



Design and Assurance of Control Software

Prof. Nancy Leveson

Aeronautics and Astronautics

MIT

Paper to appear in IEEE Trans. on Software Engineering (soon)

STAMP 2025 Workshop

- You've carefully thought out all the angles
- You've done it a thousand times
- It comes naturally to you
- You know what you're doing, it's what you've been trained to do your whole life.
- Nothing could possibly go wrong, right?

Think Again.



The Problem

Software

- is a major component in control of potentially dangerous systems
- Has grown enormously in size
- Requires high assurance and important properties, e.g., safety and security
- Sometimes requires regulatory certification before can be used

But

- Software assurance is very expensive and limited in power
- Difficult to operate, maintain, and evolve complex software within reasonable cost limits

The Goal

- Design safety, security, correctness, etc. into systems from the beginning
- Integrate assurance into development
- Enable easier maintenance, evolvability
- Enhance certification

Basic Software/System Design Principles

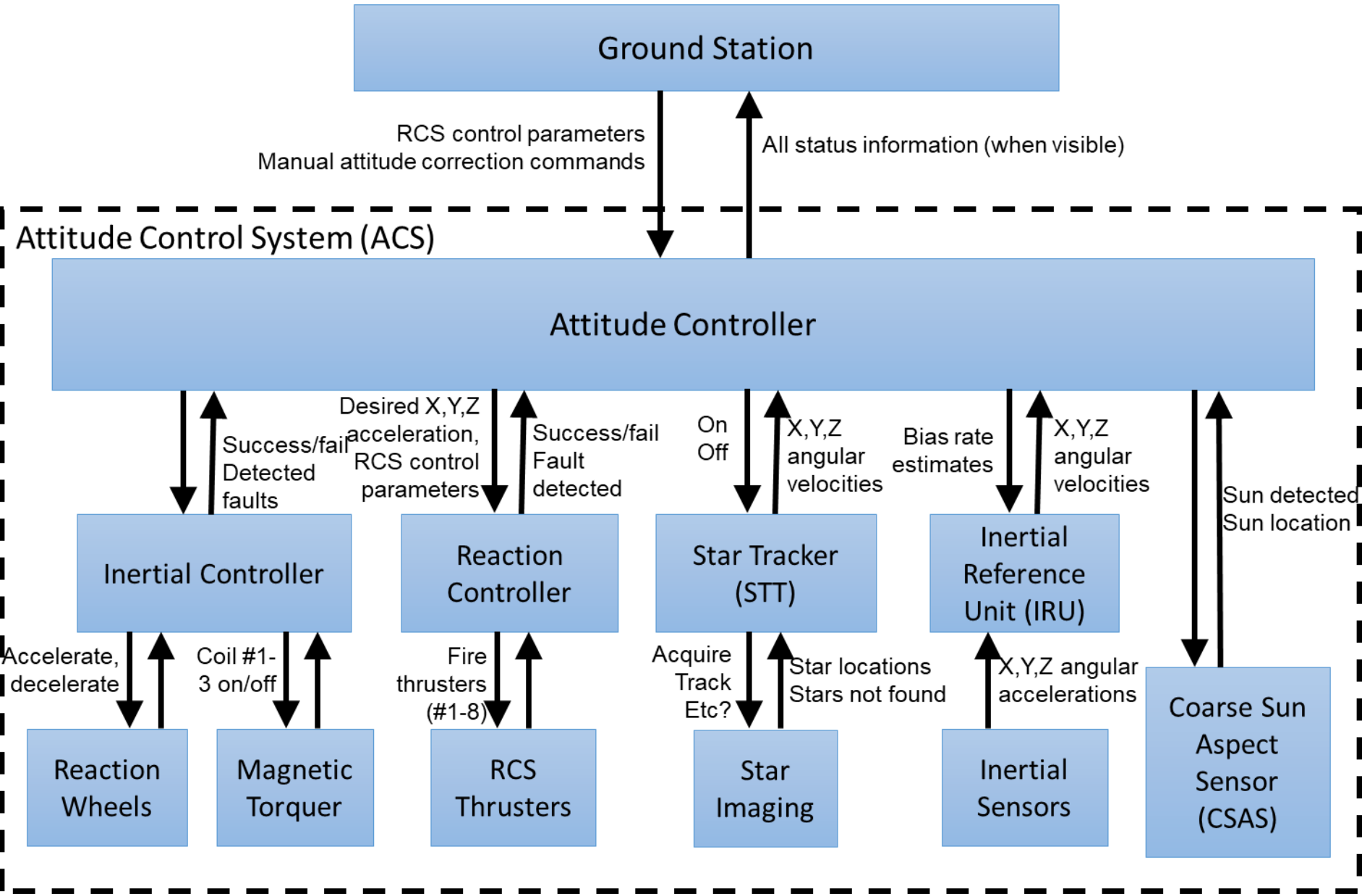
- Good design involves mastering complexity
- Design process should assist in understanding the problem to be solved
- Successful design depends on design strategies and specification methods used.

Design Approaches to Deal with Complexity

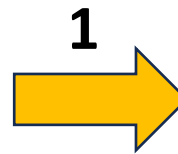
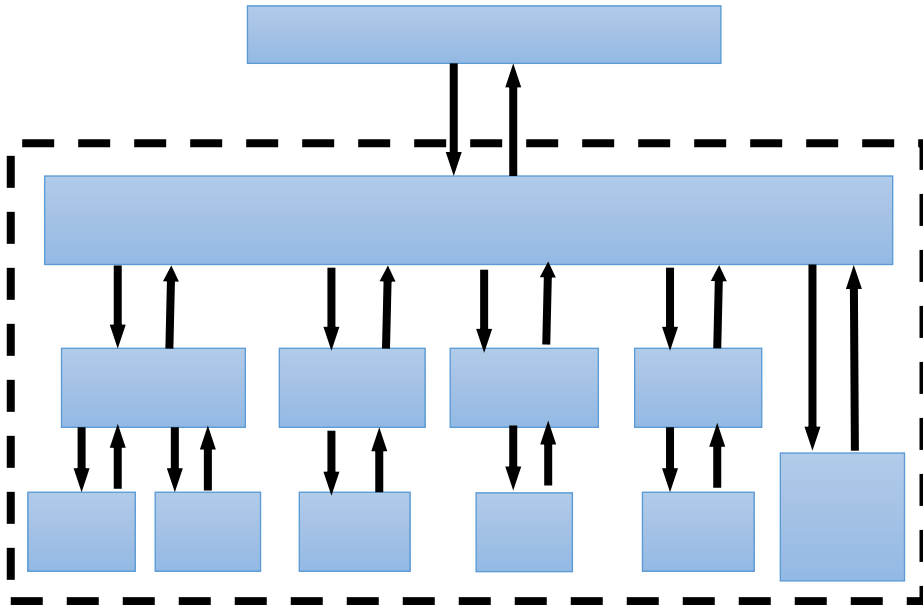
- Separation of concerns
- Restricted visibility (locality of information)
- Hierarchical Abstraction
- Simplicity
 - Semantic distance
- Proper ordering of design decisions

Summary of Design Principles

- Goal of design process is to master complexity, i.e., make the problem and solution intellectually manageable.
- Key to intellectual manageability is the structure of the artifacts used in the design process and the final design
- The design (solution) structure should
 - Match the problem structure (reduce semantic complexity)
 - Reflect mental models of users and operators
 - Augment our ability to produce user-centered designs: partnership between human factors experts and engineers focusing on physical and logical parts of the system

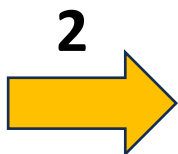


STAMP Model

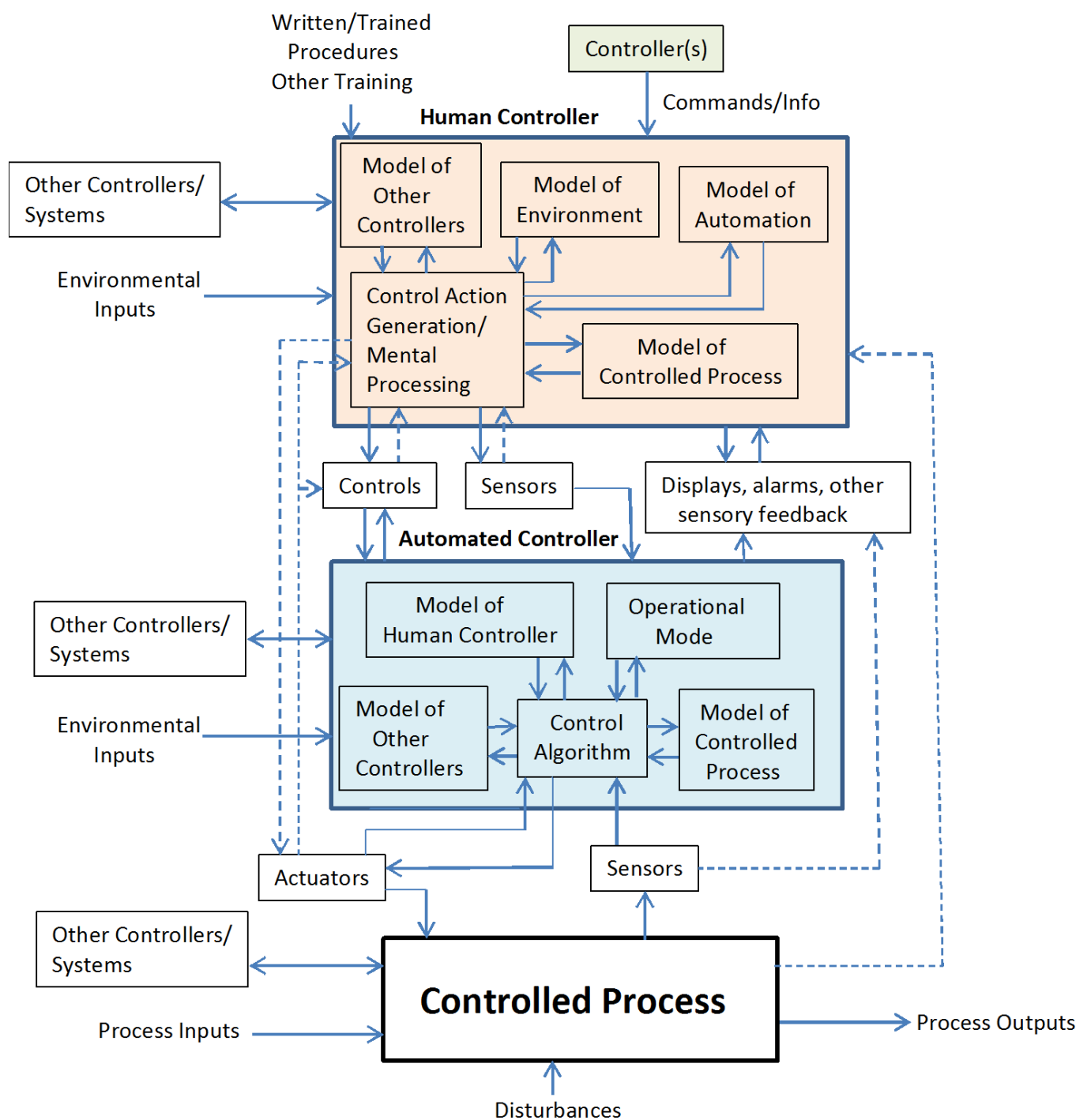


Software Architecture

- Client-Server
- Event-Driven
- Microkernel
- Pipe-Filter
- Blackboard
- Service-Based
- Peer-to-Peer
- Layered
- Broker
- Event-Bus
- ...



Example only



Potential Advantages

- System modeling and analysis (MBSE)
- Distributed assurance and development
- Assurance
- Certification
- Maintenance and Evolution
 - Changing software without introducing hazardous or undesired behavior
 - Reassessing and reassuring safety/security without enormous cost