Adele / OSATE Synchronization

Dominique Blouin*
Alain Plantec*
Pierre Dissaux†
Franck Singhoff*

*Lab-STICC, Université de Bretagne Occidentale
Brest, France
†Ellidiss Technologies
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
The Adele Graphical Editor

- Graphical editor for AADL developed by Ellidiss.
- Stores AADL models in its own format (meta-model).
- AADL code generated from the Adele model (on demand by user).
- Built on top of the TOPCASED Modeling Framework (TMF).
Adele Issues

- Does not cover the complete AADL V2 language.
  - Modes
  - Flows
  - Prototypes
  - Etc.

- Would require a significant additional development effort to cover all AADL language.
  - Is it really needed, knowing that some AADL constructs are better edited in text.

- Solution: develop a seamless integration between the tools.
  - Use OSATE for textual edition.
  - Use Adele for graphical edition.
  - Replace existing code generation with state of the art model synchronization techniques.
The Adele Joint Project

- 10 months project (funded by US Army USAITC-A grant).

Objectives:

- Develop a synchronization layer between Adele and OSATE:
  - Bi-directional.
  - Real **synchronization**; do not regenerate the objects but update them to maintain consistency.
  - Transparent for the user.

- Make the synchronization layer reusable as much as possible for integration of other tools with OSATE.

- Implement some missing AADL constructs if time allows:
  - Modes.
  - Flows.
  - Etc...
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
Adele Meta-Model

Project Dependent

Properties Configuration (ODS)

Language Dependent

Adele Specialization

Invariant

Adele Kernel
Adele Kernel Layers

- Valid for any hierarchical box-arrow diagram.

- Supports editing of both the declarative and instance models.

```
Component Implementation
     ↓
     1
Subcomponent
     1
      * 
Classifier

Component Implementation
     1
Subcomponent
     * 
Multi-levels Hierarchy
     1
Instance
```
Impact on Diagrams

Declarative Bottom-up Modeling

Instance Top-down Modeling

Adele / OSATE Model Synchronization

AADL Standards Meeting, July 9th, 2013
Adele Meta-Model

- 48 classes (7 in Kernel and 41 in specialization layer).

- Implements the AADL categories as specializations of kernel entities:
  - Components
  - Features
  - Connections

- Makes changes of the AADL standard easy to handle.
  - Example: moving from AADL v1 to AADL v2.

- Enables flexible addition of graphical sub-languages.
  - Example: Behavior Annex states-transitions.

- Enables support of other DSMLs.
OSATE Meta-Model

- 273 classes (declarative model only).
- Strongly typed (much more than Adele).
  - Stronger semantics.
- E.g.: Rules for features.
OSATE Subcomponents Composition Rules
OSATE Meta-Model

- 273 classes (declarative model only).

- Strongly typed (much more than Adele).
  - Stronger semantics.

- OSATE:
  - Good for analyses.

- Adele:
  - Good for diagramming.
Model Synchronization Tools

- Very few tools can perform model synchronization.
  - The most well known tools are all based on Triple-Graph Grammars (TGG).

- Three main tools currently developed:
  - TGG Interpreter (University of Paderborn).
  - eMoflon (Technische Universität Darmstadt).
  - MoTE (Hasso-Plattner-Institute, University of Potsdam)

- Compared in:
Generic Architecture of TGG Tools

From Hildebrandt et al., 2013.
Comparison of Tools Features

- All tools have their strengths and weaknesses and the best tool depends on the use case.

- TGG interpreter and MoTE:
  - Are fully EMF based (like Adele and OSATE).
  - Can perform model synchronization.

- eMoflon:
  - Synchronization not implemented yet.
  - Specification and integration environment defined as plugins of distinct tools.
    - TGG specification in Enterprise Architect.
    - Transformation integration in Eclipse EMF.
    - Operational rules implemented as EMF Java classes.
Performances

- Both MoTE and eMoflon have good performances.
- TGG interpreter about 10 times slower, but still remains fast.

From Hildebrandt et al., 2013.
Tool Selection

- MoTE seems to be the best compromise for Adele / OSATE.
- Fully EMF based like Adele and OSATE.
- Fast.
- Support available.
- Version 2 under development.
  - Improved TGG expressivity and performances.
- Transformation of TGG towards another tool should be possible since similar language.
Demo

- TGG development environment.
- A few rules and their corresponding Story Diagrams (SD).
- Generation of SDs.
- Synchronization at work.
- Global Model Management.
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
Features and subcomponents are duplicated in Adele:
- From type in implementations and subcomponents.
- From extended and refined types.

Useful for system edition mode.

Duplication handled by the Adele editor, as user navigates through the diagrams.

Connections refer to the duplicated features and subcomponents.
The MoTE synchronization tool can perform batch transformations.  
- Used when no corresponding specification exists (e.g. the first time an Adele diagram is saved).

Features and subcomponents must be duplicated at AADL to Adele transformation time to be able to transform connections.

The MoTE TGG language provides post-creation call actions  
- E.g.: Can call a Java method after the object is created.

Post-creation actions were not generated in the SD rules so we had to implement it.
Reference Problems

- Model elements are often spread in different files (resources) and need to refer to each other:
  - Components of the instance model referring to types of the declarative model.
  - RDAL requirements assigned to AADL model elements.
  - Adele graphical model elements to Adele semantic model elements.
  - Traceability model used to synchronize Adele and OSATE.
Reference Problems (cont’d)

- The Adele and OSATE meta-models do not identify any attribute in their classes that can be used as IDs for cross file references.

- In such case, by default, EMF uses an indexes mechanism.

- E.g. for instance to declarative model:

```
<property href="../Plugin_Resources/SEI.aadl#@ownedProperty.20"/>
```

- As model elements are added or removed, all references to these elements are shifted and wrong elements are referred.
After discussion with Peter, it was decided to use a symbolic name mechanism to refer to AADL model elements of the declarative model.

- Implemented as an Xtext fragment provider using qualified names.
- Next task could be to agree on file system independent URI for AADL specifications.

The Adele meta-model was reviewed to declare the existing id attribute as ecore ID, so that it could be used as reference.
Cross File References in MoTE

- The MoTE tool operates on a root resource that is assumed to contain all model elements.
  - Model objects that are not within the root resource are not matched.

- This is obviously not the case for the AADL.

- A workaround (as suggested by the MoTE team) consists of treating the external references as attribute assignments.
  - Implement a translation function between Adele and AADL model elements in Java.

- In the long term, the tool could be adapted to handle cross file references in a transparent way.
The expressivity of the TGG language needs to be improved.

- The Adele / AADL grammar contains more about 150 rules.
- Many are just duplicated for each component category.
- E.g.: The set Adele / AADL TGG rules for component type features:
Rules Duplication (cont’d)

- Currently, Java classes are developed to duplicate the rules automatically.

- A TGG rule inheritance mechanism would be nice.
EMF Compare Issues

- MoTE requires to maintain the object instances for synchronization.
Some editors like do not maintain the objects references when edited.
  ◦ E.g.: the Xtext editor repars-es the objects every time the document is modified.

We use EMF Compare to merge the changes from outside into the MoTE resources cache (resource set).

Two issues have been discovered in EMF compare.
Error in counting the non null features:
- AADL no-like features (noFeatures(), noSubcomponents, noConnections, etc.) always counted as set.
- Use Ecore eIsSet instead of testing null value.

Error in merging references to newly created object with deep hierarchy:
- Reference is set as the object of the source resource while it should refer the merged target resource.
- Issue currently being fixed.
The transformation of TGG rules into story diagrams does not seem to scale very well.

No incremental compilation of TGGs implemented in MoTE.

Investigated how it could be implemented.
- Reference problem again.
- The SD use UUIDs and reference from existing SD to regenerated SD are broken.
- Using symbolic ID’s like AADL would fix the problem.
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
TOPCASED Modeling Framework Light

- Developed a lightweight version of TMF.
  - Removed dependencies on Papyrus and other plugins.
    - Old Eclipse update manager (before P2).

- Now using our own branch of TMF (TMF2).

- A few issues were fixed:
  - Dependency problem in the code generator for the Juno platform.

- Improved TMF features:
  - Specification of super class of generated code in model.
  - Specification of container feature of semantic model elements for graphical elements.
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
The Adele meta-model has been reviewed before developing the TGG:

- Cleaned the dependencies on Adele in the SPICES kernel.
- Removed the “ADELE_” prefix from the meta-model class names.
  - TMF editor models updated and editor regenerated.
- Make use of opposite reference mechanism for parent / children.
- Reference problem.

Other various issues have been fixed:

- Synchronization between semantic and graphical models.
- Load resource action.
- Connection legality rules between ports.

Other minor issues with OSATE is addition to the reference problem.
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
Next Work

- EMF Compare issue.

- Adele / OSATE Synchronization Layer:
  - Feature refinement.
  - Complete connection rules.
  - With clause.
  - System edition mode.
  - Functional tests.
  - Documentation.
  - Private / public package sections.
  - Comments problem.

- Adele:
  - Synchronization of the semantic and graphical models.
  - Feature group declarations and classifier management.
  - Flows, Modes, etc.
Next Work (cont’d)

- Potential work on MoTE:
  - Partial TGG grammar compilation.

- Adele sources and releases will be hosted at Ellidiss.
Agenda

- Project Summary
- Adele / OSATE Synchronization Layer
- Difficulties in Synchronization Layer Development
- TOPCASED Modeling Framework
- Issues with Adele and OSATE
- Next Work
- Conclusion
The development of the synchronization layer required a lot of effort.

Several open-source frameworks / tools required to be improved:
- TMF
- Adele
- MoTE
- OSATE

Probably the first complete synchronization use-case with TGG.
MoTE is a great tool but could benefit from a few improvements to reduce synchronization layer development effort.
- Cross-file reference.
- Rule inheritance.
- Incremental compilation of TGGs.

The result is promising since model integration is a recurrent problem.

Many of the efforts spent on Adele / OSATE synchronization can be reused for other tools.