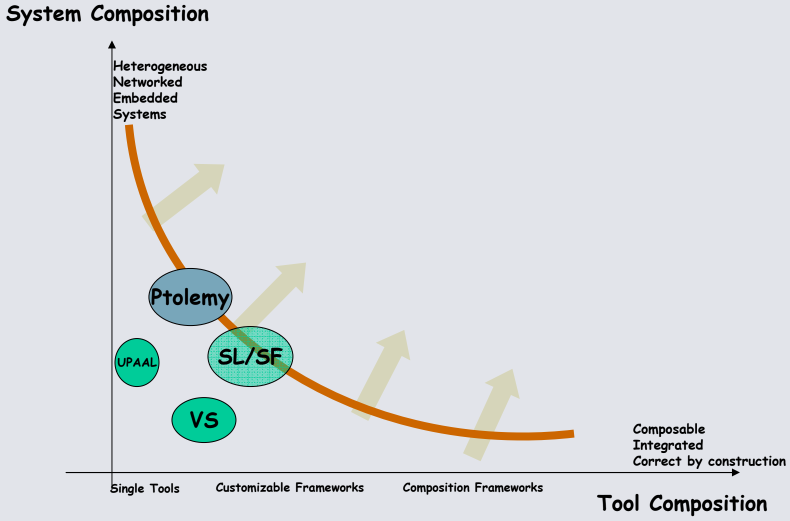
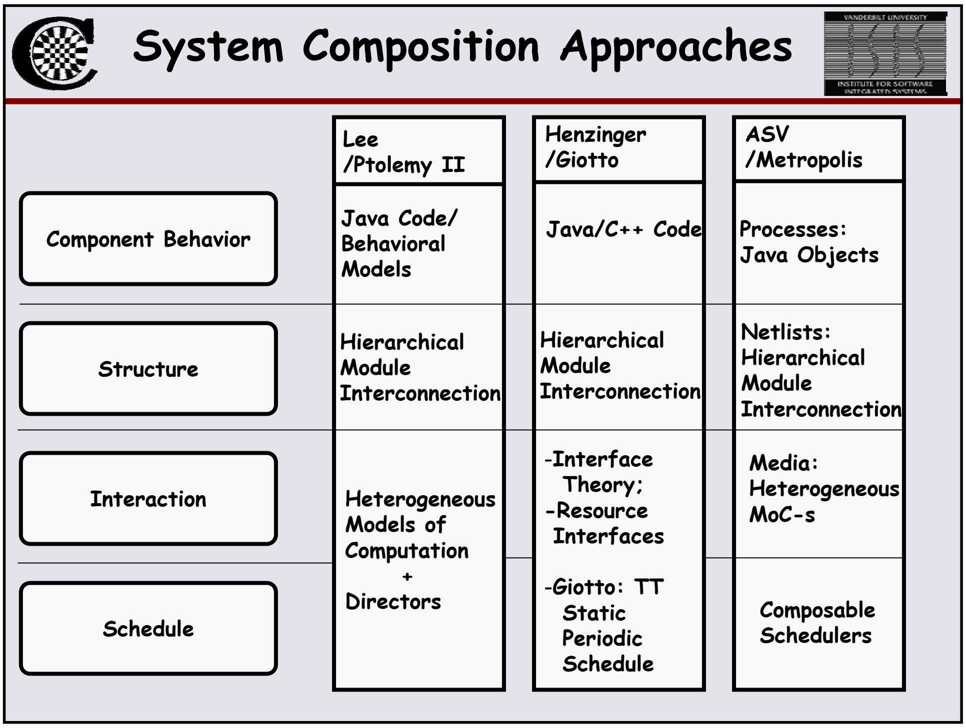
Advances In MIC Tools for Networked Embedded Systems Applications

Edited and Presented by
Janos Sztipanovits
ISIS, Vanderbilt University

Chess Review
May 11, 2005
Berkeley, CA

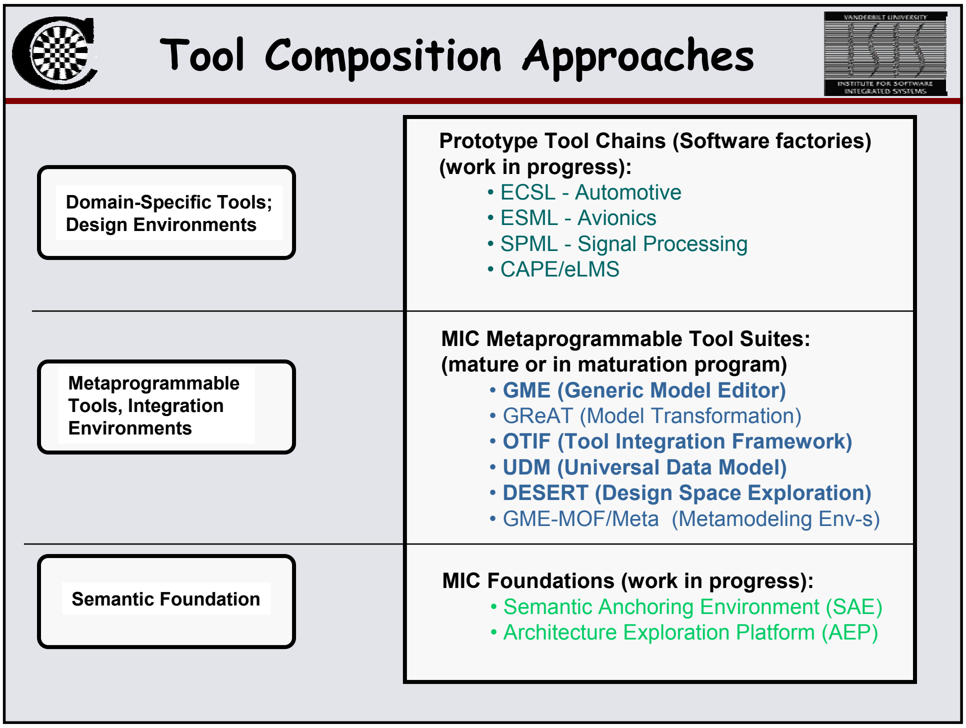
Two Dimensions of Model-Based Design





System Composition Approaches

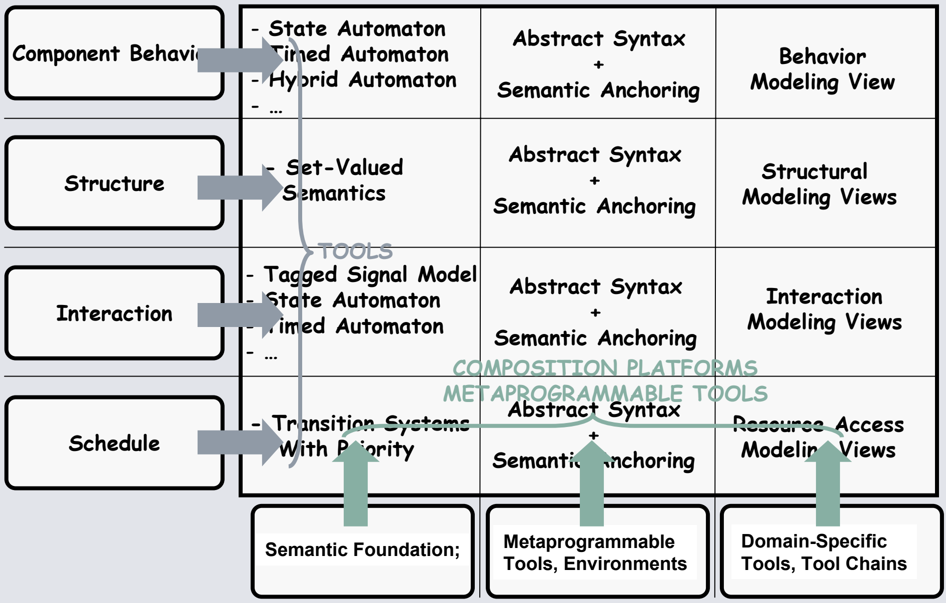
	Lee /Ptolemy II	Henzinger /Giotto	ASV /Metropolis
Component Behavior	Java Code/ Behavioral Models	Java/C++ Code	Processes: Java Objects
Structure	Hierarchical Module Interconnection	Hierarchical Module Interconnection	Netlists: Hierarchical Module Interconnection
Interaction	Heterogeneous Models of Computation + Directors	-Interface Theory; -Resource Interfaces	Media: Heterogeneous MoC-s
Schedule		-Giotto: TT Static Periodic Schedule	Composable Schedulers



Tool Composition Approaches

Domain-Specific Tools; Design Environments	Prototype Tool Chains (Software factories) (work in progress): • ECSL - Automotive • ESML - Avionics • SPML - Signal Processing • CAPE/eLMS
Metaprogrammable Tools, Integration Environments	MIC Metaprogrammable Tool Suites: (mature or in maturation program) • **GME (Generic Model Editor)** • GReAT (Model Transformation) • **OTIF (Tool Integration Framework)** • **UDM (Universal Data Model)** • **DESERT (Design Space Exploration)** • GME-MOF/Meta (Metamodeling Env-s)
Semantic Foundation	MIC Foundations (work in progress): • Semantic Anchoring Environment (SAE) • Architecture Exploration Platform (AEP)

Interrelations



Component Behavi	- State Automaton - Timed Automaton - Hybrid Automaton - ...	Abstract Syntax + Semantic Anchoring	Behavior Modeling View
Structure	- Set-Valued Semantics	Abstract Syntax + Semantic Anchoring	Structural Modeling Views
Interaction	- Tagged Signal Model - State Automaton - Timed Automaton - ...	Abstract Syntax + Semantic Anchoring	Interaction Modeling Views
Schedule	- Transition Systems With Priority	Abstract Syntax + Semantic Anchoring	Resource Access Modeling Views

TOOLS

COMPOSITION PLATFORMS

METAPROGRAMMABLE TOOLS

Semantic Foundation;

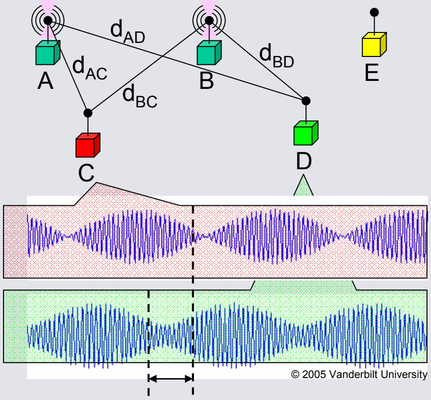
Metaprogrammable Tools, Environments

Domain-Specific Tools, Tool Chains

Networked Embedded Systems Challenges

- Fine-grain NEST applications (shooter location, Berkeley mote-based)
  - New ranging method + improved self-localization
- Heterogeneous large-scale networked embedded system applications (FCS)
  - Drastically increased model complexity
  - Significant need for semantic foundations for metamodeling
  - Shift toward dynamic architectures

Radio Interferometric Ranging



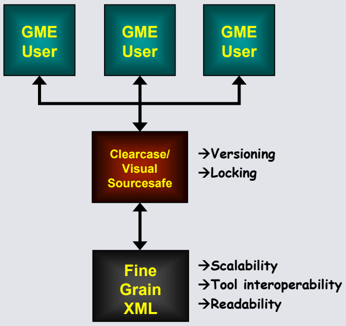
© 2005 Vanderbilt University

relative phase offset of beat frequency
= $(d_{AD}-d_{BD}+d_{BC}-d_{AC})$ modulo λ
where 65cm < λ < 75cm

- COTS radio chip (CC1000 on MICA2)
  - transmit frequency: 400-460 MHz
  - wave length: 65 cm < λ < 75 cm
  - adjustable in 64 Hz steps
- Two senders (A and B) transmit simultaneously
  - frequency separation: 100-800 Hz
  - duration of transmission: 32 ms
- Several receivers (C, D and E) measure interference
  - sample radio signal strength at 8.9 kHz
  - beat frequency: 100-800 Hz
  - samples per beat: 10-80
  - beats per transmission: 3-25
  - use time synchronization with 1 μs precision to correlate phase offsets
  - result is $(d_{AD}-d_{BD}+d_{BC}-d_{AC})$ modulo λ
    - d_{XY} is distance between X and Y
    - λ is wave length of carrier frequency
  - expected error is less than 5 cm
- Perform multiple measurements with different frequencies to obtain $d_{AD}-d_{BD}+d_{BC}-d_{AC}$

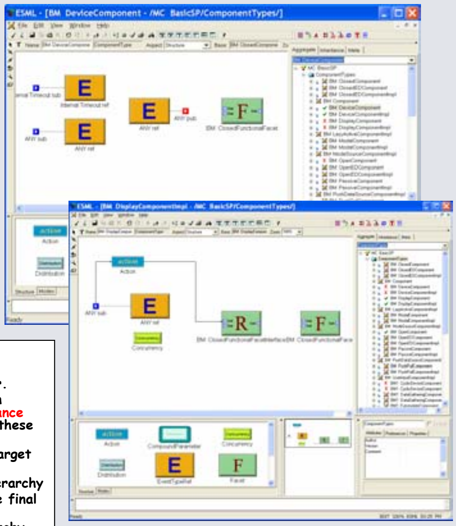
(Maroti, Ledeczi, 2005)

Multiuser Access and Model Versioning in GME



→Versioning
→Locking

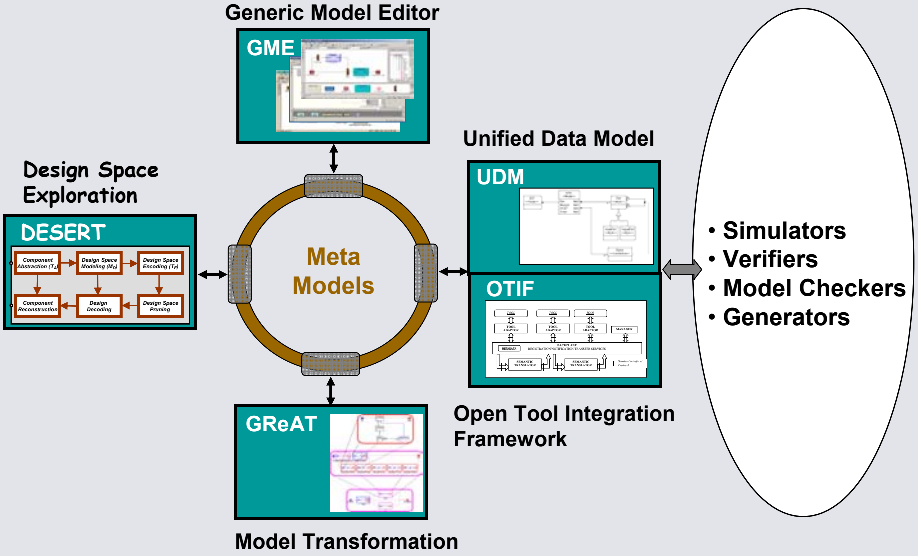
→Scalability
→Tool interoperability
→Readability



- Cache file storing all relationships locally
- Objects are not locked until user attempts to modify them
- Upon a modification attempt the model is checked out from the server.
- Model consistency must be ensured at all times. Models have very rich interrelationships. The Containment hierarchy and the type inheritance hierarchy form two orthogonal trees. Reference chains cut across these hierarchies. Strict locking policy must be enforced.
- The following relationships are followed recursively to check out the target objects also:
  - → All children (and consequently all descendants) in the containment hierarchy
  - → All references are followed in the reference chain all the way to the final target
  - → All derived types and instances are followed in the inheritance hierarchy

GME ensures that
→ Checked in models are consistent at all times
→ All available previous versions are consistent also

Metaprogrammable Tool Suite



Generic Model Editor

GME

Design Space Exploration

DESERT

Meta Models

Unified Data Model

UDM

OTIF

Open Tool Integration Framework

GReAT

Model Transformation

- Simulators
- Verifiers
- Model Checkers
- Generators

Semantic Foundation for Metamodeling

- Set-valued Semantics for Metamodels (Ethan Jackson)
  - Structural semantics for models
- Semantic Anchoring for DSML-s (Kai Cheng)
  - Specification of “Semantic Units”
  - Operational semantics
  - Asml