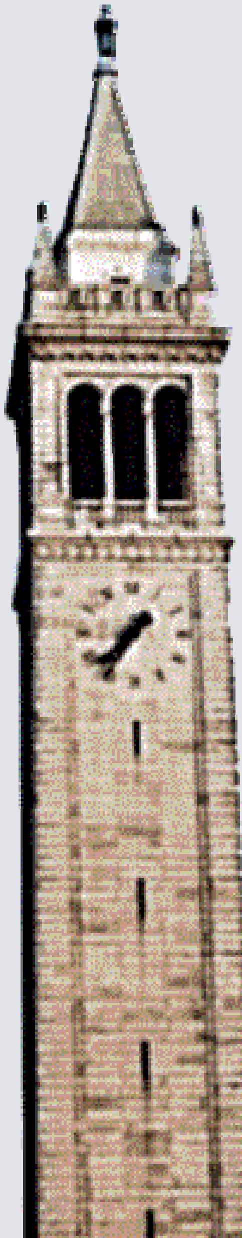
Coupled Interface Modules for Heterogeneous Composition

Edited and presented by
Ethan Jackson
ISIS, Vanderbilt University

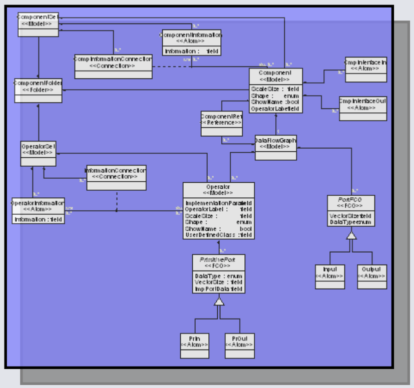
Chess Review
November 21, 2005
Berkeley, CA



Semantic Units and DSMLs

- DSMLs define a *structural semantics* or abstract syntax via a metamodel.

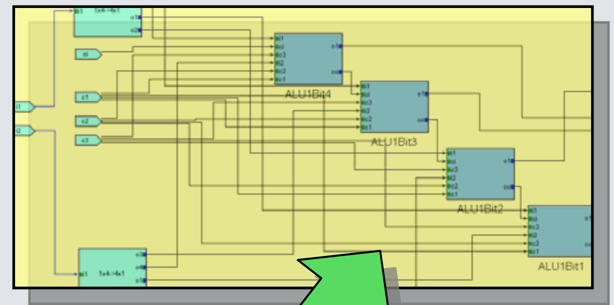
To a model



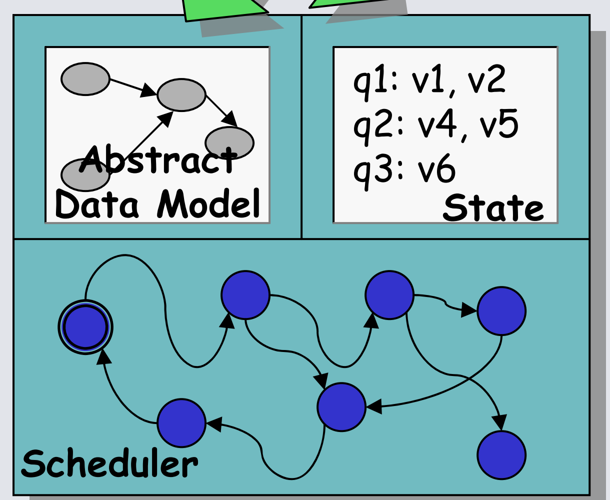
From a metamodel

Model Construction

(via GME)



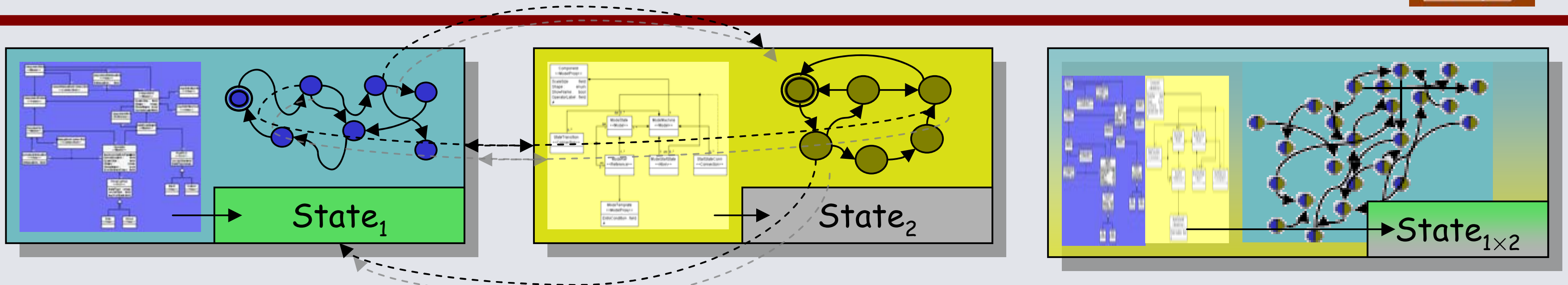
Semantic Mapping
(via graph transformation)



- Semantic units map models to initial conditions of an abstract state machine (ASM)

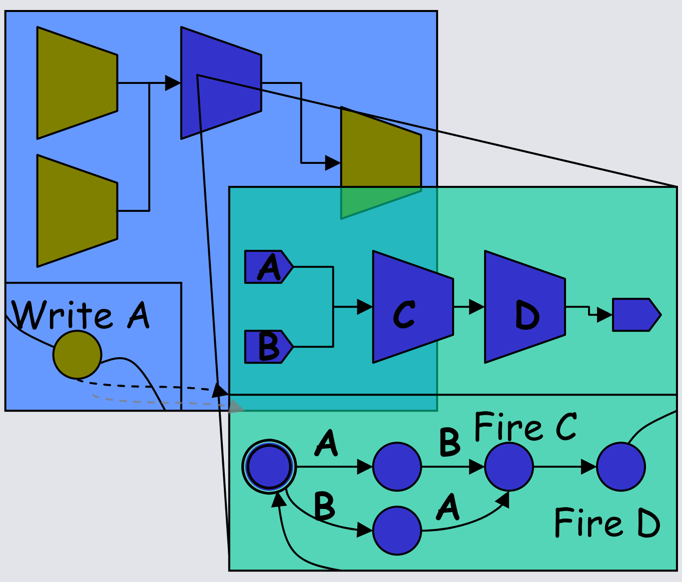
- Can leverage well-understood properties of FSMs while preserving domain specificity.

Is Composition Easy?



State Space: **Im** $M_{s1} \times$ **Im** M_{s2} Initial state = (State$_1$, State$_2$) ASM: $(F_1 \circ F_2)$

- The major problem is not expressiveness of automata composition, but rather the difficulty of unifying events (tags) while preserving abstractions.

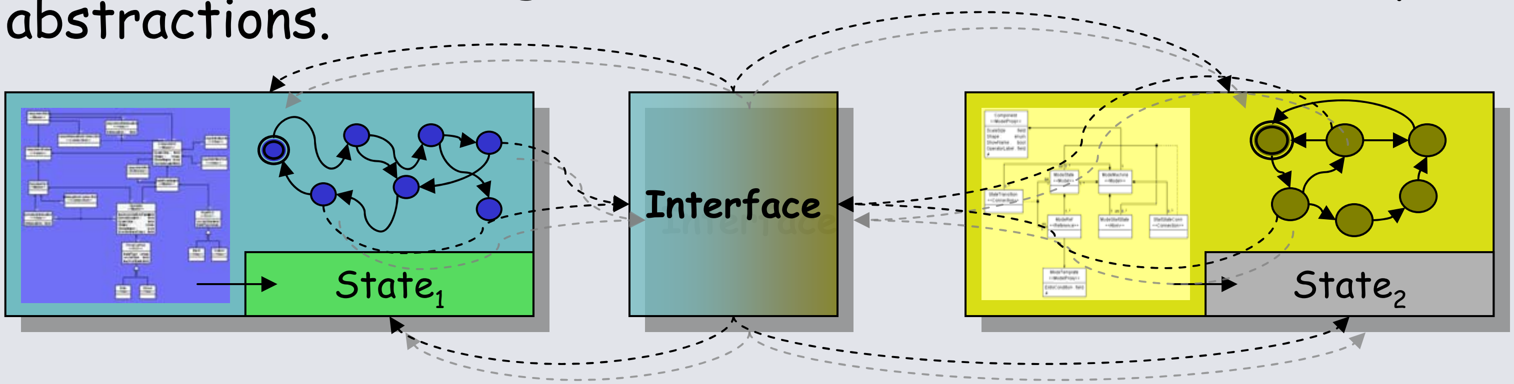


Encodes blocking condition

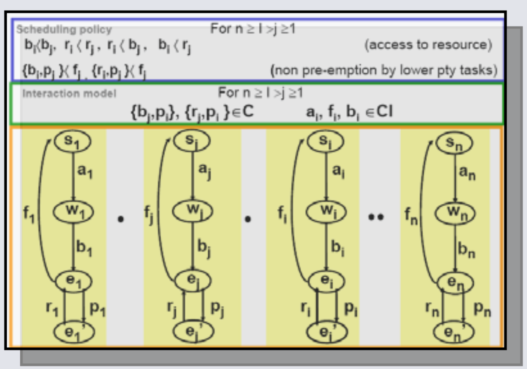
- We can check if the system blocks by performing a liveness analysis, but this ignores the obvious causality information, and is computationally harder.
- We lost abstractions by completely relying on automata composition, thus reducing problems to (generally) difficult reachability analysis.

Composition Through Interfaces

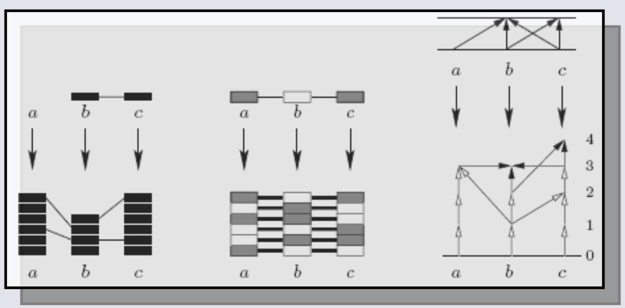
- Composition through interfaces allows us to insert another mathematical framework for describing semantics of communication that preserves the abstractions.



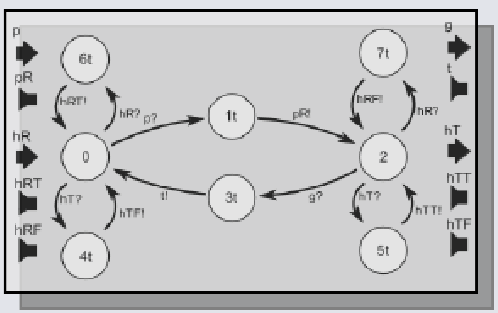
- From this perspective, there are already many existing candidates for a mathematical framework. We focus on the operational approaches.



Graphics taken from GÖSSLER, G., AND SIFAKIS, J. Composition for component-based modeling. In *Proceedings of FMCO02* (November 2002), vol. 2852, LNCS, pp. 443-466



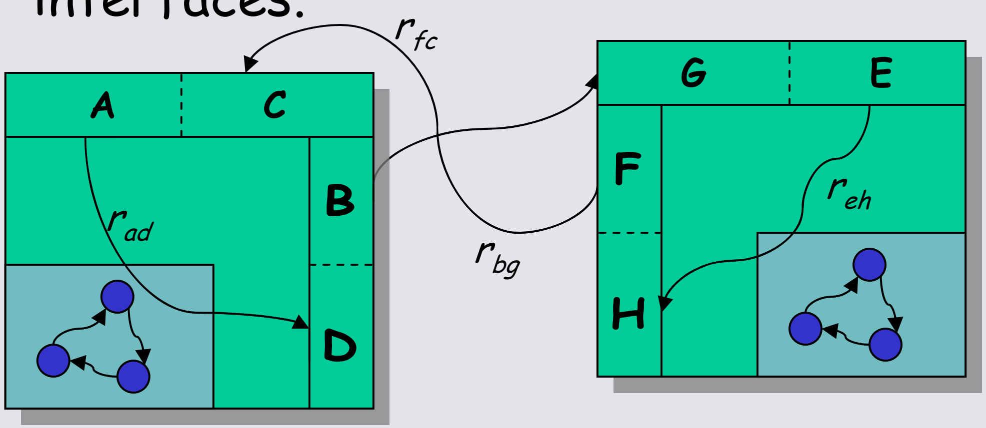
Graphics taken from BENVENISTE, A., CAILLAUD, B., CARLONI, L.P., and SANGIOVANNI-VINCENTELLI, A.L. Tag Machines *Proceedings of the Fifth International Conference on Embedded Software (EMSOFT), 2005*



LEE, E. A., and Xiong, Y., "A Behavioral Type System and Its Application in Ptolemy II," *Aspects of Computing Journal,* special issue on "Semantic Foundations of Engineering Design Languages."

Coupled Interface Modules

- Automata based methods have had success (e.g. Ptolemy II, Chic, Gratis II/GME), but, in general, do not scale. Other methods show promise, but lack mathematical maturity and generalizations.
- We propose to ground heterogeneous composition with the powerful machinery of linear algebra. Specifically, we use a generalization of vector spaces, called a *module*, to describe interfaces.



- Component interfaces are "vectors" in an event module, over which inner and tensor products are define. Event modules have equipotent bases, and operations are matrix multiplications.

$$(A^T \ M \ \Pi^k(i)P) = \left\|\Pi^k(i)P\right\|$$

Example of an interaction rule

- Components are composed through synchronous product of automata, and tensor products of interfaces and operations. One consequence: Interfaces can be factored.

$$I = (A_i \otimes A_j, P_i \otimes P_j, \\ M_i \otimes M_j + \mathbf{Re}(R_i \otimes R_j))$$