Modeling the Implementation of Stated-Based System Architectures

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Peter H Feiler
June 2009

State-based Systems Are Everywhere

What is a state-based system

- State: discrete, continuous, large, small
- State transition: change, delta, command, event
- Transition conditions & actions

Types of systems

- Control systems
- Autonomous systems
- Communication systems
- Resource management systems

What do they do

- Communication of state
- Coordination of state

Voluminous State Systems

State of physical environment

- Example: Tracking of object close to space station

Communication of state

- Series of state transmissions vs. sequence of change transmissions
- Data stream perspective
  - State: High data volume, incomplete stream ok => tolerant to transient transmission failures
  - State change: low volume, complete stream critical => requires guaranteed delivery

AADL Modeling

- Sampling of data ports for state vs. queuing event data ports for state change
- Data stream & protocol QOS properties
- Deployment to hardware

Fail-safe operation
by mixing state & deltas

Embedded Control Systems

Observe and affect state of physical systems

Continuous time state

- Time sensitive data
- Setpoints in absolute vs. relative terms (state vs. delta)
- Periodic sampling of state
- Up/down sampling of data stream across harmonic tasks
- Ordering of send & receive, write/read patterns => frame-level jitter in data stream
- Missed sample => aged data

Shared variables vs. port-based flow architecture

AADL Modeling

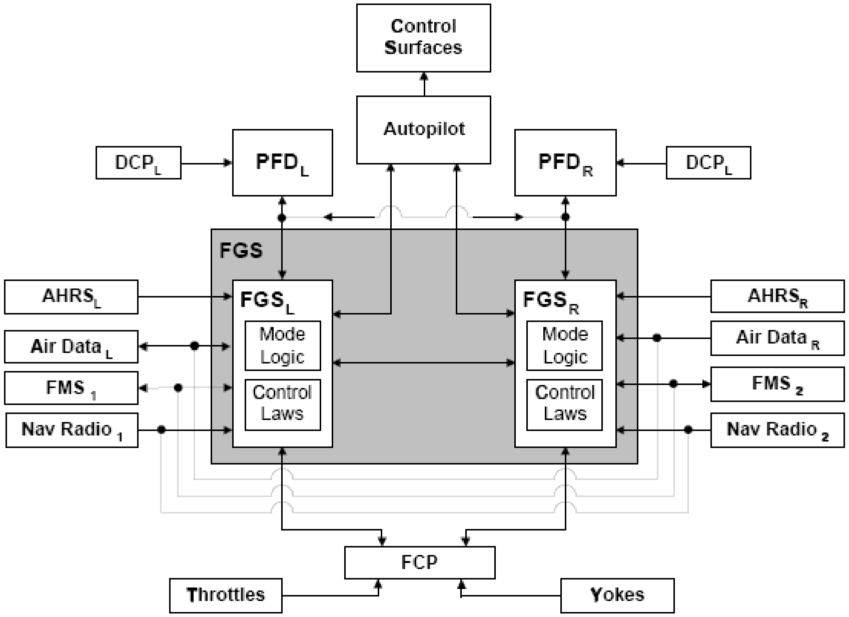
- Data ports & periodic threads
- Devices as sensors/actuators
- Input-Compute-Output model (data consistency)
- Deterministic sampling patterns (immediate, delayed)
- End-to-end flows

Time sensitivity of state impacted by scheduling & sampling communication

Embedded Discrete State Systems

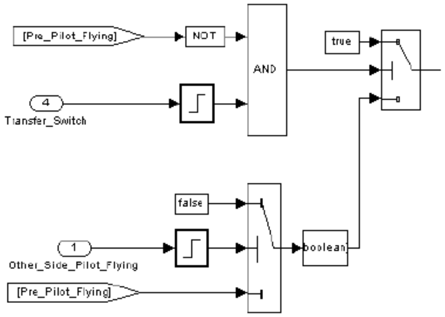
Examples

- Hybrid control systems
- Systems with operational modes
- Discrete state observations in periodic systems



Left leader
Right leader
Dual operation

Sampled Processing of Discrete State Systems



Coordinated state transitions

- Hand shaking protocols
- Replicated distributed state machine

Discrete states in control system

- Predictability of periodic task loads
- Sampled observation of events & binary states due to truth tables & Simulink
- Non-deterministic sampling leads to missed event/state change observations

Mirrored state machines

- Watch for external transition events vs. successful state change of “fraternal twin” (fail-safe)

AADL Modeling

- Events vs. sampling of states
- Modes & synchronized mode transitions
- Failure propagation modeling

Issues with event observation by sampling

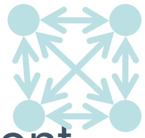
Adaptive Systems

Fail-safe operation by periodic sampling of target state

Workloads & service levels

- Supervisor
- Observes workload (global system state)
- Controls subsystem service level (assignment of resources)

Service levels as state machines



- Fully connected state machine (goto service level X)
- Linear progression through service level (Increment/decrement request)

Communication of service request

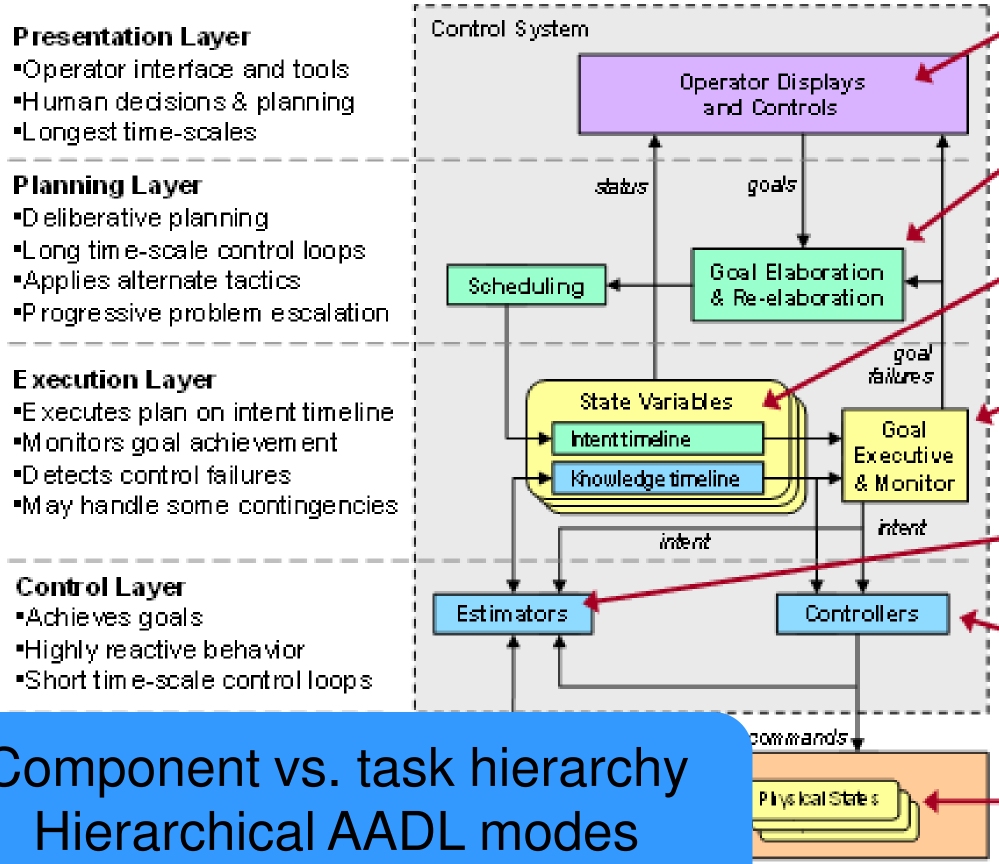
- State change requests: sampled commands => repeated action
- Target state: repeated transfer ensures fail-safe sampling
- Coordinated state transition => transient transition period

AADL Modeling

- Modes & transitions
- State as shared variables vs. communication through data ports
- Deployment, resource capacities & budgets

Autonomous Systems

Multi-layered interacting state machines



Presentation Layer
- Operator interface and tools
- Human decisions & planning
- Longest time-scales

Planning Layer
- Deliberative planning
- Long time-scale control loops
- Applies alternate tactics
- Progressive problem escalation

Execution Layer
- Executes plan on intent timeline
- Monitors goal achievement
- Detects control failures
- May handle some contingencies

Control Layer
- Achieves goals
- Highly reactive behavior
- Short time-scale control loops

Goal monitoring for early transition failure detection

Goal networks drive controller target states

Operational commands as controller modes

State variable based design of flow-based system

Time sensitive control loops

Discrete state & event observations

Component vs. task hierarchy
Hierarchical AADL modes
Reusable reference architecture

Summary

What matters about the state behavior

- Large vs. small state
- Continuous time vs. discrete state
- State vs. state change
- Absolute vs. relative reference points
- Target state vs. action steps of transition path
- Identical vs. mirrored distribution of state machine

What matters about implementation

- Sampling vs. queued events & message
- Determinism of sampling
- Guaranteed & ordered delivery
- Ports & shared data
- Fail-safe replication, distribution, mirroring

Peter Feiler

phf@sei.cmu.edu

NO WARRANTY