Outline

- Embedded Software/Systems Research
  - Introduction
  - Current & Future
- Q&A
Introduction [7]

- Trend of Change in Embedded Software
  - More Complex
    - Customers’ increasing demand for more elaborate functionality
  - Modular
  - Adaptive
    - Downloadable modules that dynamically reconfigure the system
  - Network Aware
Introduction (Cont’d)

Problems of Using Existing Techniques

- Required Domain Expertise
  - Processing sensor data or controlling actuators
- Extravagant Use of Hardware Resources
  - Layers of abstraction, elaborate algorithms, or statistical optimization
- Ad Hoc Definition of Components (Modules)
- Static Role of Components
- Unsophisticated Framework

Mechanism by Which Components Interact
Introduction (Cont’d)

Problems of Using Existing Techniques

- Use of Subroutines
  - Finite computations
    - Taking predefined arguments & producing finite results
- Use of Processes & Threads for Concurrency
  - Not easily characterizable aggregate
- Mismatched Assumptions about the Role of Time
  - Reducing time to a total order of discrete events
- Varying Communication Bandwidth & Latencies

Need for a Metaframework Dealing with Time & Concurrency

Not Suitable for Nonterminating Computation Transforming an Unbounded Stream of Data; e.g., Speech Coder
Introduction (Cont’d)

Metaframework

- Mixing frameworks hierarchically
  - A component in one framework being an aggregate of components in another
    - Domain polymorphism
      - Domain polymorphic component
      - Domain polymorphic interface that an aggregate of components exposes

Operating in Multiple Domains with Clear Semantics

Software Infrastructure with Which a Framework Is Realized
Reference