ATL:
Atlas Transformation Language

ATL Transformation Description Template
- version 0.1 -

December 2005

by
ATLAS group
LINA & INRIA
Nantes
## Content

1. Transformation Specification Sheet ................................................................. 3
2. Transformation Specification Sheet Template .................................................. 5
3. Transformation Specification Sheet Example .................................................... 6
## 1 Transformation Specification Sheet

<table>
<thead>
<tr>
<th><strong>Short Name:</strong> (&lt;t\textunderscore short_name&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short name of the transformation (e.g. UML2MSProject).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Full Name:</strong> (&lt;t\textunderscore full_name&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full name of the transformation (e.g. From UML Activity Diagram to Microsoft Project).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Short Description:</strong> (&lt;t\textunderscore short_description&gt;)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short textual description of the transformation (less than 10 lines).</td>
</tr>
</tbody>
</table>

### Source Metamodels:
- \(<mm\textunderscore name>: \(<m\textunderscore name>;\), \ldots, \(<m\textunderscore name>\>)\)
  - \(<mm\textunderscore reference>\) or \((<mm\textunderscore textual\_description>\text{and/or} <mm\textunderscore graphical\_description>)\)
  - **Pre-conditions:**
    - \(<textual\textunderscore condition\_description>\)
    - [**Specification:** \(<ocl\textunderscore condition\_specification>\)\]?

List of the transformation source metamodels. For each metamodel:
- name of the metamodel, followed by the list of source models that conform to it, followed by either a reference to the metamodel (typically a URI or a bibliographic reference), or a textual and/or a graphical representation of the metamodel;
- metamodel pre-conditions. These conditions must specifically apply to the metamodel (e.g. restricting the range of an integer attribute). For each pre-condition:
  - textual description of the condition followed by an optional OCL condition specification.

### Target Metamodels:
- \(<mm\textunderscore name>: \(<m\textunderscore name>;\), \ldots, \(<m\textunderscore name>\>)\)
  - \(<mm\textunderscore reference>\) or \((<mm\textunderscore textual\_description>\text{and/or} <mm\textunderscore graphical\_description>)\)
  - **Post-conditions:**
    - \(<textual\textunderscore condition\_description>\)
    - [**Specification:** \(<ocl\textunderscore condition\_specification>\)\]?

List of the transformation target metamodels. For each metamodel:
- name of the metamodel, followed by the list of target models that conform to it, followed by either a reference to the metamodel (typically a URI or a bibliographic reference), or a textual and/or a graphical representation of the metamodel;
- metamodel post-conditions. These conditions must specifically apply to the metamodel (e.g. restricting the range of an integer attribute). For each post-condition:
  - textual description of the condition followed by an optional OCL condition specification.

### Additional Pre-Conditions:
- \(<textual\textunderscore condition\_description>\)
  - [**Specification:** \(<ocl\textunderscore condition\_specification>\)\]?

List of the additional pre-conditions. It includes all pre-conditions applying to the source models. For each pre-condition:
- textual description of the condition followed by an optional OCL condition specification.
Additional Post-Conditions:
- `<textual_condition_description>`
  
  **[Specification: `<ocl_condition_specification>`]?**

List of the additional post-conditions. It includes all post-conditions applying to the target models. For each post-condition:
- textual description of the condition followed by an optional OCL condition specification.

**Pseudo Code: `<pseudo_code>`**

Any style of pseudo code is acceptable.
## 2 Transformation Specification Sheet Template

<table>
<thead>
<tr>
<th>Short Name:</th>
<th>&lt;t_short_name&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Name:</td>
<td>&lt;t_full_name&gt;</td>
</tr>
<tr>
<td>Short Description:</td>
<td>&lt;t_short_description&gt;</td>
</tr>
<tr>
<td><strong>Source Metamodels:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;mm_name&gt;</code>: <code>&lt;m_name&gt;</code>, …, <code>&lt;m_name&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;mm_reference&gt;</code> or (<code>&lt;mm_textual_description&gt;</code> and/or <code>&lt;mm_graphical_description&gt;</code>)</td>
<td></td>
</tr>
<tr>
<td><strong>Pre-conditions:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;textual_condition_description&gt;</code></td>
<td></td>
</tr>
<tr>
<td>[Specification: <code>&lt;ocl_condition_specification&gt;</code>]?</td>
<td></td>
</tr>
<tr>
<td><strong>Target Metamodels:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;mm_name&gt;</code>: <code>&lt;m_name&gt;</code>, …, <code>&lt;m_name&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>&lt;mm_reference&gt;</code> or (<code>&lt;mm_textual_description&gt;</code> and/or <code>&lt;mm_graphical_description&gt;</code>)</td>
<td></td>
</tr>
<tr>
<td><strong>Post-conditions:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;textual_condition_description&gt;</code></td>
<td></td>
</tr>
<tr>
<td>[Specification: <code>&lt;ocl_condition_specification&gt;</code>]?</td>
<td></td>
</tr>
<tr>
<td><strong>Additional Pre-Conditions:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;textual_condition_description&gt;</code></td>
<td></td>
</tr>
<tr>
<td>[Specification: <code>&lt;ocl_condition_specification&gt;</code>]?</td>
<td></td>
</tr>
<tr>
<td><strong>Additional Post-Conditions:</strong></td>
<td></td>
</tr>
<tr>
<td>• <code>&lt;textual_condition_description&gt;</code></td>
<td></td>
</tr>
<tr>
<td>[Specification: <code>&lt;ocl_condition_specification&gt;</code>]?</td>
<td></td>
</tr>
<tr>
<td><strong>Pseudo Code:</strong></td>
<td>&lt;pseudo_code&gt;</td>
</tr>
</tbody>
</table>
3 Transformation Specification Sheet Example

**Short Name:** UML2MSProject

**Full Name:** From UML Activity Diagram to Microsoft Project

**Short Description:** The UML2MSProject transformation generates a MS Project from a loop free UML activity diagram (describing some tasks series). The transformation is based on a simplified subset of the UML State Machine metamodel. This transformation produces a project defined in conformance to a limited subset of XML format loaded by MS Project.

**Source Metamodels:**
- UML2.0 : Um1

**Pre-conditions:**
- Considered metamodel is restricted to the Activity Diagram part of UML specification

**Target Metamodels:**
- MSProject : MsProject

```java
package MSProject {
    class MSProject {
        reference tasks[1-*] container : Task;
    }

    abstract class NamedElement {
        attribute name : String;
    }

    class Task extends NamedElement {
        attribute UID : String;
        reference predecessors[*] : Task;
    }
}
```

**Post-conditions:** Empty

**Additional Pre-Conditions:**
- The source model Uml must be loop-free

**Additional Post-Conditions:**
- Task identifiers (UID) of the target model MsProject must be unique

**Specification:**
- context MSProject!Task:
  - not MSProject!Task.allInstances() ->exists(e | e.uid = self.uid and e <> self)

**Pseudo Code:**
- -- Rule 'Main'
- -- This rule generates the Project element. Contained tasks are those
- -- associated with:
- -- * UML Final State
- -- * UML Action State
- -- * UML Pseudostate of "initial" kind.

- -- Rule 'Pseudostate'
- -- This rule generates a Task for the Pseudostate of "initial" type (that is,
-- the diagram initial state).
-- The generated initial Task has no predecessors (since it corresponds to the
-- initial state of the UML activity diagram).

-- Rule 'StateVertex'
-- This rule generates Tasks for both ActionStates and FinalStates.
-- The set of predecessors of a Task is computed by the getPredecessors helper.
-- It corresponds to the set of ActionState/"initial" Pseudostate pointing to
-- the current state directly, or through one or several "fork" and "join"
-- Pseudostates.