ESWG #1

UC Berkeley
Mobies Technology Project

Process-Based Software Components
for Networked Embedded Systems

PI: Edward Lee
CoPI: Tom Henzinger
Goals

• Project Goals
  - This project focuses on software creation for embedded systems with emphasis on systematically handling cross-cutting concerns such as safety, real-time, concurrency, liveness, adaptability, and integratability. The approach is component based, with compositional semantics and system-level type systems.

• Research Area
  - Heterogeneous modeling and design, code generation

• Targeted Development Context
  - The Ptolemy II tool will be used for modeling, simulation, and design of concurrent, real-time, embedded systems.
Products

• Ptolemy II
  - Java-based framework
  - Open source
  - Open architecture
  - Platform for domain-specific languages
  - Toolkit for code generators
  - GUI toolkit

• Domains
  - Continuous-time
  - Discrete-time
  - Discrete events
  - Communicating sequential processes
  - Process networks
  - Dataflow
  - Time-triggered
  - 3-D graphics
  - Synchronous/reactive
  - Publish/Subscribe

Preliminary at this time

Lee & Henzinger
Are actors active? passive? reactive?
• Are communications timed? synchronized? buffered?

A domain answers these questions in a disciplined way

Lee & Henzinger
Interfaces

• **Design Data**
  - MoML XML schema
  - Configurable GUI toolkit
  - Java API
  - **CORBA components**
    - Preliminary

• **Components for Component-Based Design**
  - Domain polymorphic
  - Data polymorphic
  - Type constraint propagation
  - Structured types (records, arrays)
  - System-level types
  - Real-time types

Lee & Henzinger
Ptolemy II Visual Syntax

GR domain (synchronous semantics)

CT domain (continuous-time semantics)
Hierarchical Heterogeneity vs. Amorphous Heterogeneity

Amorphous

Color is a communication protocol only, which interacts in unpredictable ways with the flow of control.

Hierarchical

Color is a domain, which defines both the flow of control and interaction protocols.

Lee & Henzinger
Example Execution

3-D Graphics

PtPlot tool
Example Domain

- **Giotto domain**
  - Synchronous semantics
  - Concurrent finite-state machines
  - Time triggered, multirate
  - Verifiable semantics

- **Realization in Ptolemy II**
  - About 1000 lines of code/comments in four classes
    - GiottoDirector
    - GiottoScheduler
    - GiottoReceiver
    - GiottoActorComparator
  - Leverages pre-existing FSM domain.
  - Leverages pre-existing GUI and actor library.
MoML Specification

MoML is an XML schema representing only the abstract syntax of component-based designs.

<?xml version="1.0" standalone="no"?>
<!DOCTYPE model SYSTEM "DTD location">
<br /></br />
</model>
Domain semantics defines communication, flow of control

Ptolemy II model

Schedule:
- fire Gaussian0
- fire Ramp1
- fire Sine2
- fire AddSubtract5
- fire SequenceScope10

Parser

Scheduler

Code generator

Abstract syntax tree

Target code

All actors will be given in Java, then translated to embedded Java, C, VHDL, etc.

First version created by Jeff Tsay.
Software Practice

- Object models in UML
- Design patterns
- Careful package organization
- Layered software architecture
- Design and code reviews
- Design document (to be a book)
- Nightly build
- Regression tests
- Sandbox experimentation
- Code rating
Ptolemy II Packages

- **kernel** (clustered graphs)
- **actor** (executable models)
- **data** (tokens, expressions)
- **vergil** (API for UIs)
- **graph** (graph algorithms)
- **math** (math algorithms)
- **plot** (plotting utilities)
- **domains** (modeling framelets)

Lee & Henzinger
Ptolemy II Key Classes

UML static structure diagram for the key classes in the kernel, kernel.util, and actor packages.
Code Rating & Coverage

Nightly build creates a web page displaying code rating, code coverage of regression tests, and results of the build and the tests.

<table>
<thead>
<tr>
<th>Package .ptolemy.kernel</th>
<th>Compilation Units</th>
<th>method</th>
<th>constructor</th>
<th>basic block:</th>
<th>branch</th>
<th>switch</th>
<th>ccat</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComponentEntity.java</td>
<td>8/8=100%</td>
<td>3/3=100%</td>
<td>23/25=92%</td>
<td>19/22=86%</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>ComponentPort.java</td>
<td>30/30=100%</td>
<td>3/3=100%</td>
<td>101/116=87%</td>
<td>74/92=80%</td>
<td>0/0=N/A</td>
<td>0/4=C</td>
<td></td>
</tr>
<tr>
<td>ComponentRelation.java</td>
<td>8/8=100%</td>
<td>3/3=100%</td>
<td>28/34=82%</td>
<td>26/32=81%</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>CompositeEntity.java</td>
<td>29/29=100%</td>
<td>3/3=100%</td>
<td>119/136=87%</td>
<td>93/104=89%</td>
<td>0/0=N/A</td>
<td>0/8=C</td>
<td></td>
</tr>
<tr>
<td>Entity.java</td>
<td>17/17=100%</td>
<td>4/4=100%</td>
<td>53/56=94%</td>
<td>28/32=87%</td>
<td>0/0=N/A</td>
<td>0/2=C</td>
<td></td>
</tr>
<tr>
<td>Port.java</td>
<td>17/17=100%</td>
<td>3/3=100%</td>
<td>65/73=89%</td>
<td>47/56=83%</td>
<td>0/0=N/A</td>
<td>0/4=C</td>
<td></td>
</tr>
<tr>
<td>Relation.java</td>
<td>10/10=100%</td>
<td>4/4=100%</td>
<td>35/40=88%</td>
<td>19/20=95%</td>
<td>0/0=N/A</td>
<td>0/5=C</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>119/119=100%</td>
<td>23/23=100%</td>
<td>424/481=88%</td>
<td>306/358=86%</td>
<td>0/0=N/A</td>
<td>0/2=C</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package .ptolemy.kernel.util</th>
<th>Compilation Units</th>
<th>method</th>
<th>constructor</th>
<th>basic block:</th>
<th>branch</th>
<th>switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute.java</td>
<td>4/4=100%</td>
<td>3/3=100%</td>
<td>19/20=90%</td>
<td>16/18=88%</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>ChangeListener.java</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>ChangeRequest.java</td>
<td>3/3=100%</td>
<td>1/1=100%</td>
<td>9/14=64%</td>
<td>5/14=35%</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>CrossRefList.java</td>
<td>17/17=100%</td>
<td>6/6=100%</td>
<td>66/75=88%</td>
<td>66/78=84%</td>
<td>0/0=N/A</td>
<td></td>
</tr>
<tr>
<td>DebugEvent.java</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td>0/0=N/A</td>
<td></td>
</tr>
</tbody>
</table>
Interaction Mechanisms

• Software releases
  - December 2000 (0.5 beta)
  - February 2001 (0.5)...

• CVS access over the internet
  - For the brave, with whom we work closely

• Kluwer book (est. March 2001)
  - Developers guide
  - UML object models of everything

• Ptolemy Miniconference
  - March 22-23, 2001 Claremont Hotel, Oakland
Near Term Schedule

- **Software releases**
  - December 2000 (0.5 beta)
  - February 2001 (0.5)
  - December 2001 (0.6 beta)
  - February 2002 (0.6)

- **Milestones**
  - Record types (February 2001)
  - Miniconference (March 2001)
  - System-level type definitions (Summer 2001)
  - Real-time type definitions (Summer 2002)
  - ...
None
- Source code provided.
- Can be commercialized.
- Can be used in commercial products.
Required from OEP

**Domain definitions**
- What is a component?
- How do components interact?
- Disciplined interactions, for
  - understandability
  - verifiability