Carnegie Mellon | Electrical & Computer ENGINEERING | Carnegie Mellon SCHOOL OF COMPUTER SCIENCE

Design & Analysis of Cyber-Physical System Architectures

Shang-Wen Cheng, David Garlan, Bruce Krogh, Akshay Rajhans, Bradley Schmerl and Bruno Sinopoli

Supported by Grant: CNS0834701

Objectives and Approach

- Blend disparate design and analysis approaches for software and physical systems into a unified approach for cyber-physical systems
- Extend the model structures and analyses from software architecture to cyber-physical systems
  - structural annotations to specify and check correct interconnections and interfaces
  - semantic annotations for formal analysis
  - design trade-offs at the architectural level
  - reuse recurring architectural patterns

Architectural primitives for cyber-physical systems

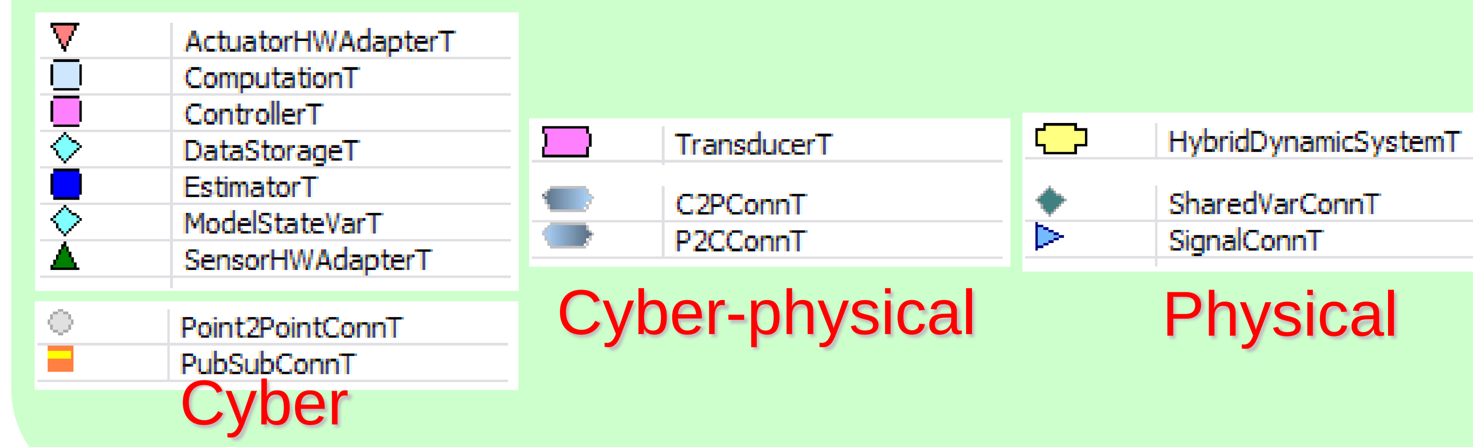
Components

- Cyber: computation, data-storage, controller, estimator
- Physical: hybrid dynamic system, physical subsystem
- Cyber-physical: transducer

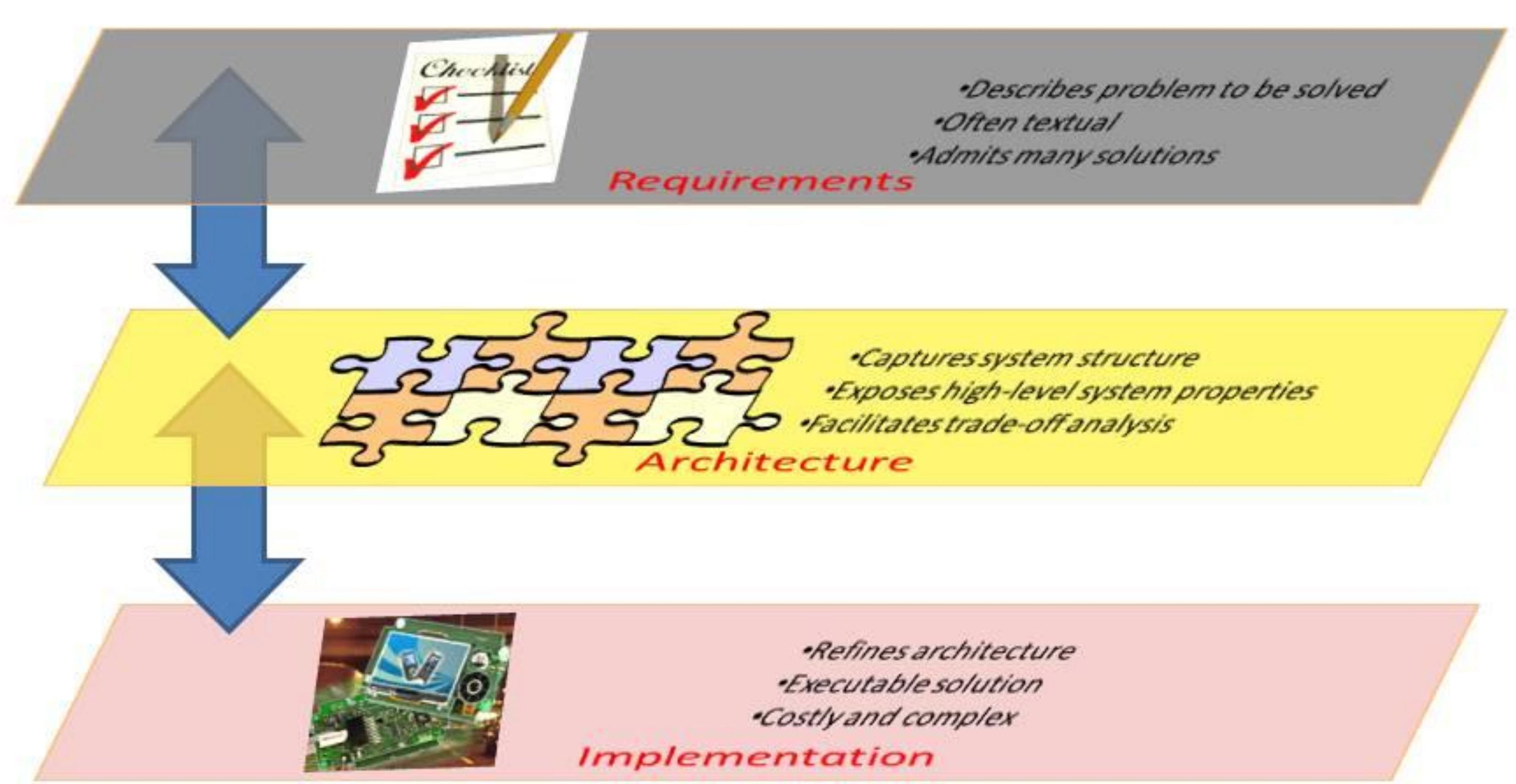
Connectors

- Cyber: point-to-point, publish-subscribe
- Physical: signal-flow (directed), shared-variable (undirected)
- Cyber-physical: cyber-to-physical, physical-to-cyber

Architectural types in AcmeStudio



Architectures



Software architecture

- Provides principled approach for design and analysis of software systems
- Well-established description languages, e.g., Acme
- Design environments, e.g., AcmeStudio

Modeling extensions for cyber-physical systems

- Properties of physical components and physical environments
- Compositions of physical elements
- Interfaces and interconnections between cyber and physical domains

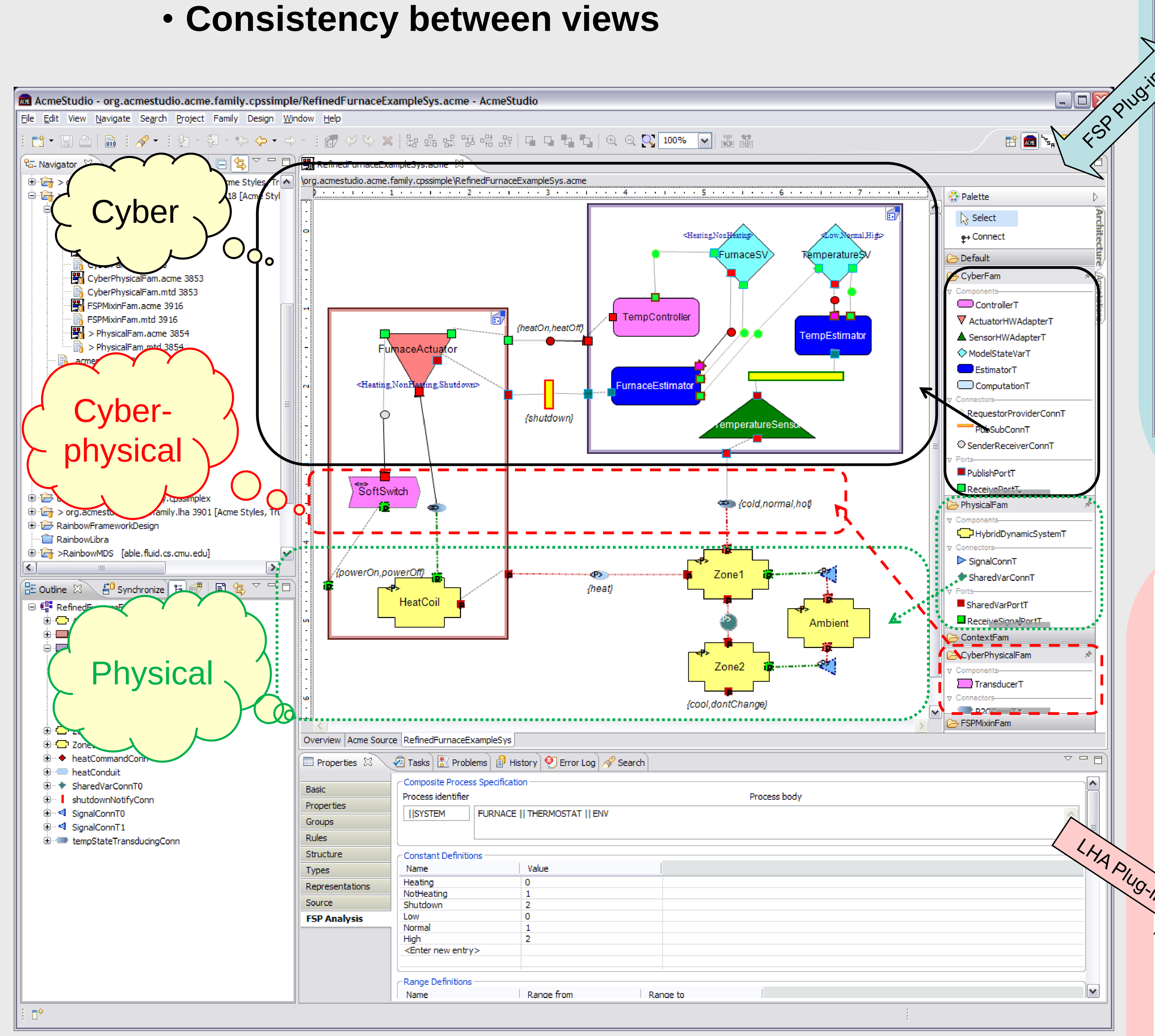
Analyses of cyber-physical architectures

- Correctness of dynamic behaviors
- Impact of communication on performance

Architectural and behavioral analysis

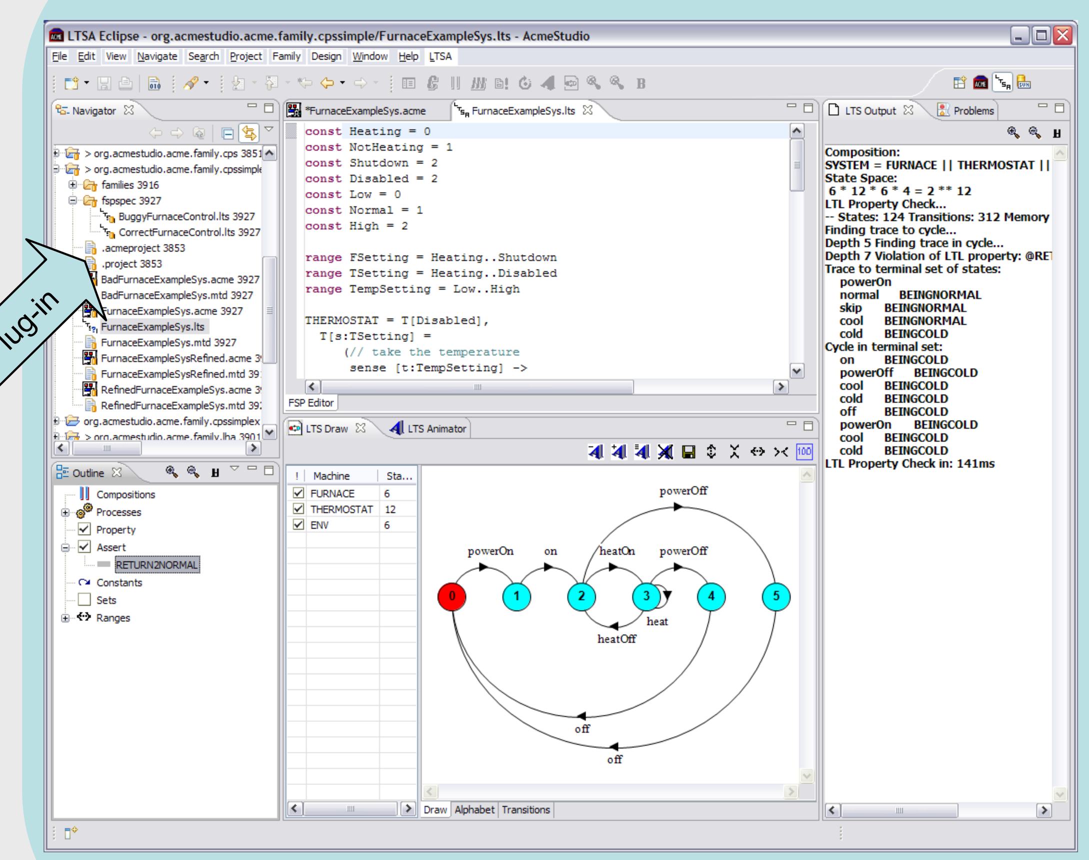
Architectural analysis

- Correct use of component and connector types
- Satisfies constraints over structure
- Required properties specified
- Consistency between views



AcmeStudio

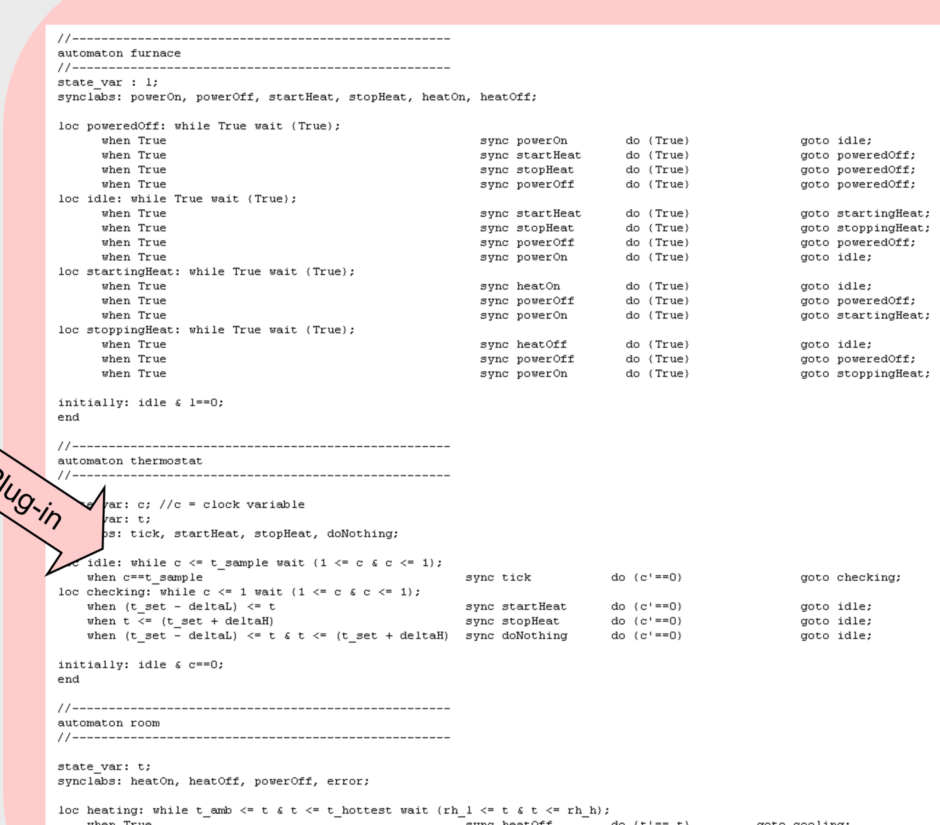
Behavioral analysis



LTSA tool

FSP Analysis:

- **L**abelled **T**ransition **S**ystem **A**nalyser
- Safety properties, e.g., *Temperature never exceeds max value*
- Liveness properties, e.g., *Temperature eventually becomes normal whenever it gets cold*
- Protocol checking, e.g., deadlock-freedom



PHAVer code

LHA Analysis

- **P**olyhedral **H**ybrid **A**utomaton **Ver**ifyer
- Richer set of hybrid dynamics, e.g., *Temperature is a continuous variable*
- Richer specification language - specification itself can be an LHA, e.g., *expected_behavior automaton*
- Simulation relation checking, e.g., *system does at least as much as what is required by the expected_ behavior automaton*