ATL:
Atlas Transformation Language

ATL Transformation Description Template
- version 0.1 -

December 2005

by

ATLAS group
LINA & INRIA
Nantes

Content

1 Transformation Specification Sheet

Short Name: <*t_short_name*>
Short name of the transformation (e.g. UML2MSProject).
Full Name: <*t_full_name*>
Full name of the transformation (e.g. From UML Activity Diagram to Microsoft Project).
Short Description: <*t_short_description*>
Short textual description of the transformation (less than 10 lines).
Source Metamodels: • <*mm_name*>**:** <*m_name*$_1$>, …, <*m_name*$_n$> <*mm_reference*> or (<*mm_textual_description*> and/or <*mm_graphical_description*>) **Pre-conditions:** o <*textual_condition_description*> [**Specification:** <*ocl_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: o textual description of the condition followed by an optional OCL condition specification.
Target Metamodels: • <*mm_name*>**:** <*m_name*$_1$>, …, <*m_name*$_n$> <*mm_reference*> or (<*mm_textual_description*> and/or <*mm_graphical_description*>) **Post-conditions:** o <*textual_condition_description*> [**Specification:** <*ocl_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: o textual description of the condition followed by an optional OCL condition specification.
Additional Pre-Conditions: • <*textual_condition_description*> [**Specification:** <*ocl_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

Short Name: <*t_short_name*>
Full Name: <*t_full_name*>
Short Description: <*t_short_description*>
Source Metamodels: • <*mm_name*>**:** <m_name_1>, …, <m_name_n> <*mm_reference*> or (<*mm_textual_description*> and/or <*mm_graphical_description*>) **Pre-conditions:** o <*textual_condition_description*> [**Specification:** <*ocl_condition_specification*>]?
Target Metamodels: • <*mm_name*>**:** <m_name_1>, …, <m_name_n> <*mm_reference*> or (<*mm_textual_description*> and/or <*mm_graphical_description*>) **Post-conditions:** o <*textual_condition_description*> [**Specification:** <*ocl_condition_specification*>]?
Additional Pre-Conditions: • <*textual_condition_description*> [**Specification:** <*ocl_condition_specification*>]?
Additional Post-Conditions: • <*textual_condition_description*> [**Specification:** <*ocl_condition_specification*>]?
Pseudo Code: <*pseudo_code*>

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 : Uml

http://www.omg.org/technology/documents/formal/uml.htm

Pre-conditions:

- Considered metamodel is restricted to the Activity Diagram part of UML specification

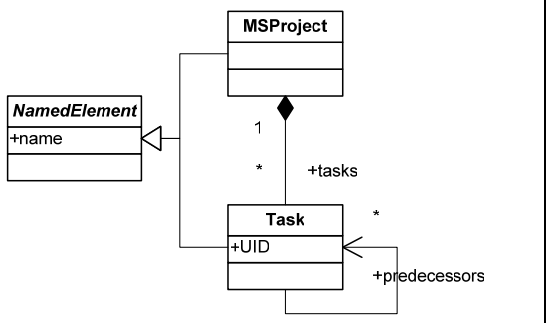
Target Metamodels:

- MSProject : MsProject

```
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 (sine it corresponds to the
-- intial 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.
```