



ATL 2.0 Ganymede Simultaneous Release

ATL PMC

Overview



- Introduction
- Features
- Non-code Aspects
- Testing and Packaging
- Community & Support
- Intellectual Properties
- Project Plan

Introduction



- ATL : ATLAS Transformation Language
- ATL is a language and a Virtual Machine dedicated to model transformation
- ATL is an Eclipse Model-to-Model (M2M) component, inside of the Eclipse Modeling Project (EMP)
- ATL has been moved from GMT to M2M in 2007



Features (core)



- A syntax adapted to Model To Model transformation
 - Hybrid (Declarative and Imperative)
 - Model navigation using OCL
- A Virtual Machine
 - Executes ATL transformations pre-compiled into low level transformation-specific bytecode
 - Provides execution environment for any transformation language
 - The M2M QVT Relational project is based on the ATL Virtual Machine
 - [http://wiki.eclipse.org/M2M/Relational_QVT_Language_\(QVTR\)](http://wiki.eclipse.org/M2M/Relational_QVT_Language_(QVTR))
 - A use case implements a QVT Operational Mappings compiler
 - <http://www.eclipse.org/m2m/atl/usecases/QVT2ATLVM/>

Features (UI)



- An Eclipse based IDE
 - Project nature and builder
 - Perspective, wizards
 - A Launch Configuration type
 - A textual Editor
- ATL editor
 - Syntax color
 - Code assist
 - Outline

```
rule Model2Database {  
  from  
    m : UML!Model (  
      m.hasStereotype('Database')  
    )  
  to  
    out : Relational!Database (  
      name <- m.name,  
      ownedSchemas <- m.packagedElement  
        | comment  
        | ownedSchemas  
    ),  
    inte | url
```

Non-code aspects



- Wiki-based FAQ, User Guide, and Tips & Tricks
- Web site:
 - ATL Transformation Zoo
 - 100+ scenarios, with contributions from the community
 - Complete use cases (20+, with contributions from the community)
 - Articles
- On the wiki
 - Advanced user documentation
 - Developer documentation
- Help plugin, containing Javadoc

Testing and Packaging



- ATL uses the Modeling Project Releng system to build and promote versions, which is also used by: EMF, EMFT, M2T, and MDT components
- Each new build is tested with Eclipse 3.4, 3.3, 3.2
- ATL is integrated into the Ganymede update site since January 2008
- A non regression test suite checks that the engine still executes correctly existing transformations
 - Reuse of ATL Transformation Zoo
 - Resulting models are compared using EMF Compare
 - Non regression evaluated for
 - Parsing
 - Compilation
 - Execution
- A set of benchmarks also checks ATL Virtual Machine performances

Community and support



- Newsgroup : very active community, more than 2000 posts since its creation
- EclipseCon 2008 : tutorial
<http://www.eclipsecon.org/2008/index.php?page=sub/&id=402>
- Publications about ATL :
<http://www.eclipse.org/m2m/atl/publication.php>
- Bugzilla :

		Status					
		NEW	ASSIGNED	REOPENED	RESOLVED	CLOSED	Total
Severity	critical	.	.	.	<u>3</u>	.	<u>3</u>
	normal	<u>12</u>	<u>1</u>	<u>1</u>	<u>31</u>	<u>4</u>	<u>49</u>
	minor	.	.	.	<u>2</u>	.	<u>2</u>
	enhancement	<u>1</u>	.	.	<u>7</u>	.	<u>8</u>
	Total	<u>13</u>	<u>1</u>	<u>1</u>	<u>43</u>	<u>4</u>	<u>62</u>

IP Issues



- All plugins contain appropriate about and license files
- IP process followed
- A third-party library is used : antlr 3.0
 - IPzilla CQ 1548
 - Use of the matching ORBIT library for build
- IP Log available at
<http://www.eclipse.org/modeling/m2m/atl/eclipse-project-ip-log.csv>
- Released under EPL

Project Plan



- Planned improvements:
 - Performance
 - Specific language constructs for “in-place” transformations (prototypes already working)
 - Compile-time optimizations
 - Architecture
 - Tests coverage
 - Clarify API for users