NASA IV&V Perspective on Recent Trends in Executable Models

This poster focuses on the recent trend in UML simulation toolkits where tools interpret models instead of relying on generated code. This is an important development since it requires minimal configuration, can be used earlier in the lifecycle and can evolve as the design matures.

Model-Driven Validation and Verification

Model validation starts during design phase, continues throughout design constraints included as part of model, assessed via automation automated analysis (i.e., measuring coupling and cohesion, dependency analysis) runtime model checking: values, constraints, parametric conditional breakpoints (achieve desired state and assess) post-simulation analysis

OMG's fUML
http://www.omg.org/spec/FUML/
Semantics Of A Foundational Subset For Executable UML Models (fUML)
A virtual machine for executing UML Activities for verification. It supports structural and AML's System Description Language
A generic execution environment based on BaseS state charts. Useful for event-driven systems, control systems, view navigation, and system interactions. SCXML can be used for code generation.

W3 SCXML
http://www.w3.org/TR/scxml/
State Chart XML (SCXML); State Machine Notation for Control Abstraction
A generic execution environment based on BaseS state charts. Useful for event-driven systems, control systems, view navigation, and system interactions. SCXML can be used for code generation.

Parametrics
http://www.omg SysML.org/
part of the OMG Systems Modeling Language
SysML Constraint blocks specify physical properties of a system or system performance expectations. Model simulations can evaluate and track critical parameters such as size, weight, speed, power, temperature and others throughout the system life cycle. Some tools integrate with external multi solvers.

AADL
http://www.aadl.info/
Architecture Analysis & Design Language
Is designed for the specification, analysis, and automated integration of real-time performance-critical (timing, safety, schedulability, fault tolerant, security, etc.) distributed computer systems. It facilitates the analysis of system designs (and systems of systems) prior to development and supports a model-based model-driven development approach throughout the system life cycle.

Tool Feature Comparison

<table>
<thead>
<tr>
<th>Product</th>
<th>Metamodel(s)</th>
<th>Execute Activity</th>
<th>Execute State Machines</th>
<th>Model Audits</th>
<th>Visual Debugger</th>
<th>OMG XML</th>
<th>SCXML</th>
<th>AADL</th>
<th>SysML, Parametrics</th>
<th>Domain/External Classes</th>
<th>Requirements Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td>MagicDraw Cameo Simulation Toolkit</td>
<td>UML2, SysML</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>IBM Rational Software Architect</td>
<td>UML2, SysML</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>IBM Rational Rhapsody</td>
<td>UML2, SysML</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SimCafe DSL</td>
<td>UML2, SysML</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>NASA IV&amp;V Model Animator</td>
<td>UML2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

MagicDraw Cameo
Commercial
Key executable model features:
- Activity execution (OMG™ 3.1.2, standard)
- Model transformation (UML2, SCXML, standard)
- Multi-threaded execution (UML2, SCXML, standard)
- Full featured model debugger
- Full featured model animator
- Robust animation and visualization
- Mode-driven test case and scenario generation

IBM Rhapsody
Commercial
Key executable model features:
- Application generation for C, C++, Java and Ada
- Application and model integration
- Automated testing of code and execution test cases
- Simulated execution framework (UML2, SCXML, standard)
- Domain-specific language support for generic C, JAVA C++, JAVA C, and W3C SCXML for embedded systems
- Dynamically executable and executable SysML parameter diagrams
- Event-driven, model driven, model-based development

Rational Software Architect
Commercial
Key executable model features:
- Execute UML, behavior models to simulate the behavior
- Animate behavior diagrams during execution
- Control execution using breakpoints and other commands
- Animate composite structure and subdiagrams
- Control execution using breakpoints and other commands
- Animate behavior diagrams during execution
- Model-driven analytics
- Run-time monitoring
- Domain-driven processes models

Parametrics
Commercial
Key executable model features:
- Activity execution (OMG™ 3.3.3, standard)
- Model transformation (UML2, SCXML, standard)
- Multi-threaded execution (UML2, SCXML, standard)
- Full featured model debugger
- Full featured model animator
- Robust animation and visualization
- Mode-driven test case and scenario generation

TopCased
Open Source
Key executable model features:
- Activity execution
- Model transformation (UML2, SCXML, standard)
- Model-driven analytics
- Mode-driven test case and scenario generation
- Domain-driven processes models

Model Animator
Developed by NASA IV&V
Key executable model features:
- Activity execution (OMG™ 3.3.3, standard)
- Model transformation (UML2, SCXML, standard)
- Model-driven analytics
- Mode-driven test case and scenario generation
- Domain-driven processes models

Presented by Tom Gullion <gullion@ivv.nasa.gov>