Generating JMI model transformation code from UML profile models for SDM

Aligning Graph Rewriting with MDA-light

Pieter Van Gorp
Hans Schippers

Formal Techniques in Software Engineering

{Pieter.VanGorp,Hans.Schippers}@ua.ac.be
Overview

1. Demo 1:
   - Fujaba Refactoring Cycle
   - Refactoring Implementation
   ✓ Pull Up Method

2. Problem: Tool Lock-In

3. Solution: (Reflective) MDA Light

4. Demo 2:
   - Integrated Tool Context
   - Pull Up Method Refactoring
   ✓ Pull Up Method
Fujaba Refactoring Cycle

1. Import Java Sources (by directory)
2. Rearrange diagrams
3. Execute Refactoring
4. New Sources regenerate
   Old Sources
Refactoring: Structural Design

Fujaba’s Metamodell

Example Refactoring Classes
Refactoring Behavior: SDM

Model Transformation as Graph Transformation

1. Bind method
   > (Hidden) Cast target
2. Bind container
   > Link Navigation
3. Bind stub
   > Link Navigation
4. Bind superclass
   > Link Navigation
5. Remove method
   from container
6. Add method
to superclass
Constraint Checking: SDM
1. **Demo 1:**
   - Fujaba Refactoring Cycle
   - Refactoring Implementation
     - Pull Up Method

2. **Problem: Tool Lock-In**

3. **Solution:** (Reflective) MDA Light

4. **Demo 2:**
   - Integrated Tool Context
   - Pull Up Method Refactoring
     - Pull Up Method
Problem: Vendor Lock-In

1. SDM could only be drawn in Fujaba
   - Industry Requires Commercial Support

2. Fujaba code generator assumes target repository complies to Fujaba API.
   - XMI + JMI + EMF + … desirable
Solution: Reflective MDA Light

1. SDM could only be drawn in Fujaba
   - Solution: UML Profile for SDM

2. Fujaba code generator assumes target repository complies to Fujaba API.
   - Solution: Pluggable Code Templates

Reflective Architecture
Reflective Architecture (cont.)
Example “Fujaba vs JMI Specific Code”

Fujaba vs JMI Specific Code:

JMI:

// delete link
method.setOwner(null)

// create link
method.setOwner(superclass);

PullUpMethod::execute (target: ASGElement): Void

```
// bind stub : UMLGeneralization
fujaba_I
while (!fujaba__Success) {
    try {
        stub = fujaba__stub;
        fujaba__Success = true;
    } catch (JavaSDMException fujaba__InternalException) {
        fujaba__Success = false;
    }
    fujaba__Success = !fujaba__Success;
    fujaba__stub = null;
}
```
1. Demo 1:
   - Fujaba Refactoring Cycle
   - Refactoring Implementation
     ✓ Pull Up Method

2. Problem: Tool Lock-In

3. Solution: (Reflective) MDA Light

4. Demo 2:
   - Integrated Tool Context
   - Pull Up Method Refactoring
     ✓ Pull Up Method
Integrated Tool Context

- Poseidon (or ...): Transformation Specifications
- SDM as UML 1.4 Profile
- MM as Class Diagram

NetBeans MDR

MOF Repository

Transformation Spec

Model to transform

XMI

AndroMDA Codegenerator

JMI Manipulation + Poseidon Plugin Code

Poseidon embedded MDR "Application" Models
Metamodel Class Diagram
Transformation Component

SDM GT Specifications

Stereotypes as flags
High Level SDM Control Flow
Code Activity (demo purpose only)
Loop Activity (again with stereotype)

Use tagged value as reference
Attribute Assignments  
(demo purpose only)
Compiler Results

- **JMI Repository Manipulation Code**
  - View Updating through small “hack”...

- **Poseidon Plugin**
  - Extra menu item for launching transformation
Conclusions (1/2)

1. **Now transformation writers can…**
   - Create visual model transformations in Magicdraw, Together, Objecteering, Poseidon, …
   - Execute them on JMI repositories (opt: CASE tool plugins)

2. **Now, Fujaba can…**
   - be migrated to MDR

3. **Now, other CASE tool vendors can…**
   - Provide a plugin for enhanced visualization (cfr. nesting)
Conclusions (2/2)

- Lessons Learned
  - MDA Light
    - UML Profiles: adequate if MM close to UML
    - Code Templates: reuse problems
  - Graph Rewriting
    - Convenient for specifying Refactorings
- Future Work
  - EMF support… if necessary, solve reuse problems with stepwise refinement
  - Model Refinement & Weaving Case Studies
  - Compare to (and align with) ATL

Publications