Model-Based Development of Embedded Real-Time Systems

Raj Rajkumar with Dionisio de Niz
Real-Time and Multimedia Systems Laboratory
Carnegie Mellon University
raj@ece.cmu.edu
http://www.ece.cmu.edu/~raj

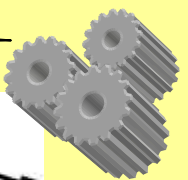
Scope of Interest

Functional

Reliability

Tolerate Faults

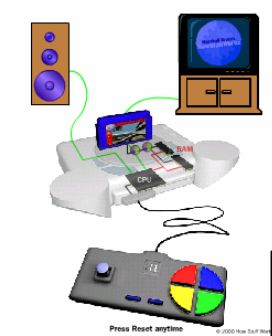
•Data elements per second



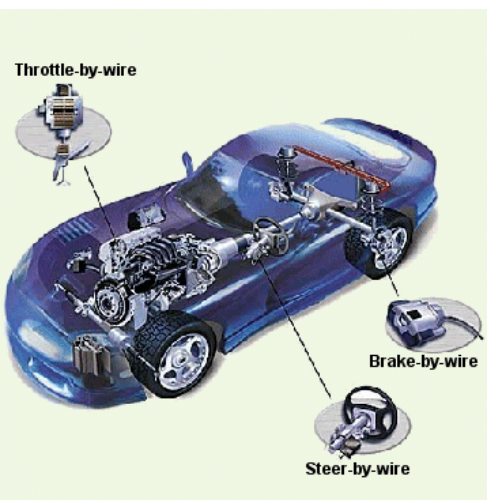
Throughput

Timeliness

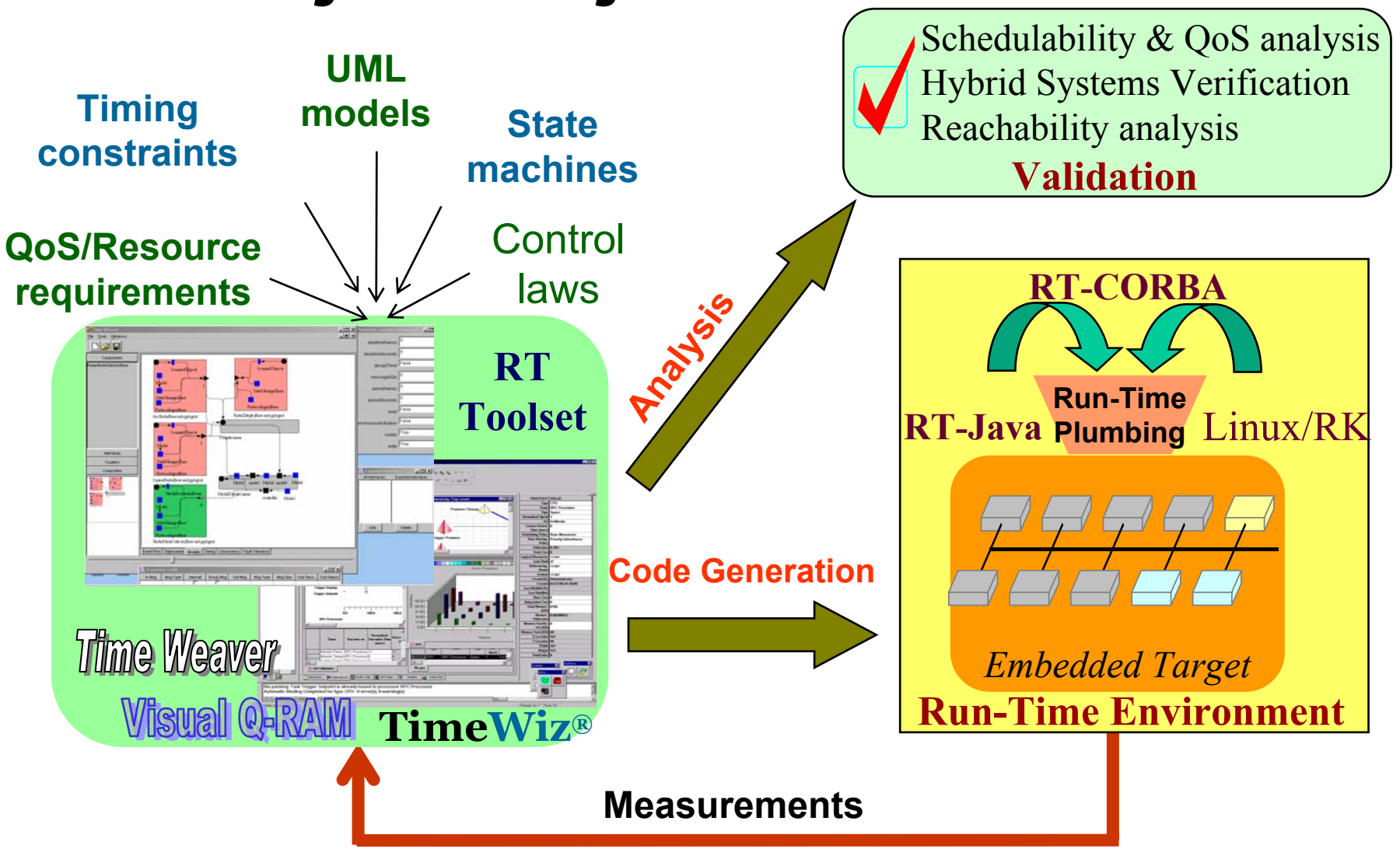
Para-functional



Guarantee End-to-end response time



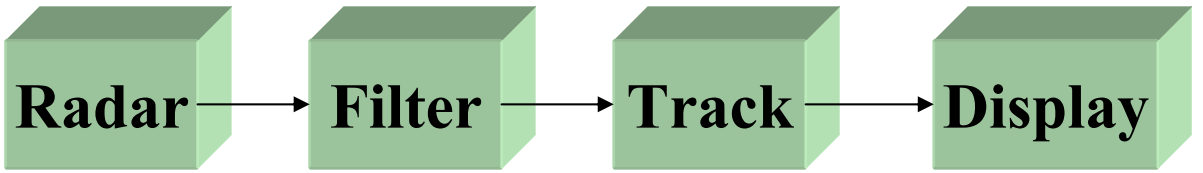
Project Objective



Multi-View Modeling + Analyses + Targeting Freedom +

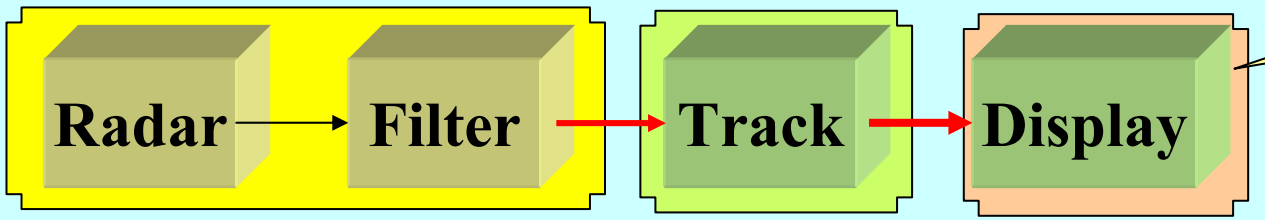
***Reusable* Embedded Software Components**

Capability Summary



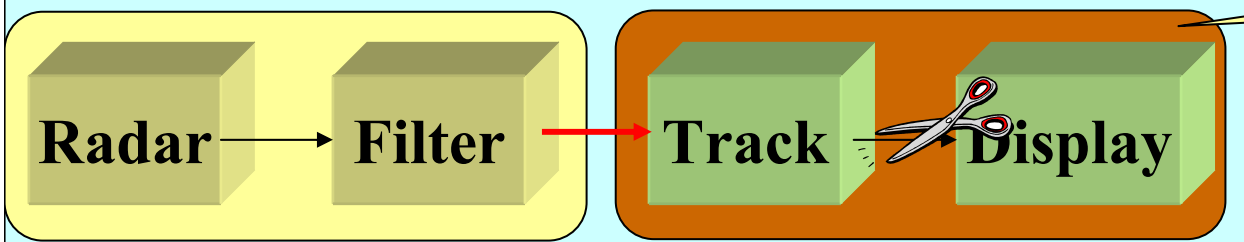
"Functional Coding Only"

Time Weaver



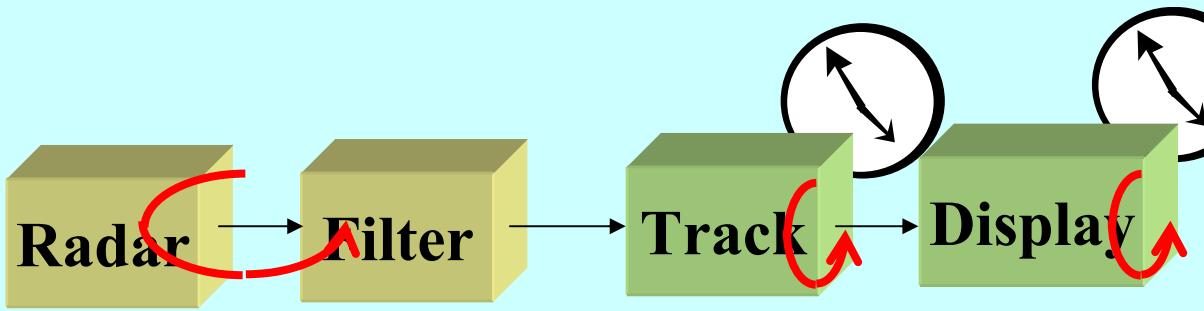
Process allocation

Add/delete
Inter-Process Comm



Processor allocation

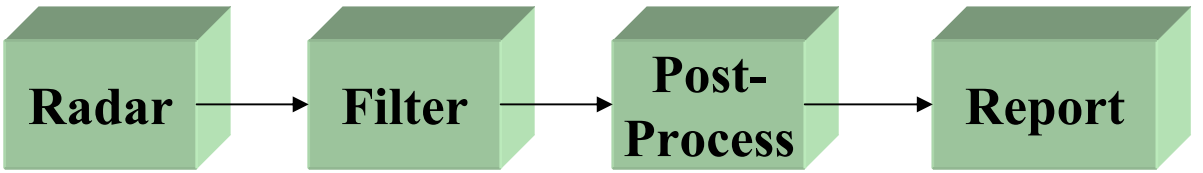
Add/remove
Inter-Processor Comm



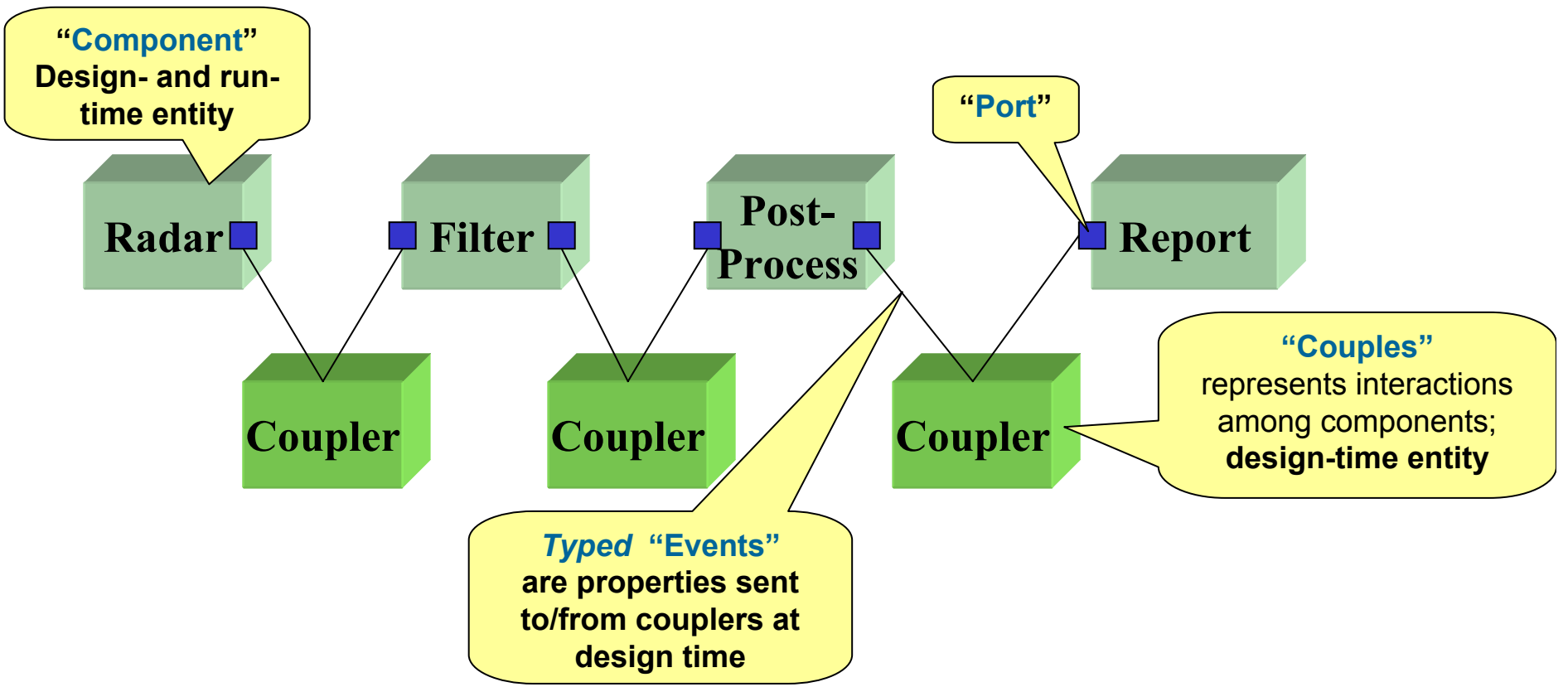
Add/remove threads,
modify timing params

Time Weaver Abstractions

Logical View



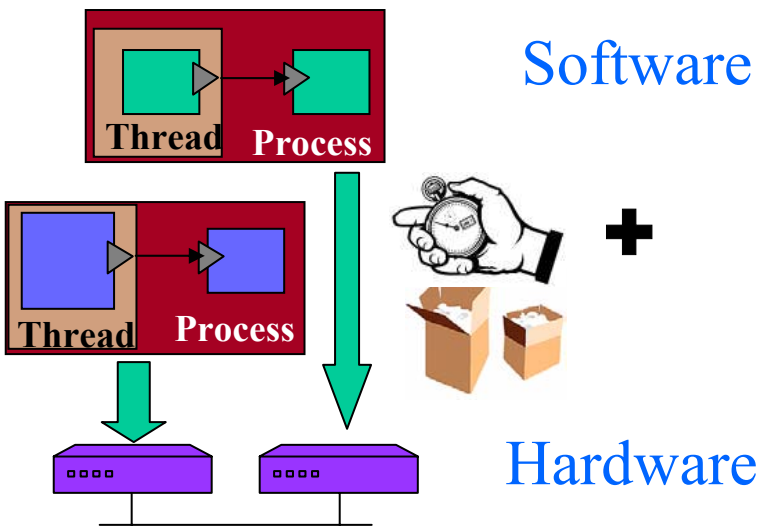
First-Level Time Weaver Representation



Embedded Systems Composition

Embedded System

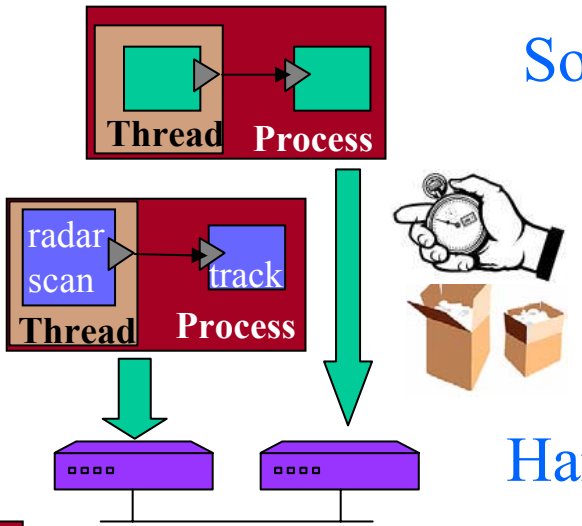
=



Embedded Software

Different Para-Functional Properties

=

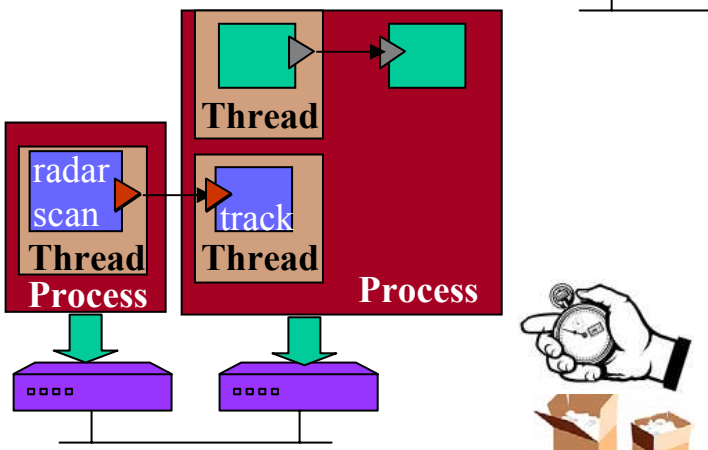


Software

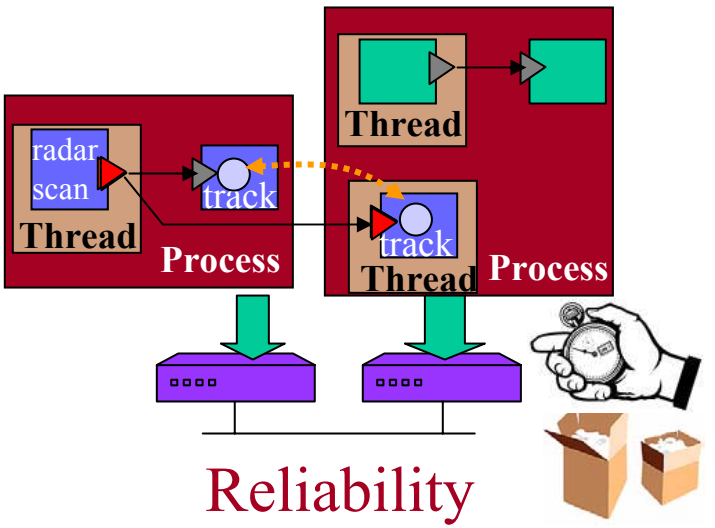
+

Embedded System

Hardware



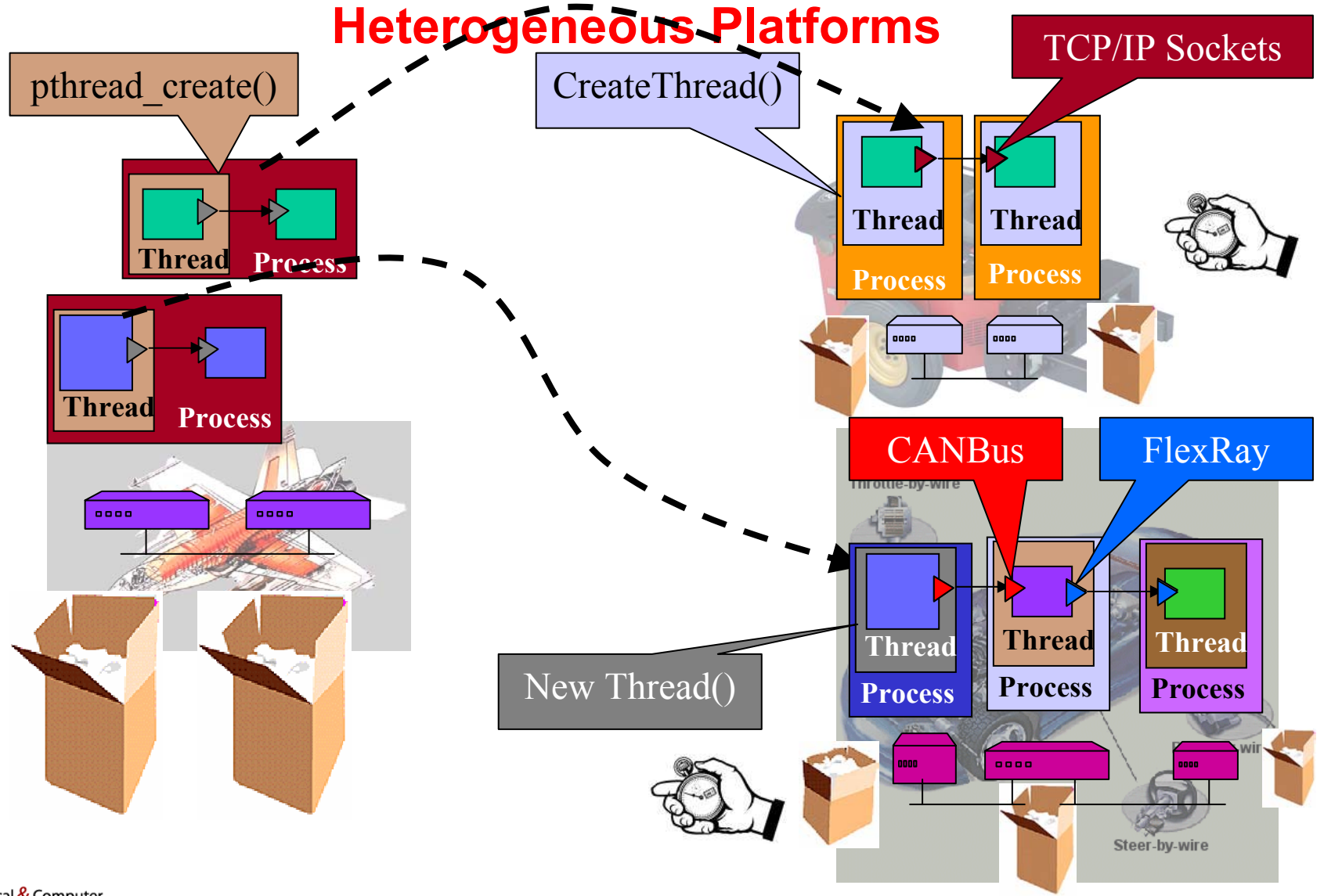
Throughput (trajectories / s)



Reliability

Embedded Software

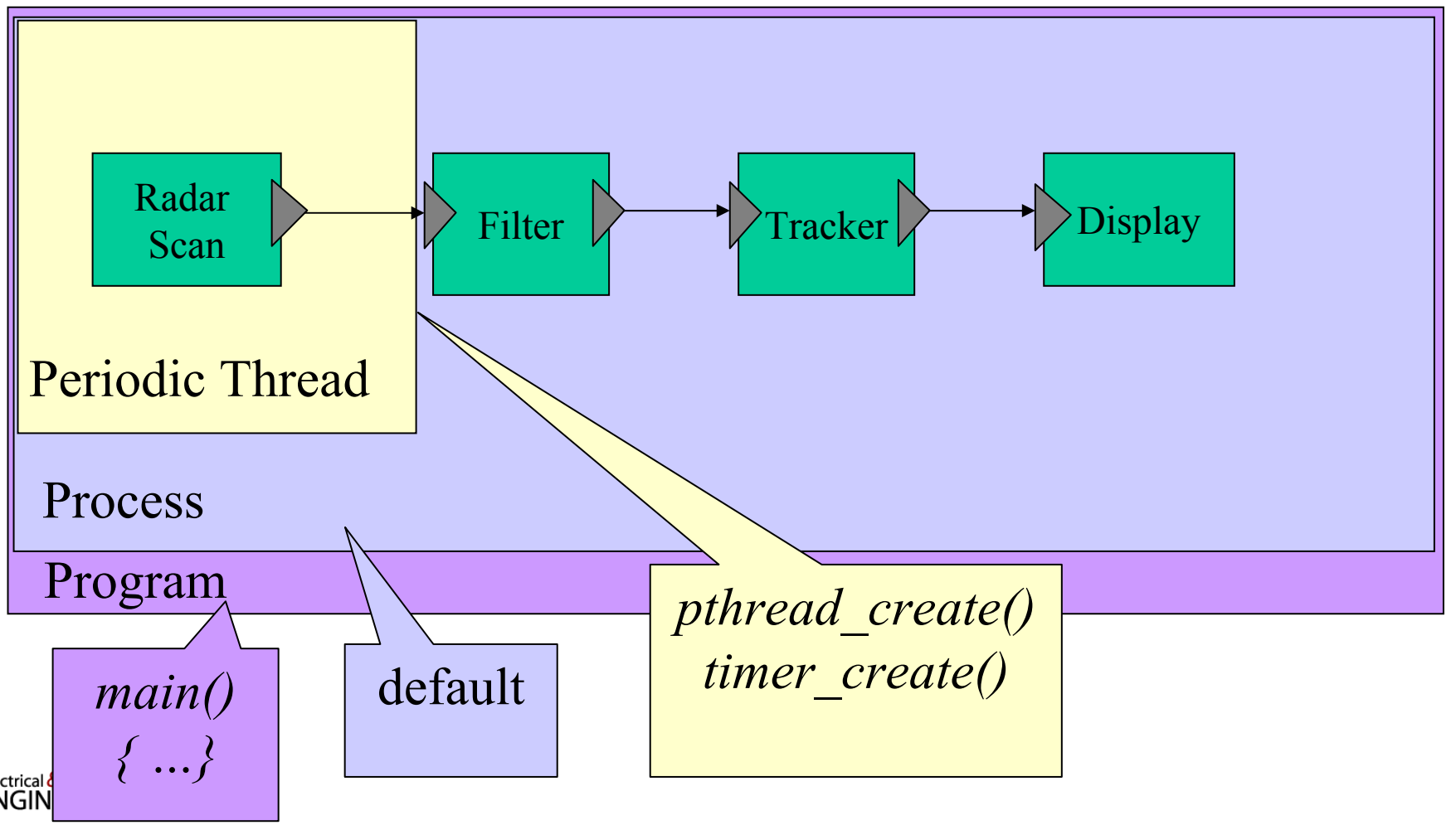
Heterogeneous Platforms



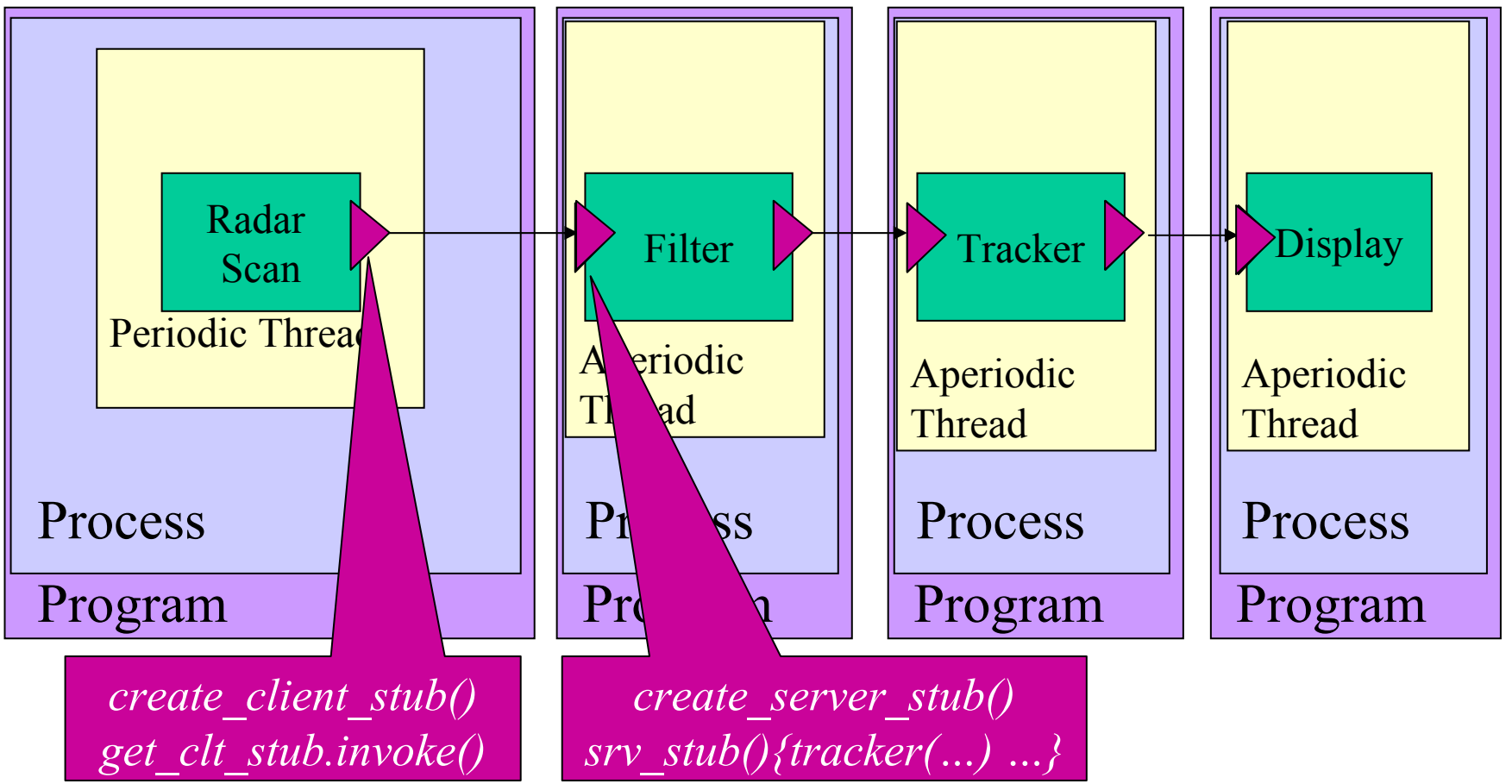
Recent Focus

- Software Reuse Through Software Components
- Complexity Reduction
- Automatic Verification of Para-Functional Properties
- Synthesis

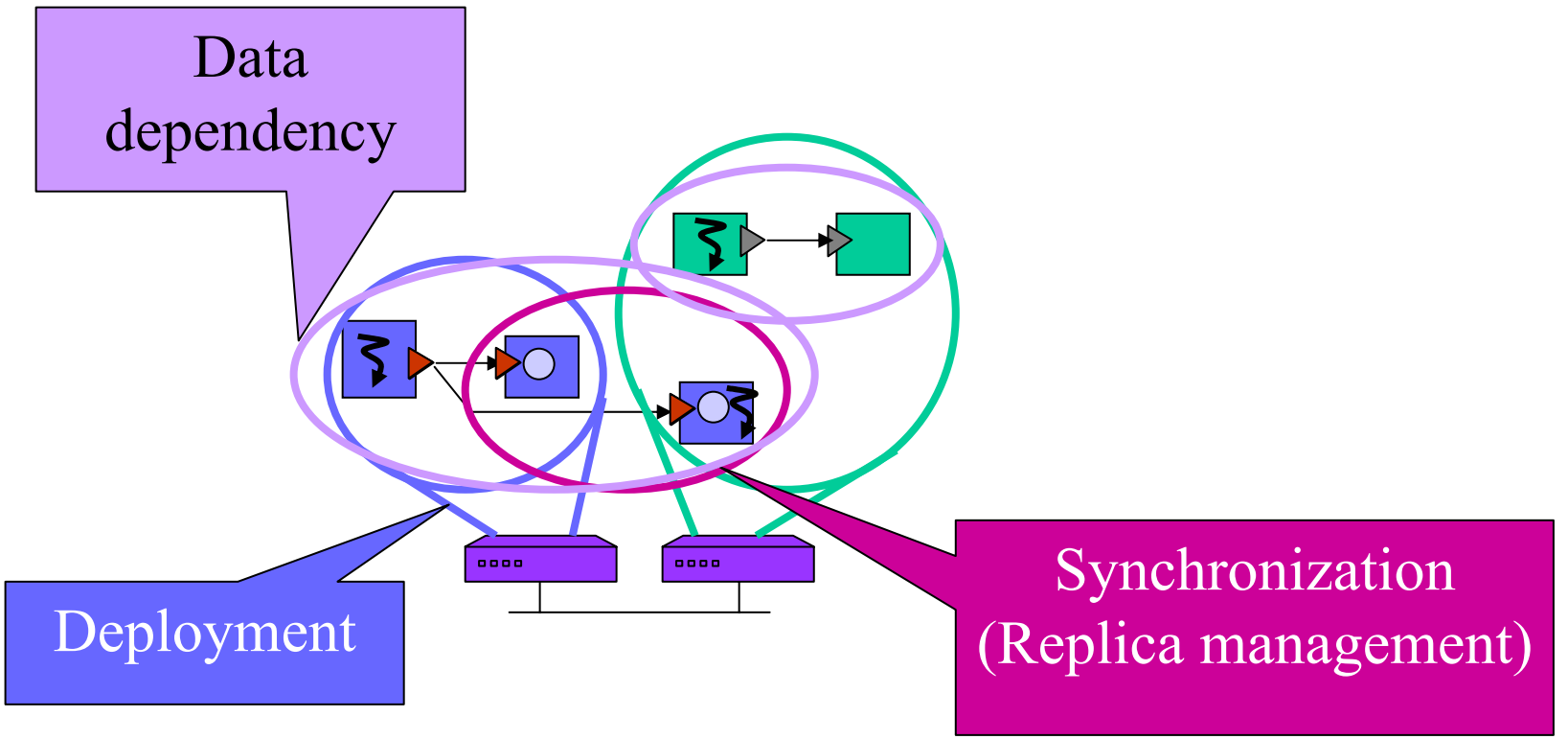
Traditional Process Radar



Traditional Process Pipeline Radar

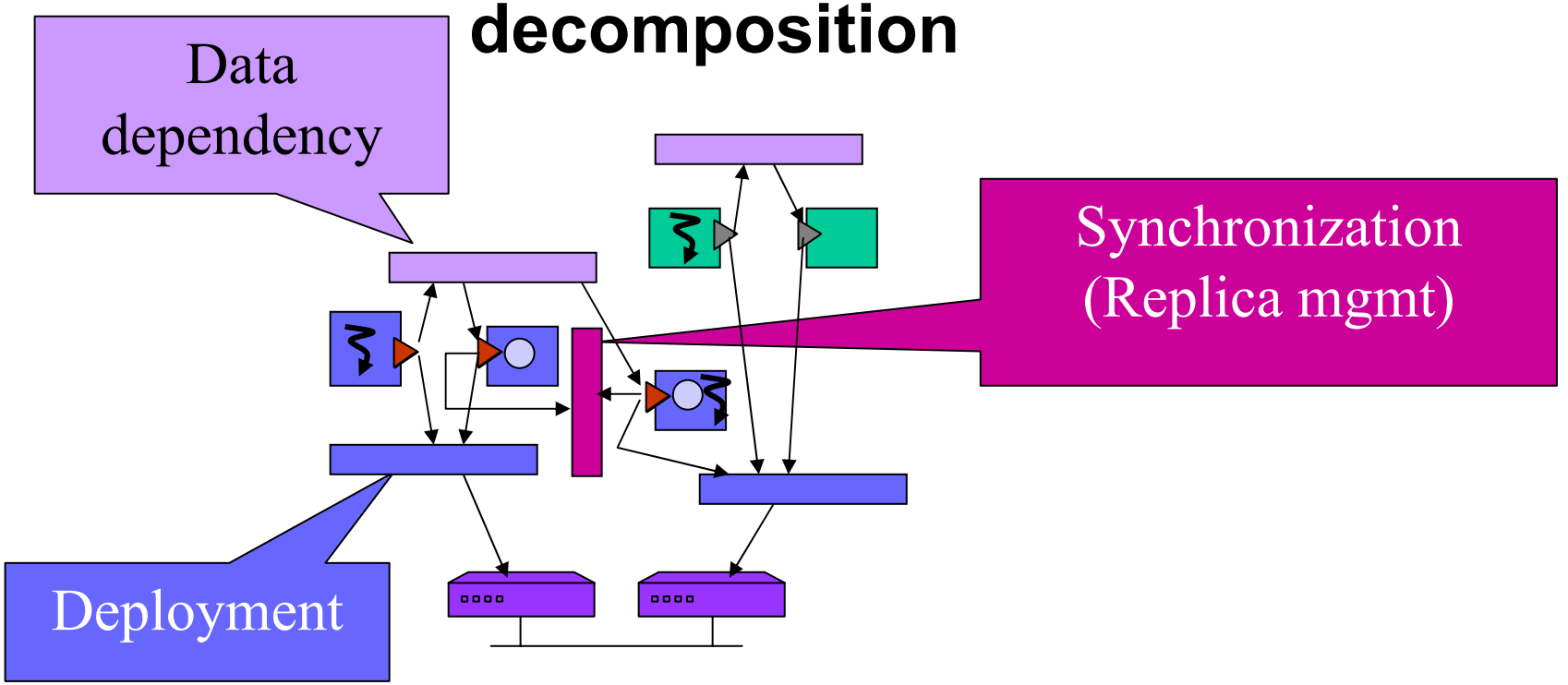


Inter-component relationships decomposition



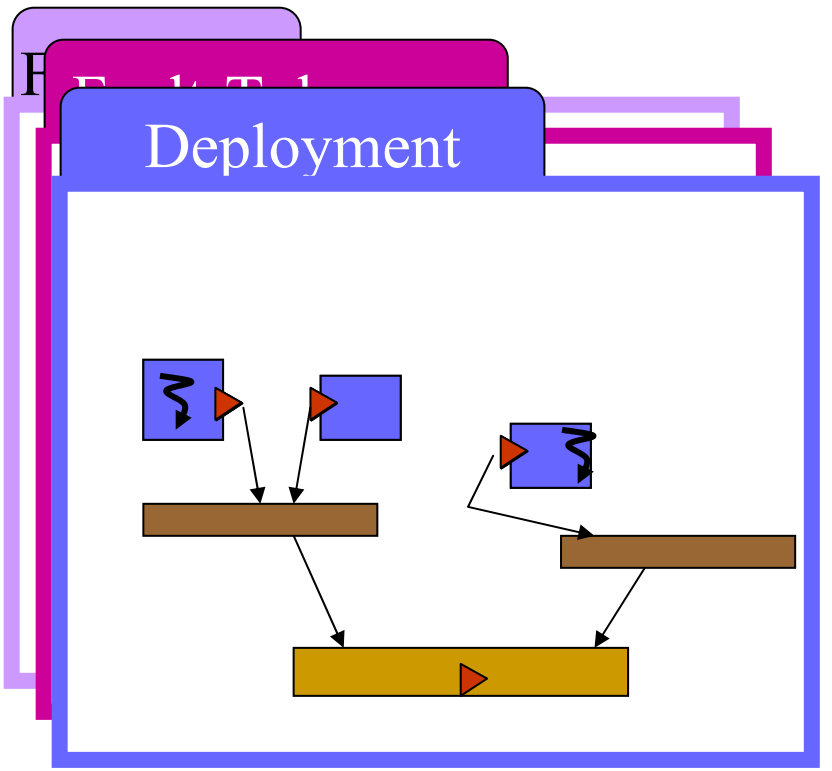
Coupler

Inter-component relationships decomposition



Coupler manipulate port elements to construct relationship

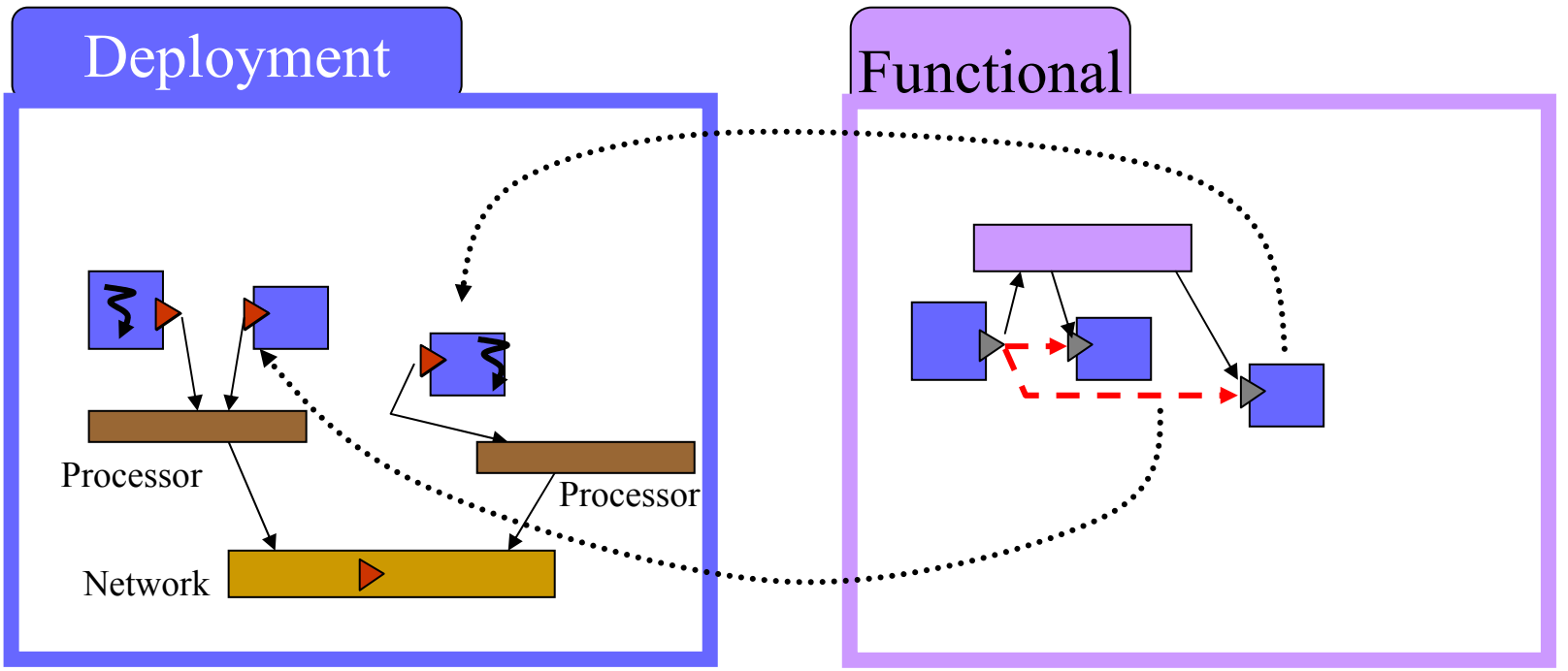
Independent Semantic Dimensions



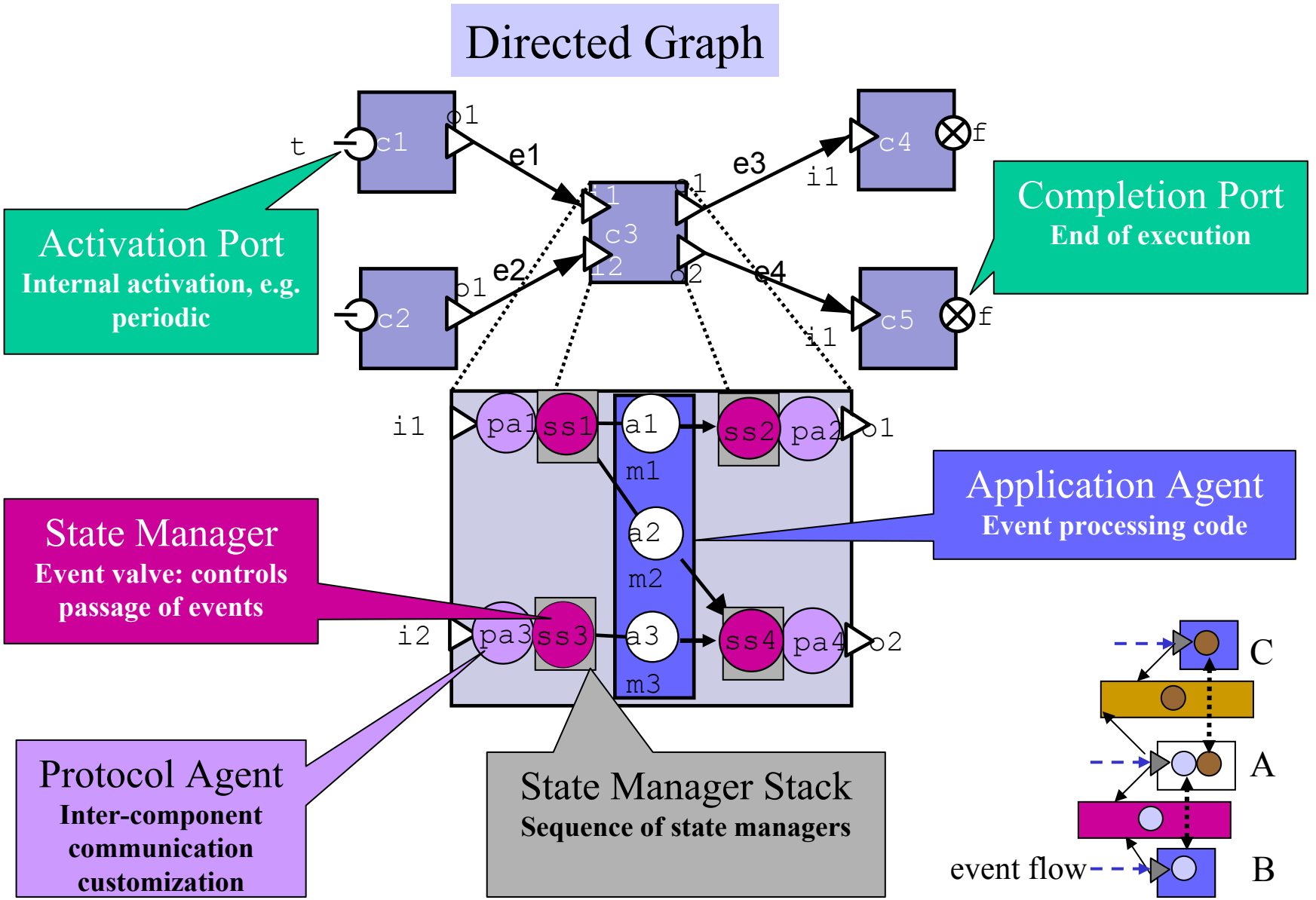
Semantic separation

Projections to other dimensions

Inter-dimensional projections

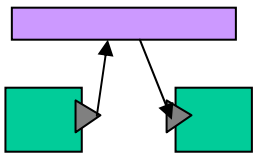


Software Model

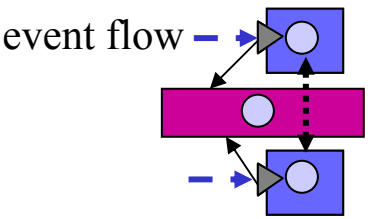


Basic Coupling

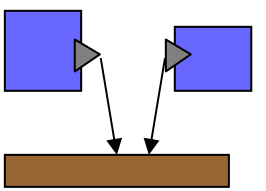
- Event Communication
  - Protocol agents interconnected



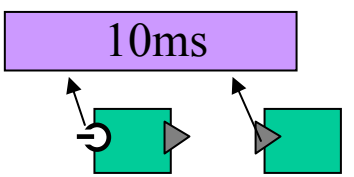
- Synchronization
  - State Managers = event valves
    - Open/close based on synch protocol



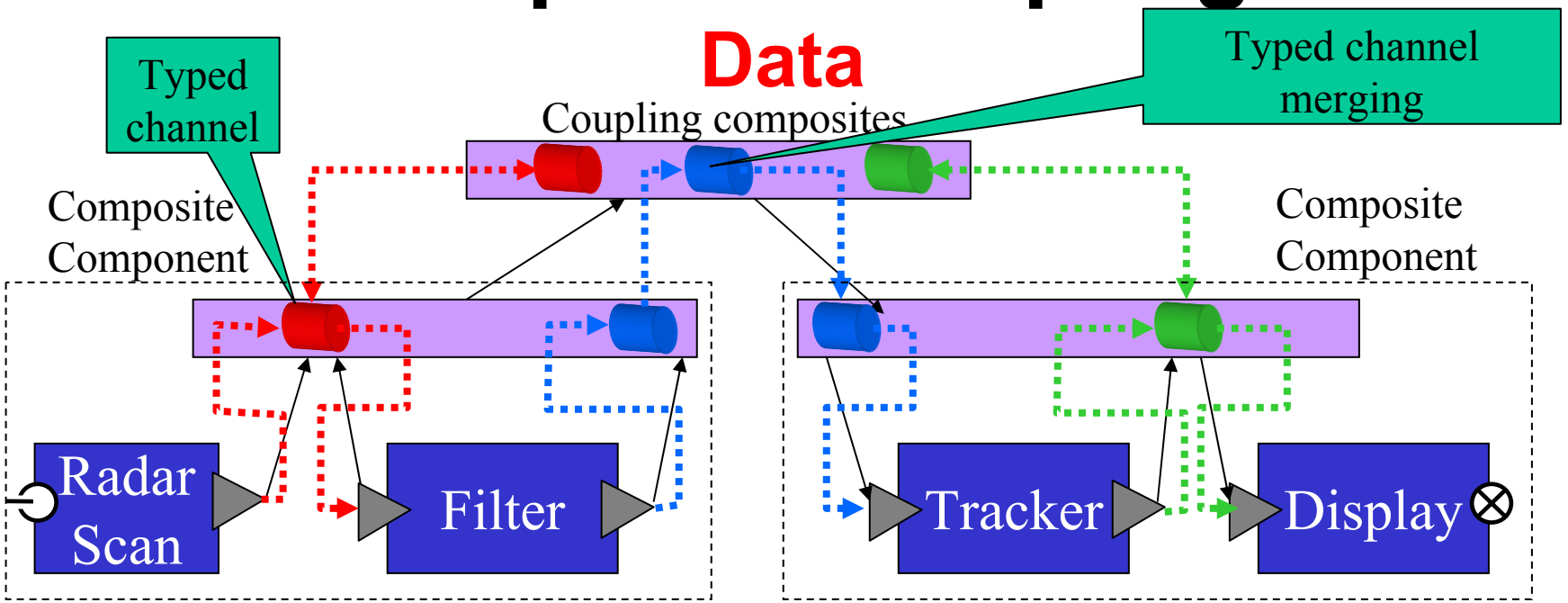
- Aggregation
  - Deploy together



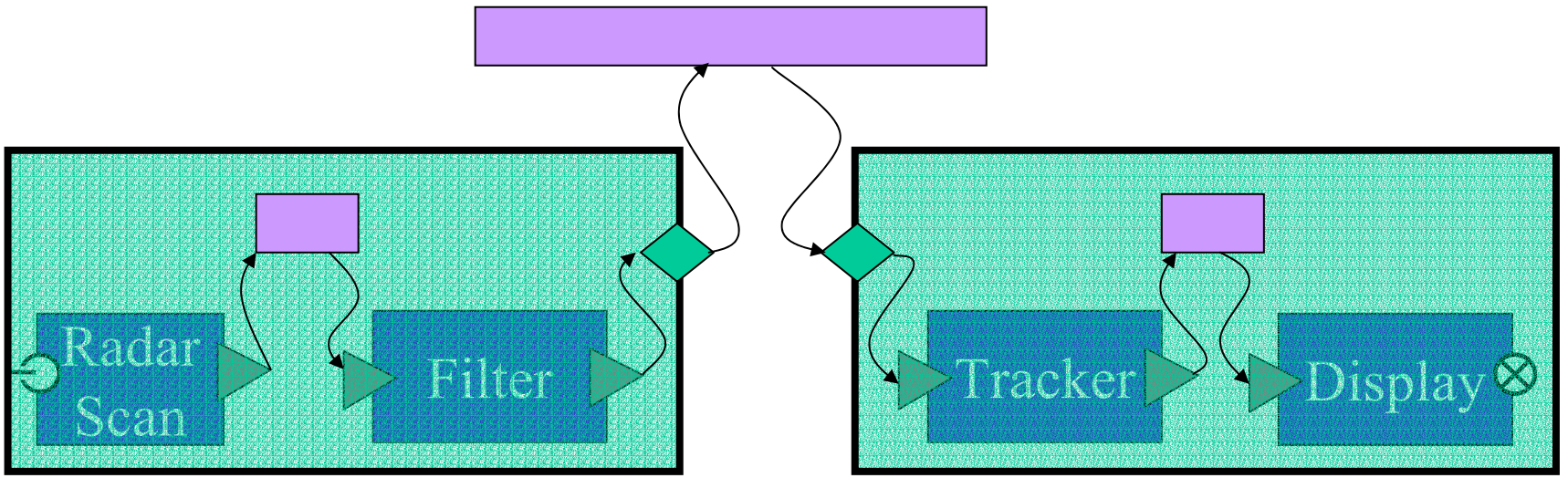
- Related properties
  - Periodic execution



Composite Coupling

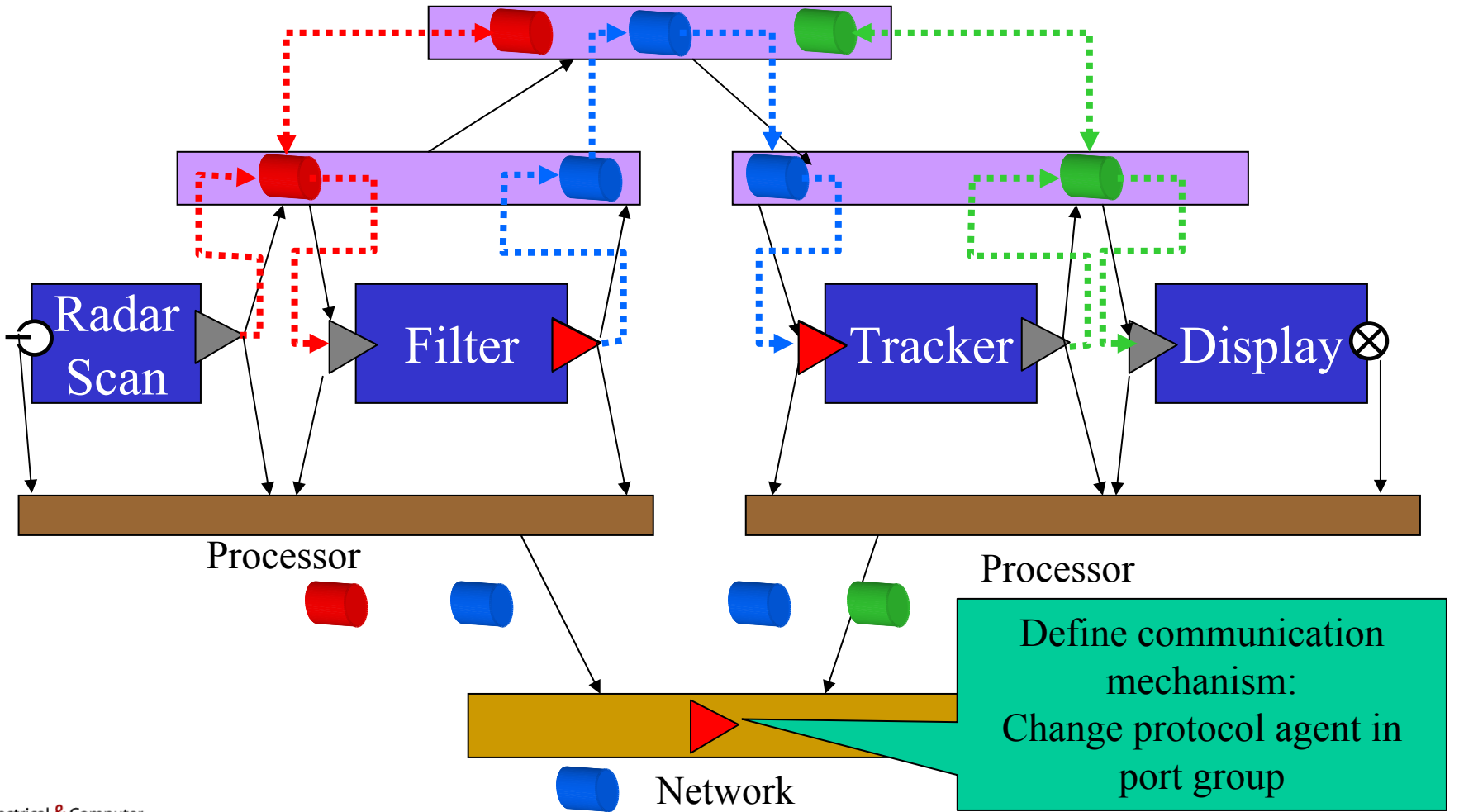


Composite Coupling
Encapsulation



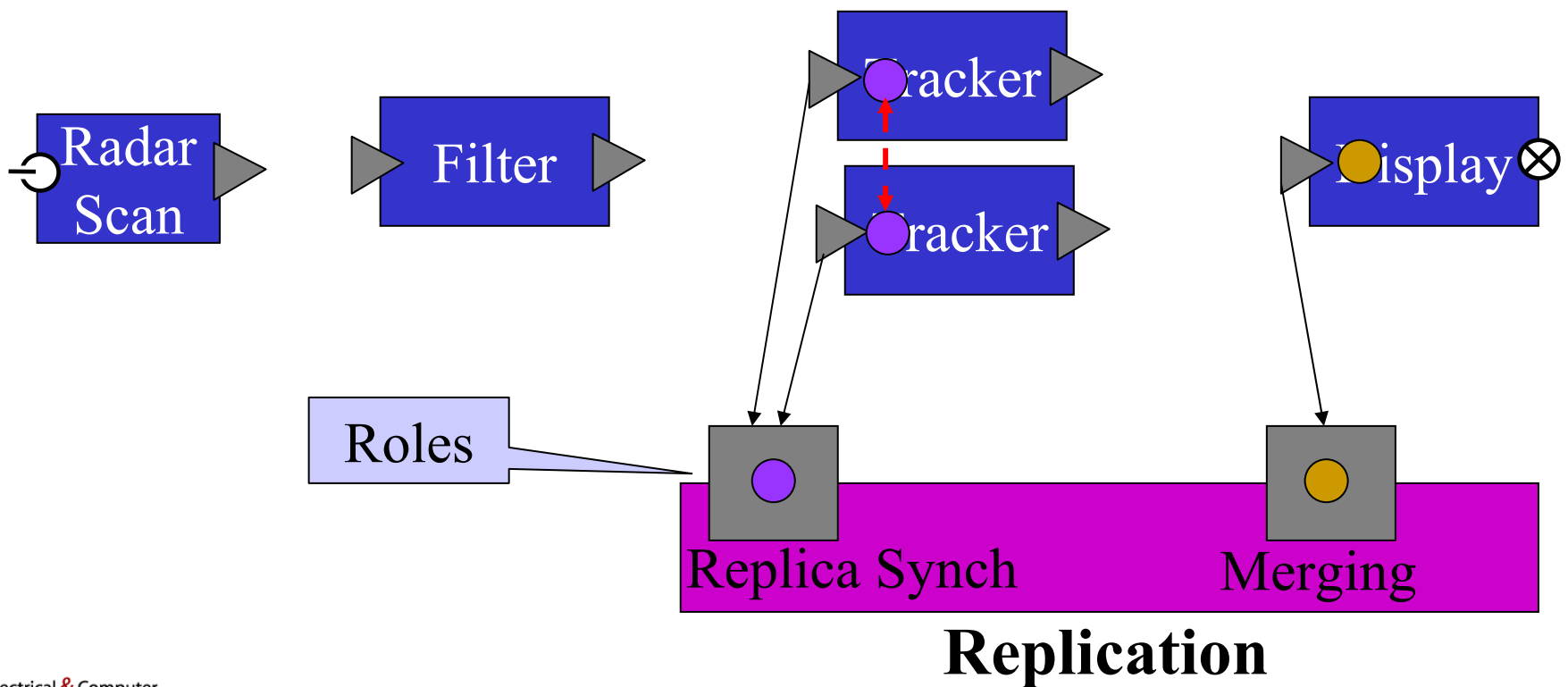
Composite Coupling

Deployment



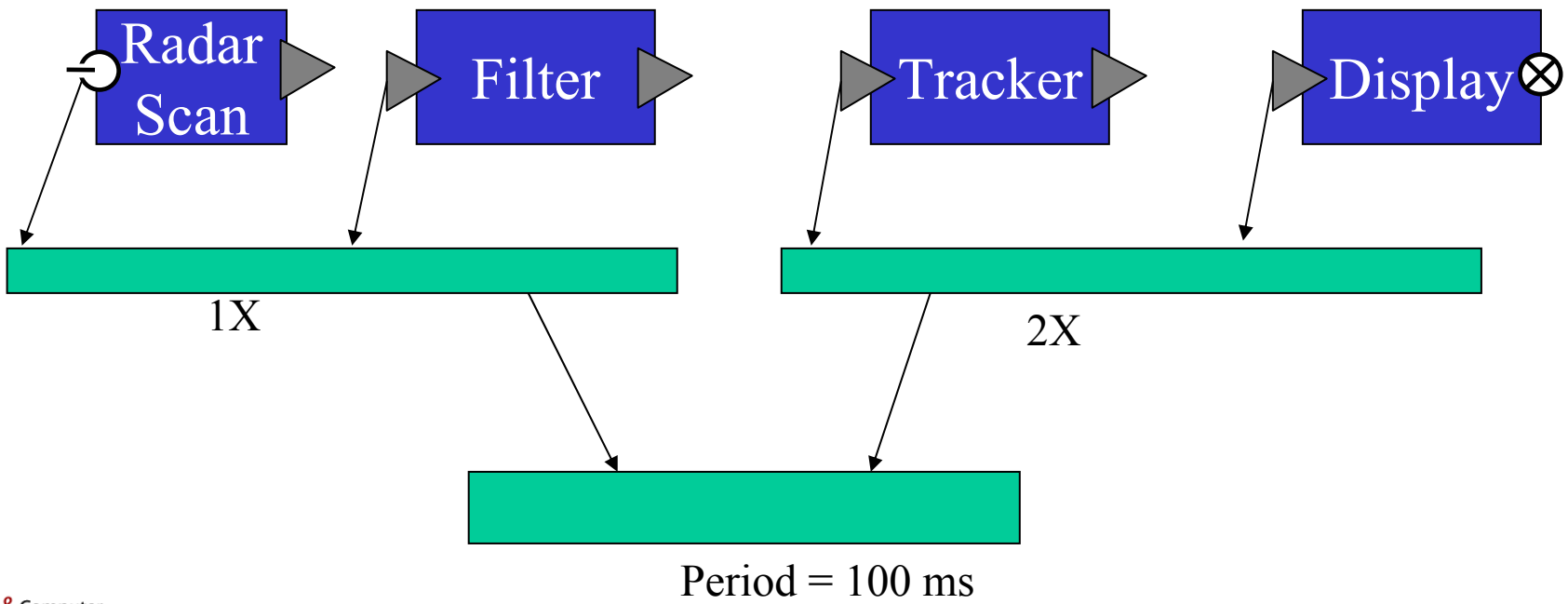
Composite Coupling

Synchronization



Composite Coupling

Timing Properties

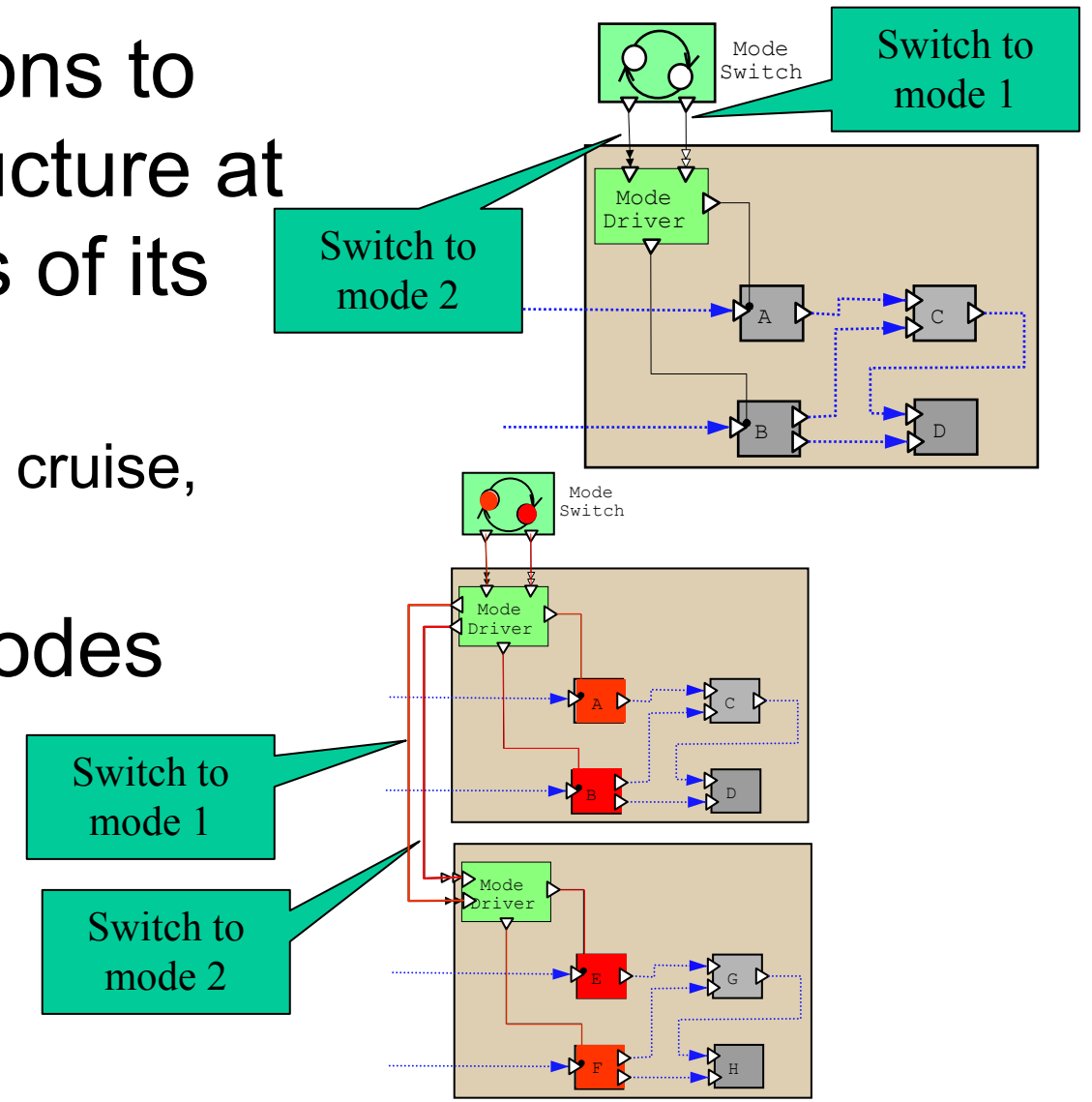


Embedded System Dimensions

- Functional
  - Event flow
- Timing
- Deployment
- Fault-Tolerance
- Modality
- Concurrency

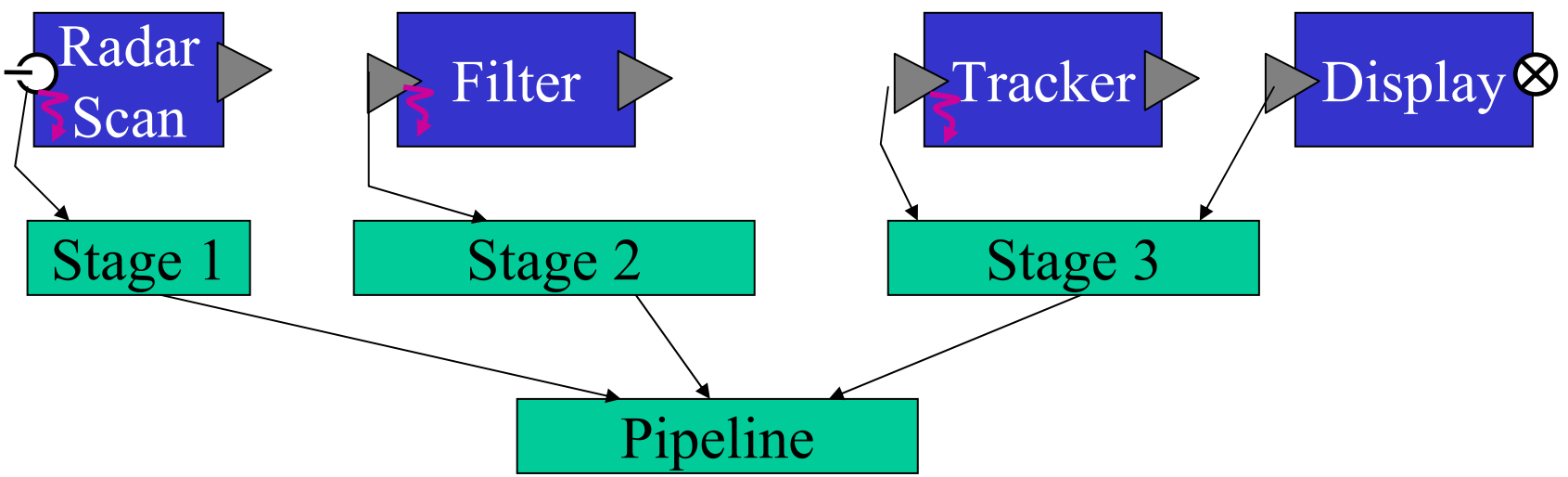
Modality

- Defines variations to the system structure at different stages of its lifetime
  - Avionics: takeoff, cruise, landing
- Hierarchy of modes



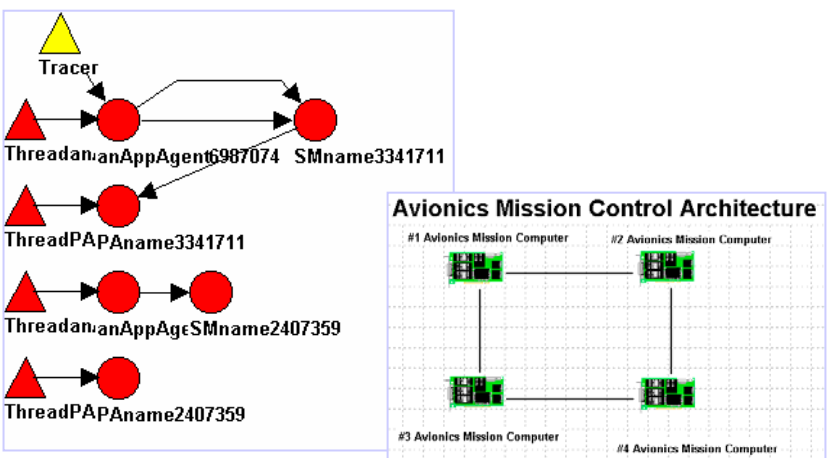
Concurrency

- Structures for concurrent execution
  - Pipeline
- Synchronization of concurrent activities



Automatic Software Model for Verification

- Generation of response chain and hardware model
- Integration with real-time analysis tools for automatic analysis (TimeWiz®)



Code Generation

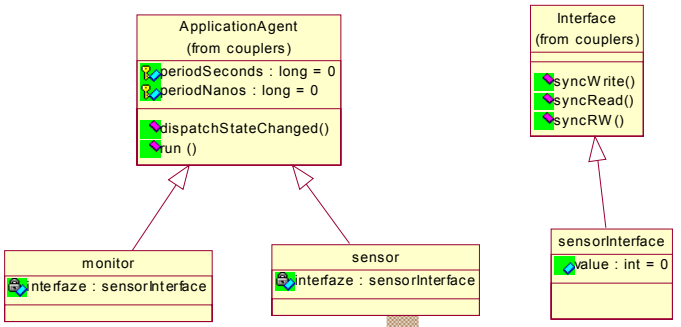
- Components can have multiple implementations
  - C, C++, Real-Time Java, Matlab, XML, …
- Target Multiple Platforms
  - RT-CORBA, RMI, …

Optimization

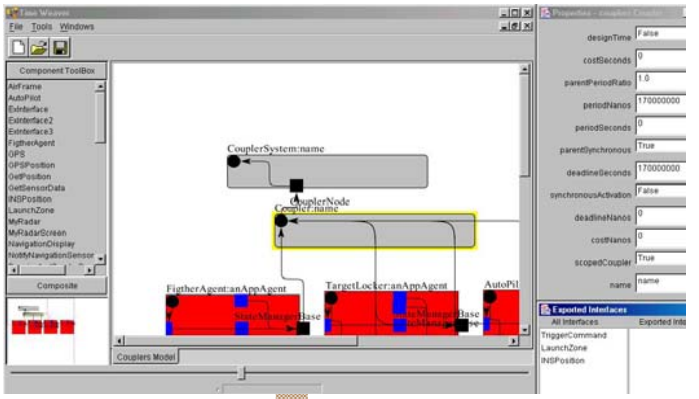
- Automatic selection of optimal communication mechanism
- Synthesis
  - Deployment
  - Hardware Sizing

Tool Integration

Rational® Rose UML

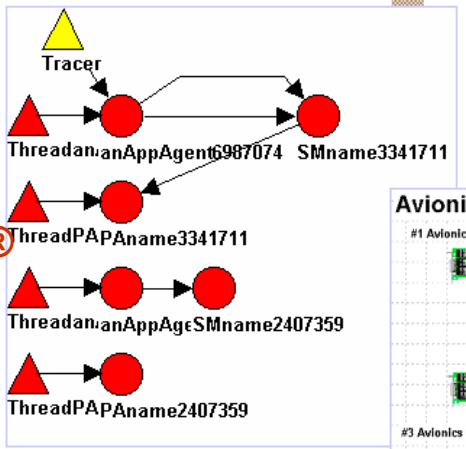


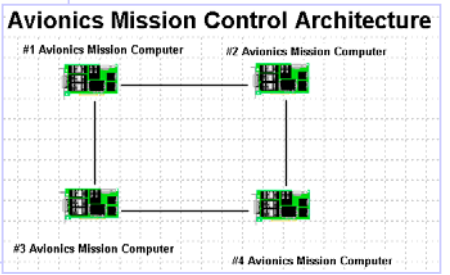
Time Weaver



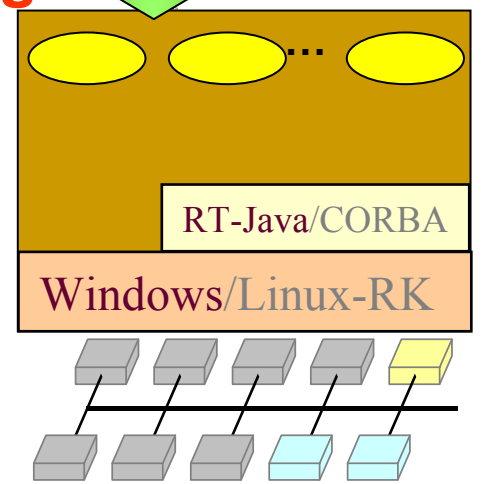
XML Repository

TimeWiz®



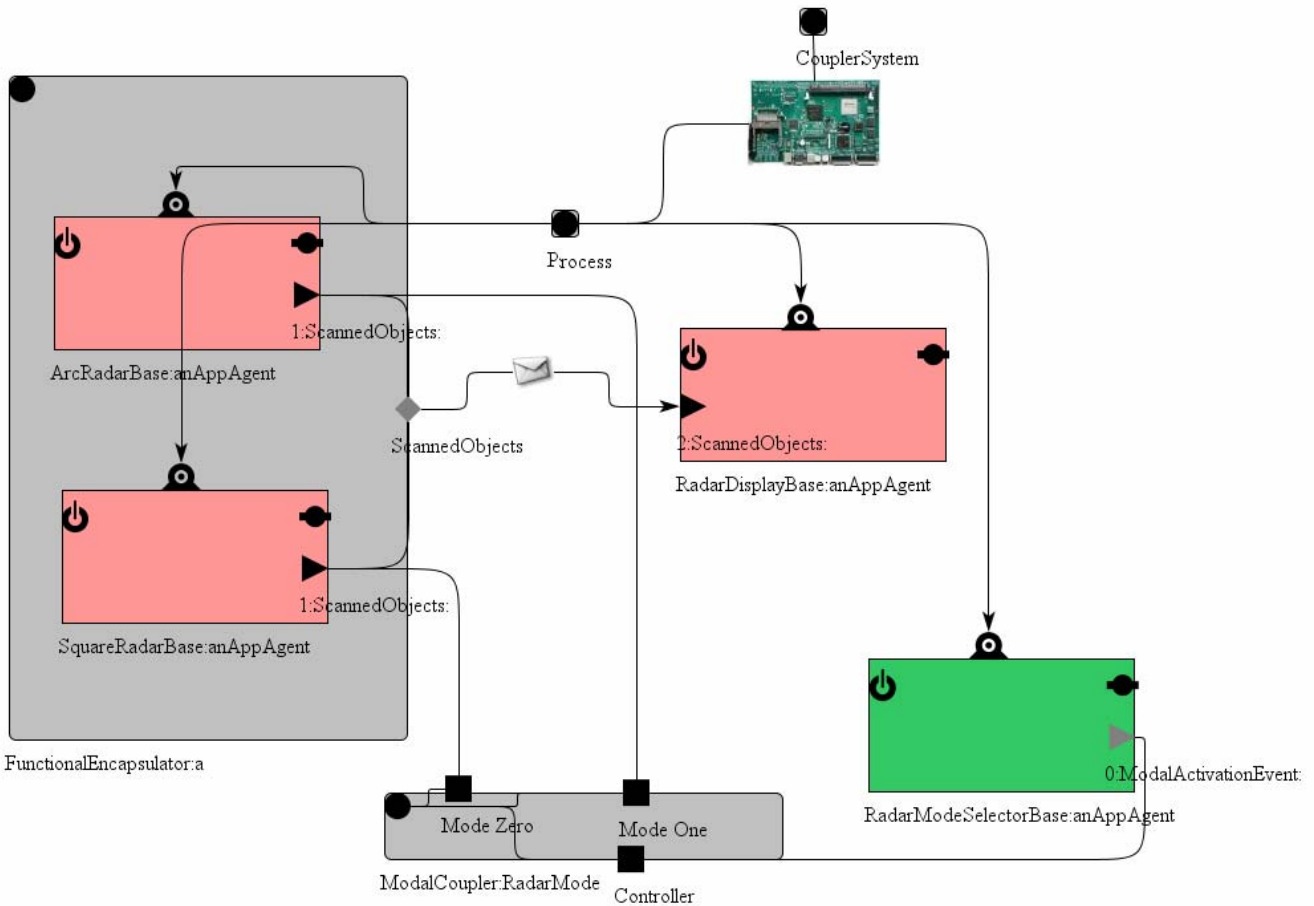


Target



Example

Event Flow Modality Timing Deployment



Case Studies

- Avionics Application Modeling
  - (~400 components)
- Software Radio Modeling
  - Data Flow
  - Throughput increase
- Automotive Testbed
  - Code generation for ARM 7 + μCOS-II
  - Working on: MPC555 + OSEKWorks

Related Work

- Software Architecture
- Meta-H
  - Component-based, Timing Analysis
  - No separate views
- AIRES
  - Hierarchical SM, control driver + functions
  - Timing Analysis
- Ptolemy II
  - Hierarchy of model of computation
  - Simulation
- AOP

Concluding Remarks

- Para-functional requirements – first-class entities
- Couplers – para-functional decomposition abstraction
- Multiple separate dimensions
- Automatic verification
- Code generation
- Tool integration