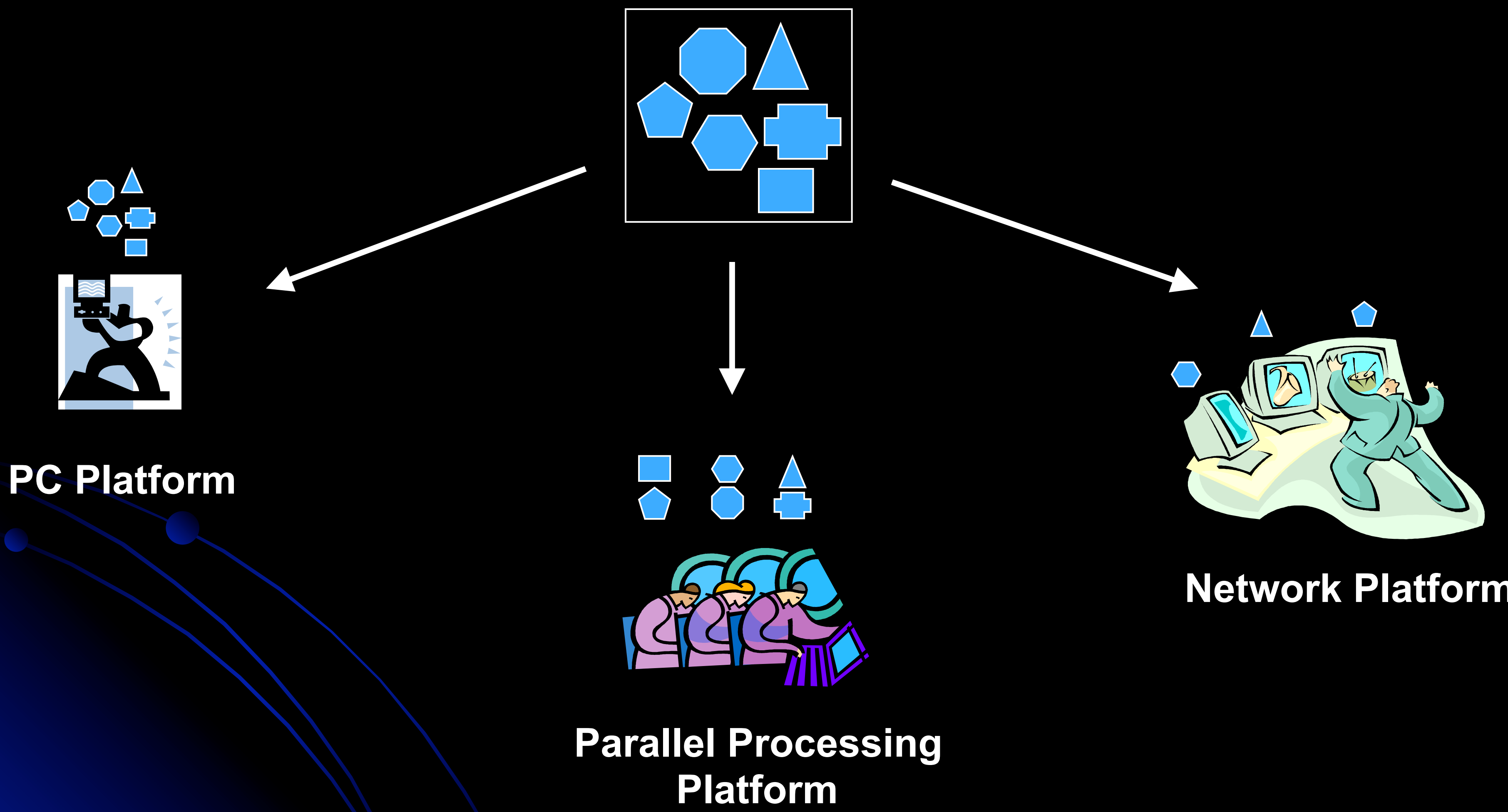
Feature 1: CCF Considerations

Group Size	Group Conditional Failure Probability (%)
2	8
3	4
4	2
5	1

- The CCF is applied using provided conditional probabilities
- The conditional probabilities are combined with simulation taking Monte Carlo sampling approach
  - Events are sampled in the order that are called by mission control agent
  - When a failure occurs, possible CCFs are identified and applied to the remaining components
  - An event once succeeds can not be a party of any CCF during the simulation
- Public attributes of agents (availability and status) are used to implement CCF

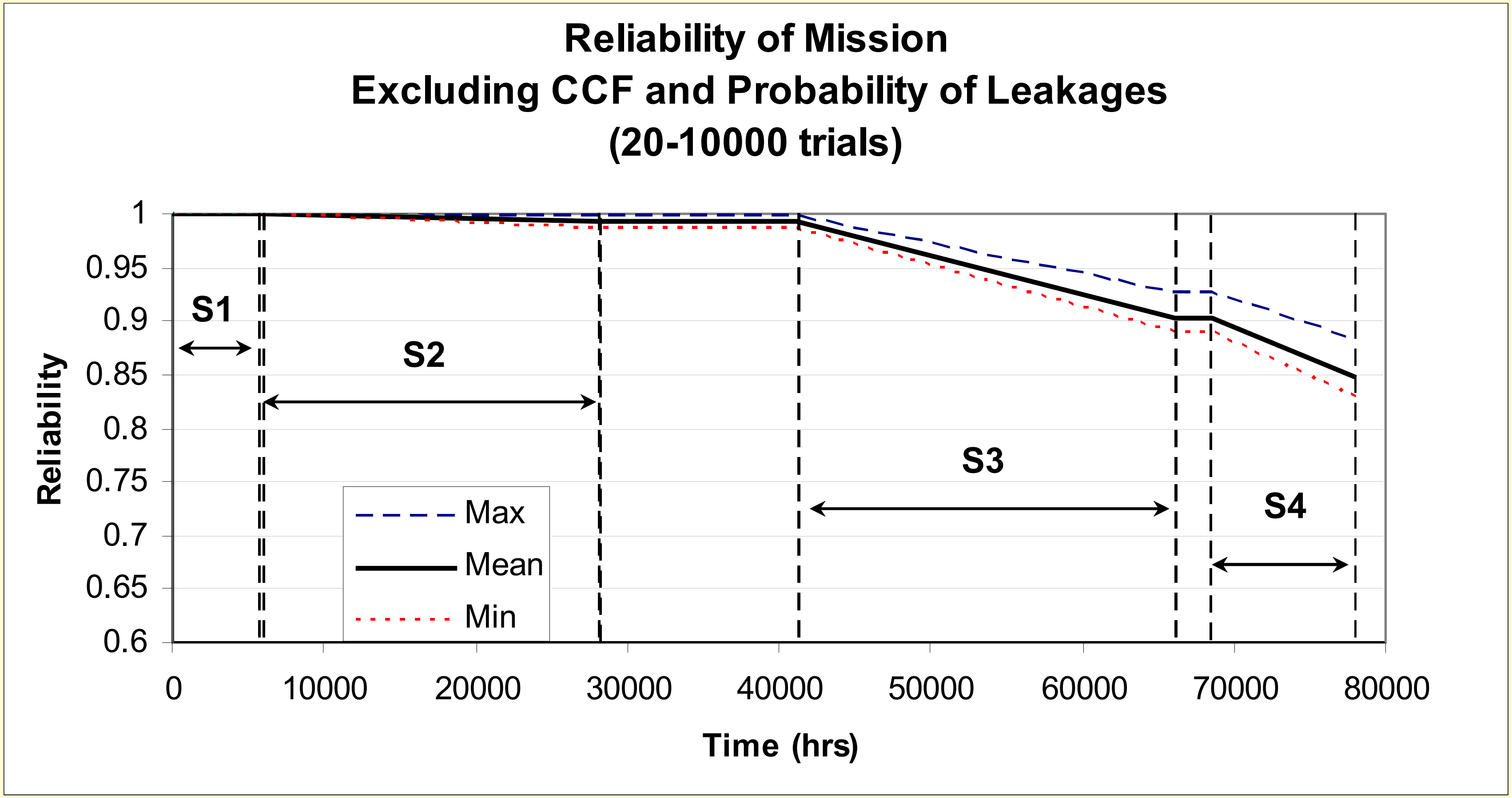
COPYRIGHT © 2007, M. Modarres

Feature 2: Computational Platform



COPYRIGHT © 2007, M. Modarres

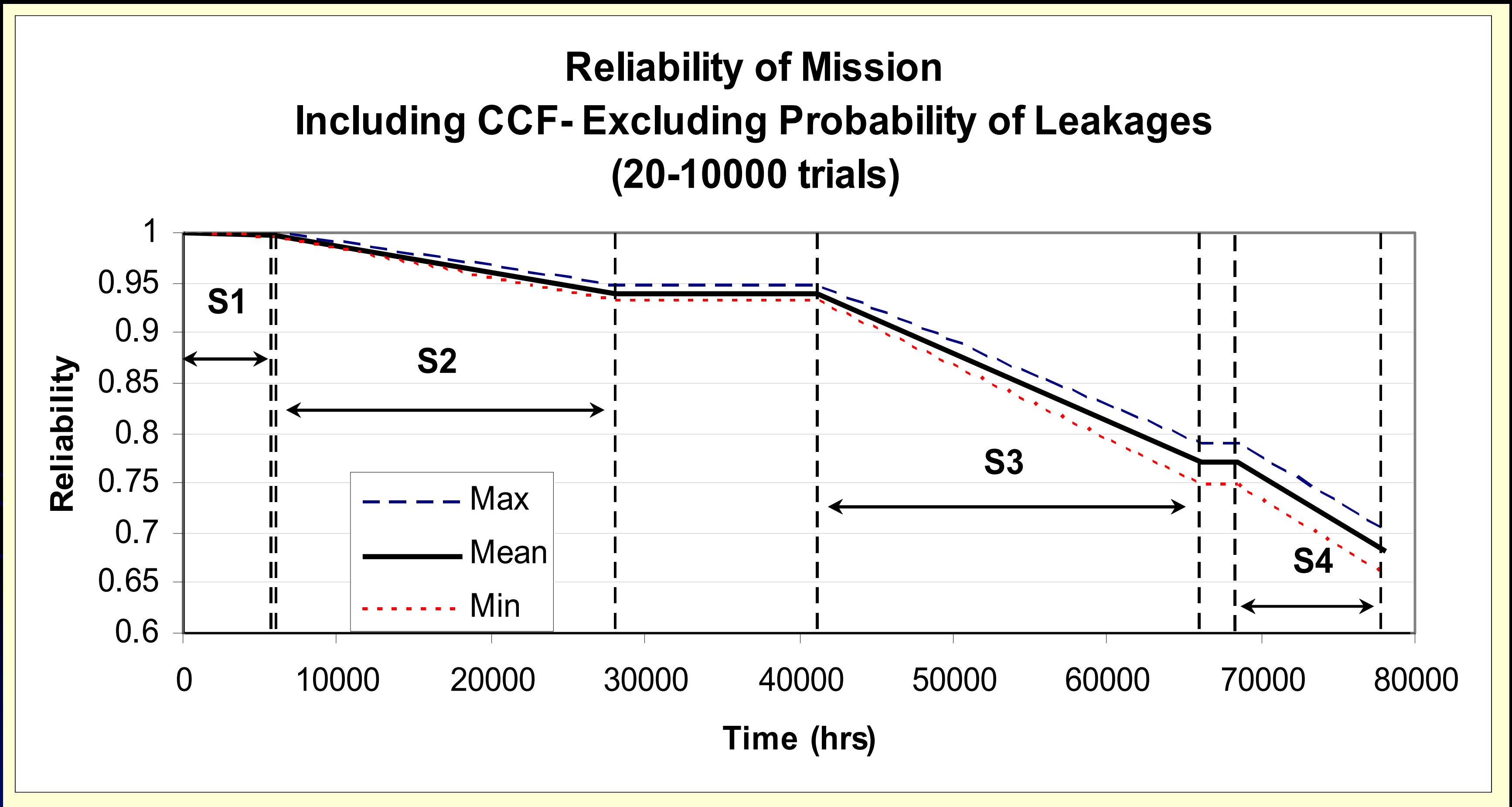
Mission Reliability
Excluding CCF & Leakages



COPYRIGHT © 2007, M. Modarres

Center for Risk and Reliability
University of Maryland

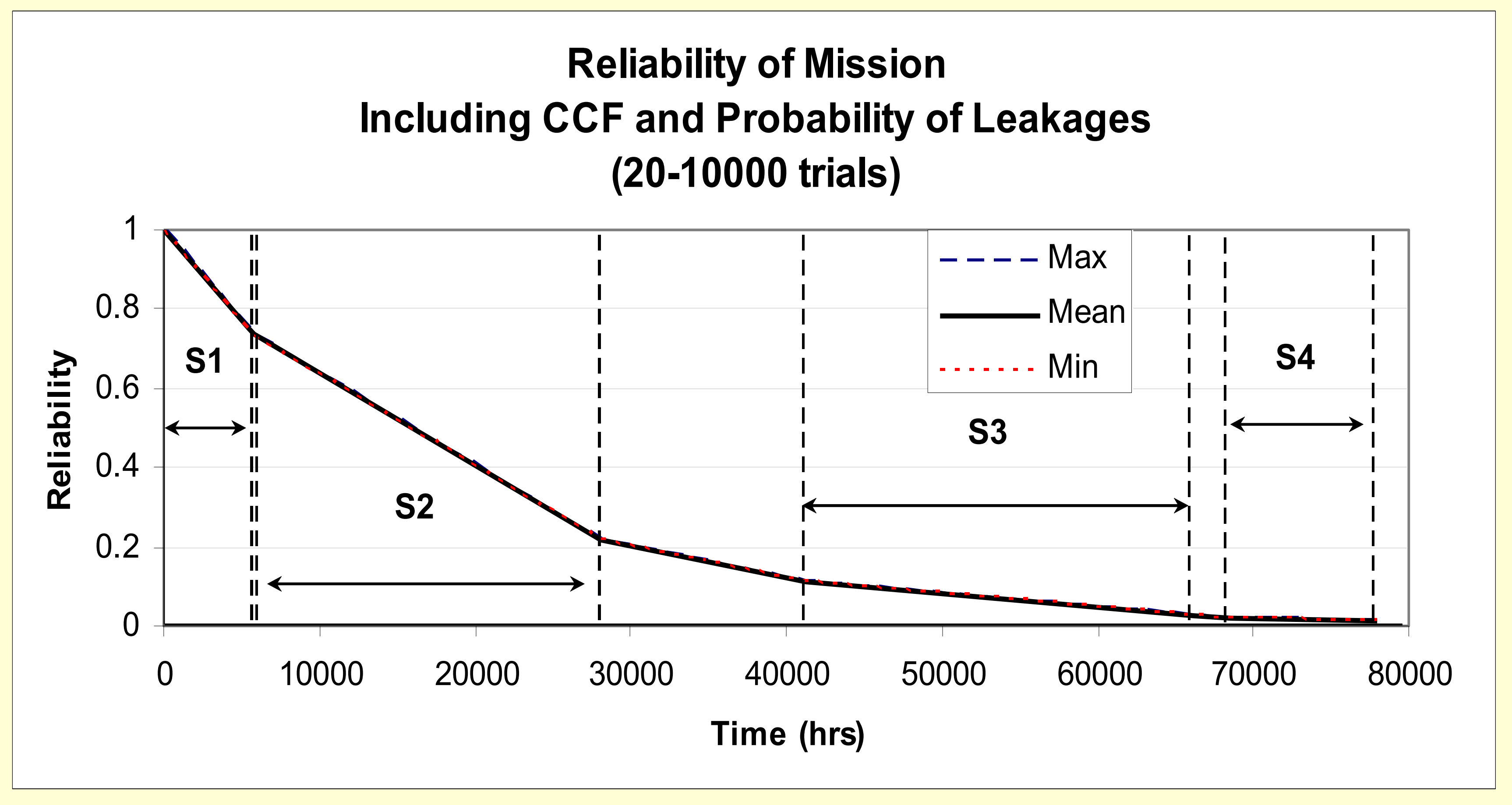
Mission Reliability
Including CCF



COPYRIGHT © 2007, M. Modarres

Center for Risk and Reliability
University of Maryland

Mission Reliability
Including CCF & Leakages



COPYRIGHT © 2007, M. Modarres

Center for Risk and Reliability
University of Maryland

Concluding Remarks

Taking this approach

- Dynamic behavior of the system is incorporated
- Complexity is reduced
  - Have a local view point to the system parts
  - Components respond autonomously without any supervision
  - Groups of scenarios can be modeled at a convenient level of details
- Dependencies (CCF) are modeled using communication skills of agents
- Modeler can use other processing resources in a parallel processing framework
- Mobility feature of agents makes the entire network a single computing platform for remote collaboration of agents

COPYRIGHT © 2007, M. Modarres

Thanks

COPYRIGHT © 2007, M. Modarres