Adele / OSATE Synchronization

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

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

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues in Synchronization Layer Tool
- Work on Adele
- Next Steps
- Conclusion
The Adele Graphical Editor

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

![Diagram of the Adele Graphical Editor workflow](image)
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 (e.g.: prototypes).
  - Is it really needed?
  - Some AADL constructs are better edited in text anyway...

- 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.
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
- The Adele / OSATE Synchronization Layer
- Demo
- Issues in Synchronization Layer Tool
- Work on Adele
- Next Steps
- Conclusion
Separated into 2 Layers

- Global Model Management Framework (GMM).
  - Developed during this project.

- Model synchronization:
  - Selected the Model Transformation Engine (MoTE).
  - From MDE Lab
  - PhD thesis of Stephan Hildebrand (Holger Giese).
GMM Layer

- DSML inspired from state of the art literature on Global Model Management / mega-models.
  - Mega-model: a model of models.
GMM Mega-model for OSATE

- Declares obligation relations for Adele-OSATE AADL models.
GMM Extension Layer for MoTE

- Uses the MoTE transformation engine.
IDE GMM Architecture

Eclipse Platform

Workbench

Editors Manager

Resources Manager

GMM Controller

Editor Adapters

GMM Engine

Workspace

myproject1

myresource.aadl

myresource.adelle

myresource.adelledi

Workbench Configuration

GMM Specification

Adele2AADL MoTE Sync. Relation

Objects Merger

Objects Cache

TGG Engine

Adle / OSATE Synchronization

AADL Standards Meeting, Sept 25th, 2013
Model Synchronization Layer

- 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)

- MoTE was selected for Adele-OSATE synchronization.
  - Fully EMF based.
  - Good performances.
  - Not mature 😞
Generic Architecture of TGG Tools

From Hildebrandt et al., 2013.
Graph Grammars

- Set of rules describing transformations of instance objects graphs*.

- Similar to classical grammars but applies to graphs instead of strings.

- Petri net example rule:
Graph Grammars

- Petri net example graph (instance objects graph).
TGG Rule for Component Types

Adele Domain

Adele 2 AADL Domain

AADL Domain

Adele / OSATE Synchronization
Operational Rules

- TGG are **declarative** rules.

- They must be transformed into something interpretable (or executable) to perform the synchronization / transformation.

- For MoTE, these are Story Diagrams (SDs).

- SDs are similar to UML activity diagrams:
  - Express behavior representing modifications of models.

- Interpreted of SDs at runtime maintains consistency between models.
Operational Rules

- For each TGG rule, 12 story diagrams are generated:
Actual Adele-OSATE TGG

- Consists of a set of 60 *generic* rules for:
  - Packages.
  - Component types.
  - Features.
  - Component implementations.
  - Subcomponents.
  - Connections.

- Translates into 714 SD.
  - (Only 6 SDs instead of 12 for the axiom rule (packages)).
  - 100 MB on disk.
  - Reduces to 7 MB when released as Eclipse plugin.
Agenda

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues with Synchronization Tool
- Work on Adele
- Next Steps
- Conclusion
Agenda

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues with Synchronization Tool
- Work on Adele
- Next Work
- Conclusion
Issues with MoTE

- Several improvements of MoTE were required to be able to use it with Adele-OSATE.

- Issues are related to:
  - Scalability.
  - Expressivity of the TGG language.
  - TGG development tool.
Scalability

- The AADL meta-model is strongly typed.
  - E.g.: subcomponents composition rules.

One property per subcomponent type
TGG Rules Constraints

- E.g. with subcomponents TGG rules:

  - Must be instantiable
  - Must be changeable
Issues

- Many TGG rules are required:
  - RuleSystemBusSubcompo2SystemBusSubcompo
  - RuleSystemDataSubcompo2SystemDataSubcompo
  - RuleProcessThreadSubcompo2ProcessThreadSubcompo
  - RuleProcessDataSubcompo2ProcessDataSubcompo

- In addition, every possible context must be handled:
Subcomponent Match Contexts

- Refined, not typed, typed as parent...
Scalability Issues

- Nearly 250 rules were expected for Adele-AADL.
- The MoTE SD generator did not scale to such number of rules.
- When modified to make it scalable:
  - Generates about 3GB of SDs!
  - Compressed to about 200 MB.
  - The SD interpreter was still working!
- Do not want to release such big data.
- Difficult to develop and maintain such number of rules.
Introducing Generic TGG Rules

Abstract

Derived

Class Rule Variable
Modified SD Interpreter

- Modified SD interpreter:
  - Runtime use of class rule variables to determine the type to instantiate.
  - Runtime determination of the property to set from property type constraints.

- Other fixed issues:
  - Post creation actions.
  - Cross resource references not handled.
  - Directional global pattern matching constraints.
  - Declare reverse navigation reference in TGG.
  - Model change listener reworked.
Potential Improvements for TGG Language

- A lot of duplication of information in rules.
  - Need for rule inheritance mechanism.
  - Rule variables should be shared across rules.

- Not easy to ensure all instance graph contexts are covered by the rules.
  - Provide automated context analysis?

- Adele meta-model needed to be changed to provide additional references for matching.
  - Improve expressivity to replace allow replacing meta-model properties by queries?
Issues with Tooling

- We added means to duplicate a diagram.
  - Eases the definition of rules for the various match contexts.

- Allowed editing constraints from EMF hierarchical tree editor.

- Open TGG diagram from EMF hierarchical tree editor.

- TGG rules validation should be improved.

- Allow the use of libraries when expressing constraints.

- Matching rules with attribute assignments used for references.
EMF Compare Issues

Eclipse Platform

Workbench

Editors Manager

Resources Manager

GMM Controller

Editor Adapters

GMM Engine

Workspace

myproject1

myresource.aadl

myresource.adele

myresource.adeledi

Workbench Configuration

GMM Specification

Adele2AADL MoTE Sync. Relation

Objects Merger

Objects Cache

TGG Engine

Adele / OSATE Synchronization
EMF Compare Issues (cont’d)

- 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 in the MoTE resources cache (resource set).

- Other issues have been fixed in EMF compare to account for synchronization.
  - Perform delete operations at the end of merge.
  - MoTE changes listener requires this for being able to repair the changes.
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.

Multi-levels hierarchy not well handled in merge.

The synchronization layer uses a customized EMF compare to take this into account.
- Customization extension points were already provided.
Agenda

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues with Synchronization Tool
- Work on Adele
- Next Steps
- Conclusion
Synchronization between Graphical and Semantic Models

- Adele uses a feature and subcomponent duplication mechanism to draw inherited features and subcomponents.

- A layer already existed in Adele to synchronize the duplicated features.

- It has been extended to a full synchronization of the graphical and semantic models.

- Layer currently under validation.
Agenda

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues with Synchronization Tool
- Work on Adele
- Next Steps
- Conclusion
Next Steps

- Make a first release.
  - Complete ongoing debugging tests (not stable enough to be released)
  - Fix comments problem.

- Adele / OSATE Synchronization Layer:
  - Define TGG grammar for System edition mode.
  - Implement automated synchronization tests (only transformation and mapping are currently implemented).
  - Complete documentation.
    - Cookbook on developing synchronization with MoTE.

- Adele evolutions:
  - Classifier management in system mode.
  - Integrate Adele documentation in online help.
  - Implement flows, modes, etc.
Agenda

- Project Summary
- The Adele / OSATE Synchronization Layer
- Demo
- Issues with Synchronization Tool
- Work on Adele
- Next Steps
- Conclusion
Conclusion

- The development of the synchronization layer was not easy.
  - We will not do as much as was originally planned in the project. 😞

- Required acquiring knowledge of several frameworks / tools:
  - Adele
  - TOPCASED Modeling Framework
  - MoTE
  - EMF Compare
  - OSATE

- One of the first real life case of synchronization with TGG:
  - Large complex meta-models.
Conclusion (cont’d)

- MoTE is a good tool but several improvements were required to make it work on large meta-models such as AADL.

- Promising approach for model integration, which is a recurrent problem in model-based engineering.

- The result from this project should be reusable for seamless integration of other tools.
References


References (cont’d)
