Way Beyond SCADA: Compositionally Assured Systems

John Rushby

Computer Science Laboratory
SRI International
Menlo Park, California, USA

The Problem

"*I cannot have the confidence to trust local control unless I can see all the raw data*"

In other words, we cannot gain system-level understanding of current SCADA systems by understanding their components

"Security, security, security" (retrofitted)

In other words, components, subsystems, systems made wrong assumptions (or did not enforce assumptions) about their environment and interfaces (probably because the system architecture is accidental)

These are manifestations of noncompositionality

So a research agenda beyond SCADA needs to develop compositional methods for design and assurance

Separation of Concerns

- Should the controller design be coping with communication delays, loss, jitter, subsystem failures, topology changes?
- Or should the underlying computer system provide more isolation and actually guarantee certain properties?
- So the controller can build on those as assumptions
  - E.g., instead of sensor samples as timestamped points, and noise on failure
  - Intelligent sensor delivers samples as ranges, with "use by date"
- Wanted: a partnership between control theorists and computer scientists

(Non)Compositional Systems

- It's routine to build systems from components
  - Compositional design
- But often assume everything works to specification
  - And environment is benign
- System failures make no such assumptions
  - They do not observe interfaces
- So analysis for assurance is generally noncompositional
  - Ignores interfaces and dives into the design of components

Compositional Assurance

- Requires that assumptions at interfaces are guaranteed
- Either because they are very weak
  - So the system must be very strong
  - E.g., Byzantine-resilient architectures
- Or because they are enforced (credibly)
  - E.g., architectural frameworks such as TTA
  - Or operating system separation kernels (MILS)
- Interfaces talk about assurance properties, not just function
- Can then compose the system assurance analysis from properties of its components A Suggested Research Partnership Computer science and control engineering should work together to define properties to be guaranteed by the underlying computer and communication system
- SCADA needs different choices than, say, aerospace

- E.g., may need fault monitoring rather than masking

- Computer science develops compositional design, analysis, assurance for architectures to deliver those properties
  - Assurance technology is compositional formal verification
- Control engineers develop methods that deliver system-level properties in a compositional way using the properties delivered by the computer scientists
  - Desired properties are usually invariants, so require reachability analysis (i.e., hybrid systems verification)