

Hybrid Workgroup

Pam Binns
Magnus Carlsson
David Corman
Bonnie Heck
Tom Henzinger
Gokhan Inalhan
Gabor Karsai, co-chair
Wallace Kelly
Edward A. Lee, chair
Brian Mendel
Andrew Moran
Linda Wills

Modeling Questions

- Language (syntax, semantics, composition)
- Modeling a distributed physical plant
- Modeling the controller
- Modeling faults and their effects
- How are models used?
- Semantics of anytime computation

Uses of Hybrid Systems

- Analysis
 - Analysis of discrete and continuous controllers interacting with a continuous plant
- Simulation
 - Evaluation of continuous models with discrete mode transitions
- Embedded modeling
 - Embedding of continuous models in deployed systems, integrated into the control algorithms (e.g. for fault identification)

Questions

What's in the OCP, and what's in the application?

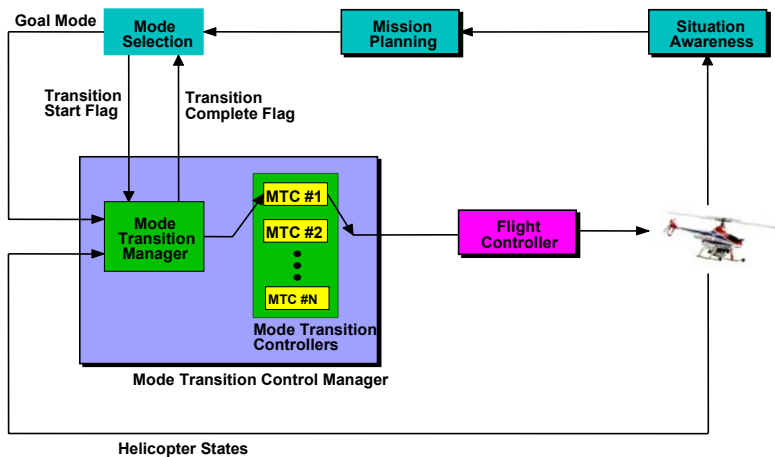
- Run-time support for hybrid models? at what cost?
- Design-time support? (e.g. generation of software)
- Anytime computation? (resource management)
- QoS guarantees? Is real-time O/S the only option?
- Admission control?
- Support for domain-specific languages?

Examples Considered

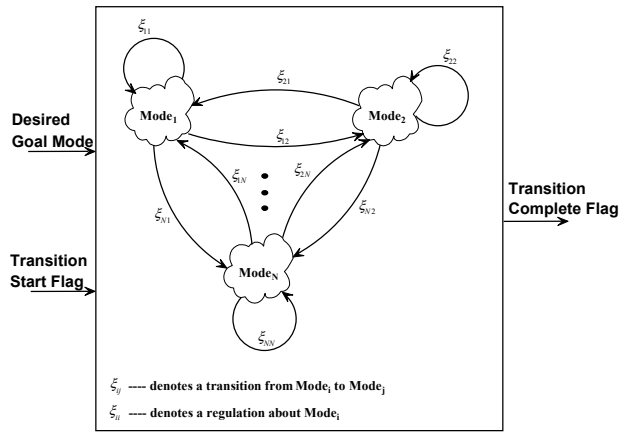
- UAV helicopter control (GATech)
- Free flight conflict avoidance (Rockwell)
- Platooning vehicles (Berkeley)
- Dragonfly – Multi-vehicle coord (Stanford)

In all cases, we talked about fault detection and adaptation using modal control.

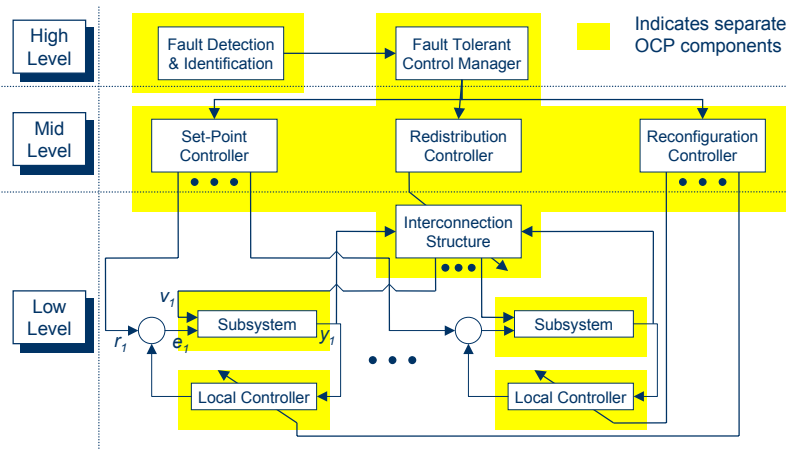
Mode Transition Control (GATech) Overall Architecture



Mode Transition Specification



FDI/FTC Demo in OCP (GATech)

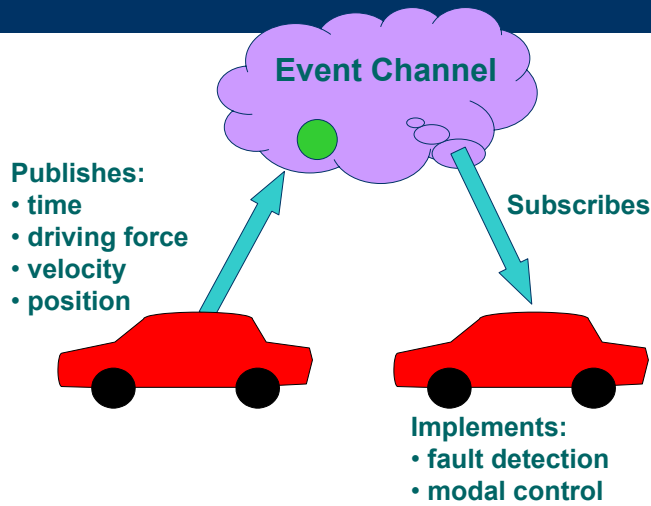


Free Flight (Rockwell)

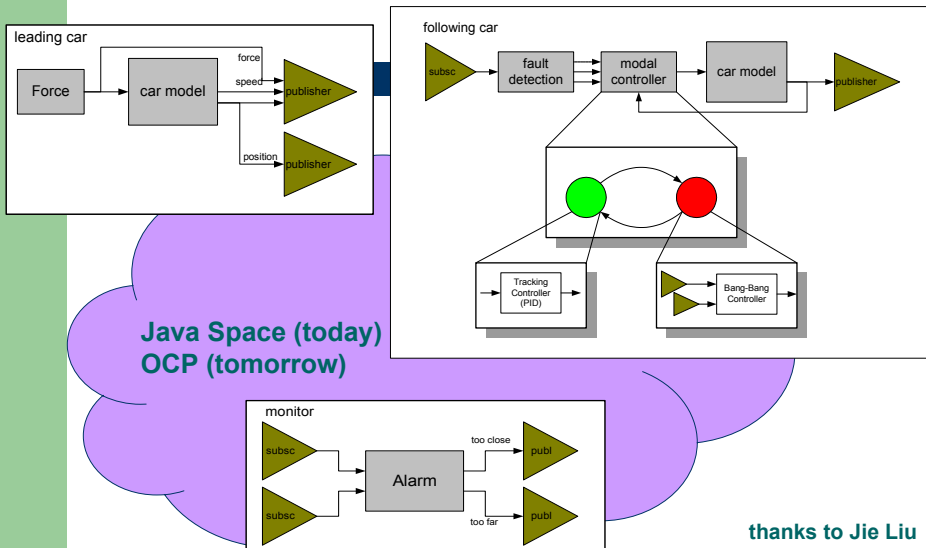
Extensions

- UCAVs with accurate models of one another – coordinated maneuvers rather than assuming the other aircraft will not react.
- Errors in transmission, faulty/damaged UCAVs – fault identification and adaptation.
- Create hybrid automaton model and use hybrid Mocha to verify that conflicts do not occur.

Scenario – Vehicle Tracking

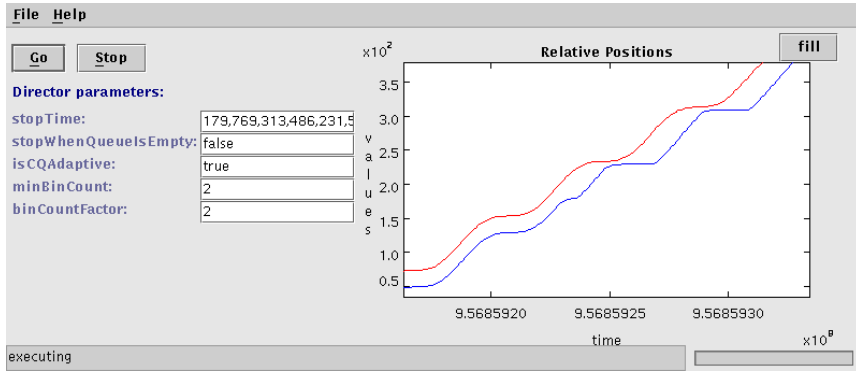


Modeling Car Tracking



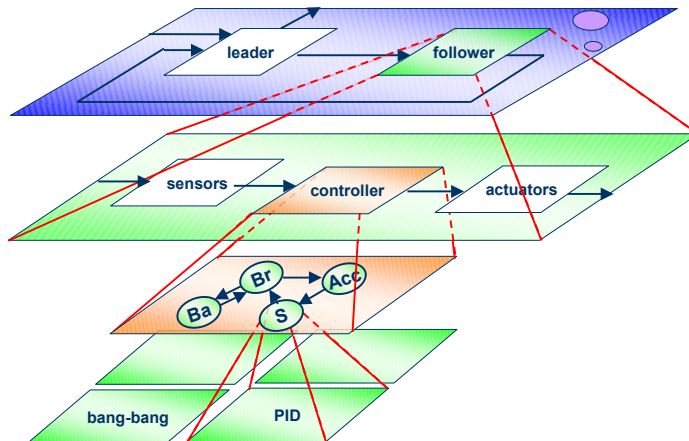
thanks to Jie Liu
and Xiaojun Liu

Execution



Hierarchical View

OCP here only?



Styles of Publish and Subscribe Interactions

- time stamped events?
- globally time stamped?
- reliable delivery?
- ordered delivery?
- signal coordination?
- synchronous delivery?
- blending of multiple publishers?
- dynamic redirection/resourcing?
- persistence?
- history?

OCP Domain #1

- time stamped events? yes
- globally time stamped? no
- reliable delivery? yes
- ordered delivery? no
- signal coordination? yes
- globally synchronous delivery? no
- blending of multiple publishers? no
- dynamic redirection/resourcing? yes
- persistence? no
- history? no

What is a Domain

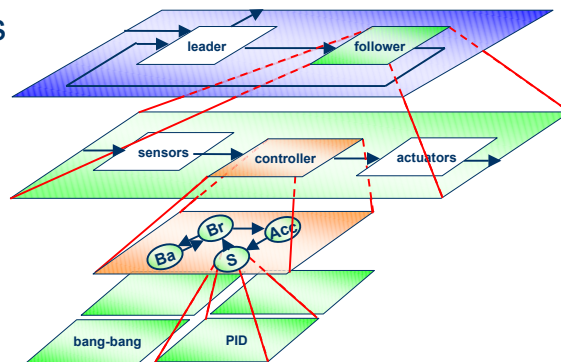
The definition of the interaction of components, and the software that supports this interaction.

Multi-domain modeling means:

- Hierarchical composition
 - heterogeneous models allowed
- Domains can be specialized
 - avoid creeping featurism
 - enable verification
- Data replication in OCP/Boldstroke is another domain
 - separation of communication mechanisms.

What technology can be shared when building domains?

- Abstract syntax
- Type systems
- Components
- Interfaces
- ...



Domains After Domain 1

- High-sample-rate, periodic event handling
- Stream-based component interaction
- Control reconfiguration management
- Time-triggered, synchronous modeling
- Continuous-time domain

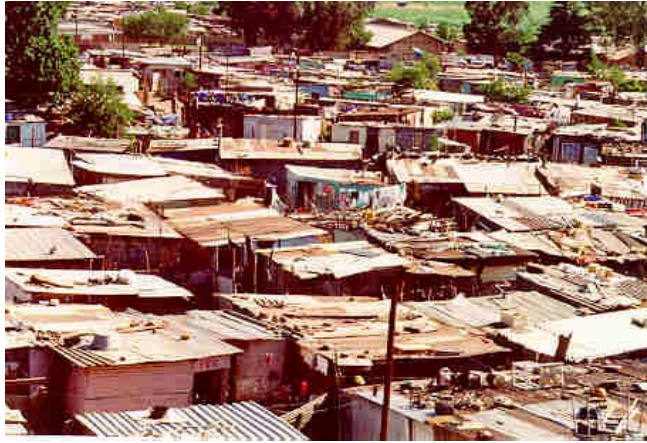
Domains need to be able to:

- Share a consistent notion of time
- Share signals/params across levels of the hierarchy.
- Export interfaces to other domains
- Import components designed in another domain

Milestones

- Use OCP in demonstrations in each of
 - Active state models
 - On-line control customization
 - Coordinated multi-modal control
- Steps
 - Define the challenge problems that justify domains.
 - Work out how mode transitions and other control reconfiguration are handled in the OCP.
 - Work out the interaction semantics (domains) supported by the OCP (mode transition control? continuous modeling? synchronous interactions?)

Architecture to Avoid



Poor common infrastructure.
Weak specialization.
Poor resource management and sharing.
Poor planning.

Also to Avoid



Elegant, unified, and beautiful, but rigid, inflexible, and difficult to adapt. Plus, it takes 100 years to build.



Elegant Federation

Elegant federation of heterogeneous models.



Two Rodeo Drive, Kaplan, McLaughlin, Diaz

Source: *Kaplan McLaughlin Diaz, R. Rappaport, Rockport, 1998*