IDM STANDARDS AND SYSTEMS SIMULATION

- SysML France -

Jérémie Tatibouët  
CEA LIST / DILS / LISE  
eremie.tatibouet@cea.fr

Arnaud Cuccuru  
CEA LIST / DILS / LISE  
arraud.cuccuru@cea.fr
Agenda

- COMPLEX SYSTEMS SPECIFICATION AND SIMULATION
- WHY USING OMG STANDARDS
- OMG STANDARDS IN SIMULATION: SEMANTIC FORMALIZATION
- PRECISE SEMANTICS OF COMPOSITE STRUCTURES
- PRECISE SEMANTICS OF UML PROFILES
**What?**

Development of Complex and Critical System (of Systems)

**Involve**

- Multiple engineering disciplines
  - Many different methods, languages and tools.
- Quality-of-Service-constrained systems
  - E.g., Resource, energy, and time constraints.
- Complex and often contradictory requirements/concerns

**Complex / Heterogeneous / Related to real-world.**
LISE: Laboratory on model-based engineering for embedded systems

A laboratory of ~40 persons

- 23 permanent members
- ~20 non-permanent members including PhD students, post-docs and CDDs

Main application domains and related partners

- Automotive: e.g., Renault, PSA, Delphi, Continental, Magnetti-Marelli.
- Railways: e.g., Alstom Transport.
- Avionics: e.g., Airbus, Thales, Esterel Technologies.
- Robotics: e.g., Thales, Dassault Aviation.
- Telecom, Consumer/microelectronics: e.g., STmicroelectronics and Ericsson.
In terms of tools and methods

- **Diapason:**

- For authoring and managing models

- Initiate new specific design workbenches
WHY USING STANDARDS?

Usage of standards participates to cost and risk reduction
- By making easier communication/exchanges between various stakeholders involved in process developments
- By improving tool interoperability
- Availability of educated engineers / Availability of education materials
- Standards enable vendor independence
  - Users have a choice of different vendors (no vendor “tie-in”)

Standards have traditionally provided major boosts to technological progress
- By fostering vendors to compete and improve their products

This is why the Object Management Group (OMG) has created the Model-Driven Architecture initiative:
- A comprehensive set of standards in support of MDE including standard modeling languages as UML2, MARTE and SysML.
• General purpose modeling languages
  - **UML**: Structure/behavior, historically software oriented, extendible using profiles
  - **SysML**: Intersection with UML + requirements /traceability + parametrics

• RTE domain-specific languages
  - **UML profile for MARTE**:
    . Software + (functional) hardware modeling
    . Modeling of extra-functional properties (NFP, VSL, Libraries)
    . Analysis models (scheduling, performance, etc.)

• Using these languages in a simulation process
  - Combine with existing formalisms and tools
  - Probably require some extensions (e.g., SysML4Modelica)
  - Need to deeply understand their meaning!
What is simulation?
- Experimentation on models through well-defined scenarios

Why simulating?
- Help engineers to figure out the constraints that their models place on the runtime structure and behavior of the modeled system
- Natural practice: Model, execute, observe results, draw conclusions, refine/correct models, etc.
- Complementary to formal analysis which may take time or fail on large models

For a meaningful simulation process
1. Modeling formalism(s) must be suited to the representation of the system under study
2. Simulation engines must reflect semantics of the input models
   Need to formalized semantics four our languages
3. Models must be “accurate abstractions” of the real system
OMG STANDARDS IN THE (SIMULATION) LOOP

What happens at interfaces?
Communication / synchronization principles?

What happens inside each component?

Different parts of UML model specified with well-suited formalisms
**fUML**
- Foundational UML (fUML) is an executable subset of standard UML with formal/operational semantics.

**Alf**
- Textual surface representation for UML modeling elements with the primary purpose of acting as the surface notation for specifying executable (fUML) behaviors within an overall graphical UML model.
- Also provides an extended textual notation for structural modeling within the fUML subset.

**UML2.5**
- Complete revision of its text description to simplify its presentation and disambiguate as much as possible its semantics.

**Precise semantics of UML Composite Structures RFP**
- Solicit a new specification defining a precise semantics for UML composite structures and their extensions.
- Containing two dedicated appendix for both MARTE and SysML.
PRECISE SEMANTICS OF UML COMPOSITE STRUCTURES

Response to Request for Proposal ad/11-12-07

2\textsuperscript{nd} Revised submission (ad/13-11-04)

Arnaud Cuccuru, CEA LIST / DILS / LISE
OMG Technical Meeting, Santa Clara, 12/2013
REMINDER ON OMG STANDARDS: FUML AND ALF

UML

fUML extended with composite structures modeling

Composite Structures

fUML (Classes + Activities)

Semantic mapping

Syntax

Semantics

Extensions for subsets of:
- MARTE HLAM
- SysML Ports and Flows

Extensions for composite structures

fUML Execution Model

Alf (Action Language for fUML):
- Textual surface notation for the fUML subset
THE SUBMISSION TEAM

Submitters

- International Business Machines (Eldad Palachi)
- Model Driven Solutions (Ed Seidewitz)
- No Magic, Inc. (Nerijus Jankevicius)
- THALES (Laurent Rioux)

Supporters

- CEA (Arnaud Cuccuru, Sébastien Gérard)
- EADS (Yves Bernard)
- INCOSE (Sanford Friedenthal)
- JPL/NASA (Nicolas Rouquette)
- Lockheed Martin (Michael Chonoles)
- NexJ Systems (Ed Shaw)
- NIST (Conrad Bock, Peter Denno)
- Simula Research Laboratory (Bran Selic)
- Universidad de Cantabria (Julio Medina)
INTEGRATION TO PAPYRUS

Semantic Model

UML syntax subset

Implementation of

Animation Framework

Moka

Class

ActivityNode

Object

ActivityNodeActivation

+ types

[?]

+ node

[0..1]

Research

Understand impacts of his modeling choices by looking at the model at runtime
EXAMPLE: SYNCHRONIZING ACTIVE OBJECTS WITH SIGNALS

1. Class diagram (~ BDD)
   - Structure
     - <Signal> Ping
     - <Signal> Pong
     - <Signal> Start
     - <Interface> Ping
     - <Interface> Pong
   - Behavior
     - A Classifier Behavior
     - B Classifier Behavior

2. Composite structure diagram (~ IBD)
   - Game
   - A
   - B

3. Activity diagrams
   - activity LaunchPingPongExample()
     - game = new Game();
     - game.b.Start();
     - game.a.Start();

4. Alf Specification on an Activity
   - Can be compiled into an equivalent, executable FUML Activity.
   - Methodological aspect:
     - The 2 notations can be combined
     - Use the most appropriate when needed

Resources:
http://www.slideshare.net/seidewitz/programming-in-uml-an-introduction-to-fuml-and-alf
PRECISE SEMANTICS OF UML PROFILES

Jérémie Tatibouët, CEA LIST / DILS / LISE
Ongoing Phd
UML Profiles Semantics

- Stereotypes imply modifications/completions to the original UML semantics

- Signal created by this action is sent to the object provided on its target input pin

- Signal created by this action is sent to every object classified under a specific type

- There are no constraints on concurrent call over class features

- Only one active object can access class features at a time. Others are suspended
UML Profiles Semantics

- Profiles usually have a semantics described in prose [1] [2]
- Not a suitable formalism to perform simulation on models
- No systematic way to specify the semantics of a profile [3]

References

- [3] - Graph, S and Ober, I. How useful is the UML profile SPT without Semantics? SIVOES. 2004
<table>
<thead>
<tr>
<th>Code generation</th>
<th>Model Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="#" alt="User" /></td>
<td><img src="#" alt="P" /></td>
</tr>
<tr>
<td><img src="#" alt="Profile Designer" /></td>
<td><img src="#" alt="L" /></td>
</tr>
<tr>
<td><img src="#" alt="Code Generator" /></td>
<td><img src="#" alt="S" /></td>
</tr>
<tr>
<td><img src="#" alt="Semantics" /></td>
<td><img src="#" alt="Abstract State Machines" /></td>
</tr>
<tr>
<td><img src="#" alt="JAVA" /></td>
<td><img src="#" alt="PDF" /></td>
</tr>
<tr>
<td><img src="#" alt="C++" /></td>
<td><img src="#" alt="M" /></td>
</tr>
</tbody>
</table>

**UML PROFILES SEMANTICS (3)**
Profiles semantics can be captured as extensions of the fUML semantic model.
Benefits compared to the state-of-the-art

- Profiled models get executable without any intermediate step(s).
  - Simulation is possible from design phase

- Rely on a standard supported by the OMG.
  - Specifications automatically comply to fUML semantic model design
  - Specifications are easiest to maintain
  - Semantics can be verified using fUML mathematical bases

- Approach is entirely model-driven.
  - Relation between stereotypes and semantics is well identified.
  - Structural design of the semantics is specified with a class model – Well known by modeling communities
  - Behavior is specified using Alf (activities) – Close to standard programming
  - No interactions with fUML implementation is required
  ➔ Approach is integrated into Papyrus

Publication

Simulation community can get benefit from MDE…

- Simulation requires software, methods, and tools
- We believe MDE can help for this.

MDE methods and tools require simulation

- Not replace: Reuse and collaborate

From the perspective of OMG standards

- Key enabling challenge: Formalize semantics
- Good trend: fUML, Composite Structures
- A step farther: precise semantics for UML profiles.

Standard semantics are the key to standard interoperability
Acknowledgments to the LISE team for their direct and indirect contributions to this presentation.
Papyrus is the official open-source Eclipse UML2 modeling tool:

www.eclipse.org/papyrus

- Papyrus provides a complete graphical editor for both UML and SysML standards based on the MDT::UML2 component for its repository.
- Papyrus addresses both key features expected from a UML2 graphical editor: modeling and profiling.
- Papyrus is highly customizable and extensible enabling DSML definitions based on standard UML profiles!
- Papyrus provides a support to MARTE 1.1 (including a rich text editor for VSL).