Tutorial of the ATL transformation language
http://github.com/jesusc/atl-tutorial

Part IV

INTEGRATION OF ATL TRANSFORMATIONS

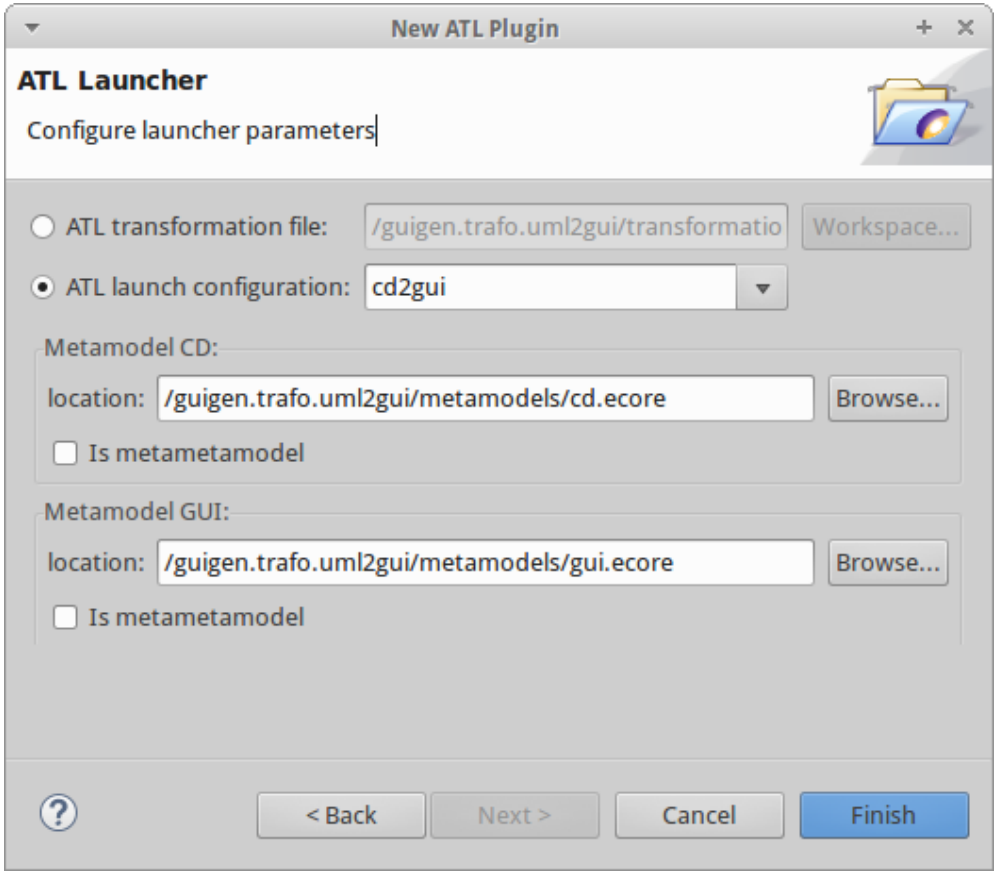
jesus.sanchez.cuadrado@gmail.com
@sanchezcuadrado
http://sanchezcuadrado.es

Motivation

- ATL transformations are created “at development time”
- We need to integrate them in actual modelling tools, notably for Eclipse
  - We cannot use launch configurations
  - We need to develop our own launching code

ATL Plug-in generator

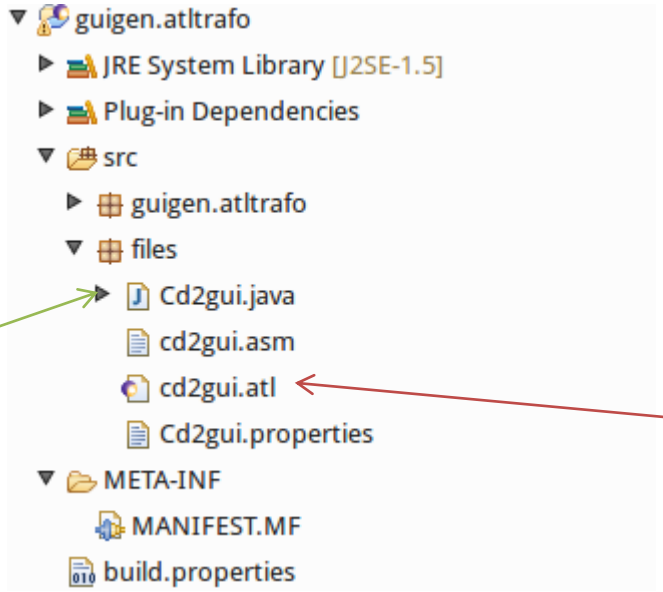
- File -> Project ... -> ATL -> ATL Plug-in



ATL Plug-in generator

- Generates an Eclipse plug-in project

Cd2Gui
loadModels(in : String) doCd2gui(monitor) saveModels(out : String)



A copy of the original transformation.
Be careful!

ATL Plug-in generator

```
# ATL modules: if several, by order of superimposition (the latter ones
overrides the former ones)
Cd2gui.modules = cd2gui.atl

# Metamodels paths or nsUris
Cd2gui.metamodels.CD = /guigen.trafo.uml2gui/metamodels/cd.ecore
Cd2gui.metamodels.GUI = /guigen.trafo.uml2gui/metamodels/gui.ecore

# Libraries paths
Cd2gui.libraries.GUIlib = /guigen.trafo.uml2gui/lib/GUILib.asm

# ATL Launching options
Cd2gui.options.supportUML2Stereotypes = false
Cd2gui.options.OPTION_CONTENT_TYPE = false
Cd2gui.options.step = false
Cd2gui.options.allowInterModelReferences = false
Cd2gui.options.printExecutionTime = true
```

ATL Plug-in generator

- My advice…
  - Don’t use it as is, just as a reference to implement your own launching code.

Resource references

Resource Type	ATL EMF API Syntax	Example
File system Resource	file:/<path>	file:/D:/eclipse/workspace/mmproject/sample_metamodel.ecore
EMF uri	<uri>	**http://www.eclipse.org/uml2/2.1.0/UML**
pathmap	pathmap:<path>	pathmap://PROFILE/sample_profile.uml#_0
Workspace Resource	platform:/resource/<path>	platform:/resource/mmproject/sample_metamodel.ecore
Plug-in Resource	platform:/plugin/<path>	platform:/plugin/mmproject/sample_metamodel.ecore

Warning, in Eclipse:

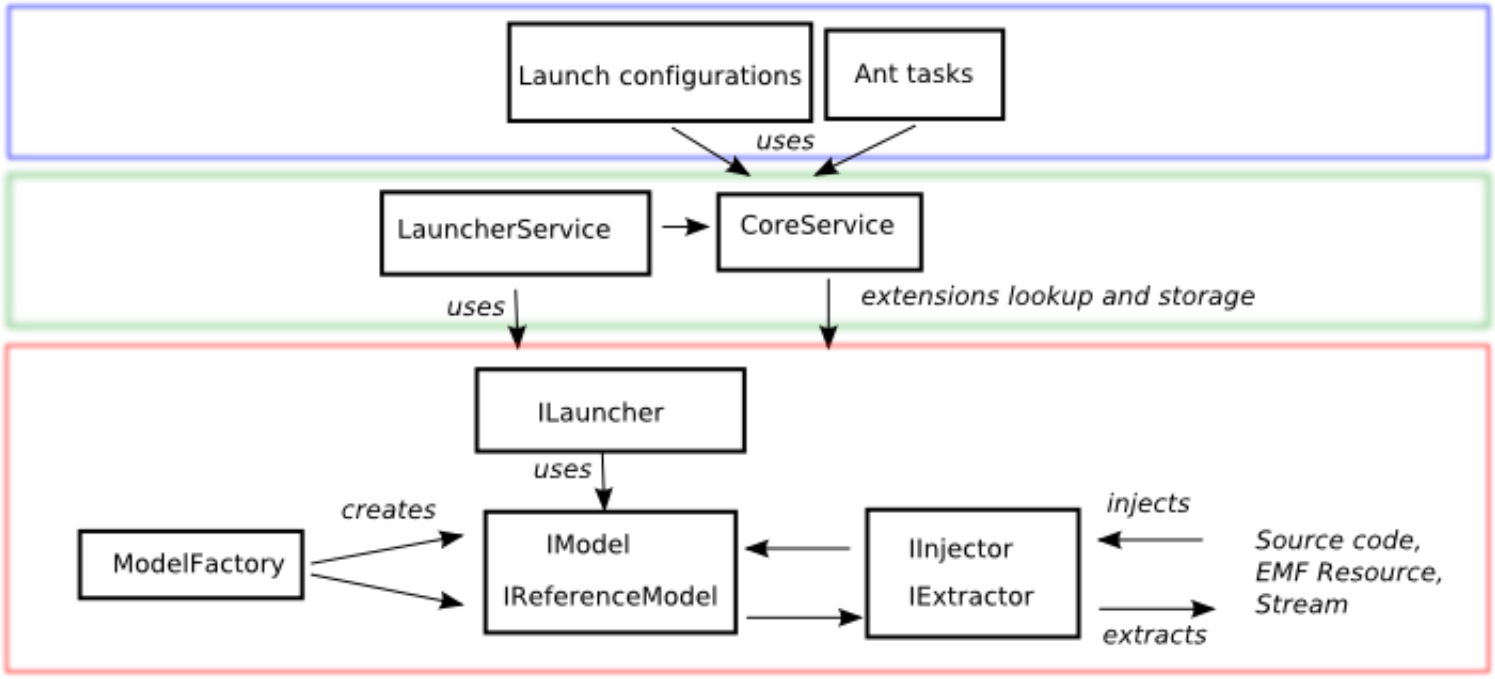
IFile.getFullPath() -> Workspace-relative path
IFle.getLocation() -> Filesystem absolute path

* From: https://wiki.eclipse.org/ATL/Developer_Guide#EMF_interactions
* See also: http://lmap.blogspot.com.es/2008/03/platform-scheme-uri.html

Resource references

- Pay particular attention to cross-model references
  - Either use workspace-relative paths
  - Or use absolute paths (IFile.getLocation())
    - getFullPath() is a false friend!

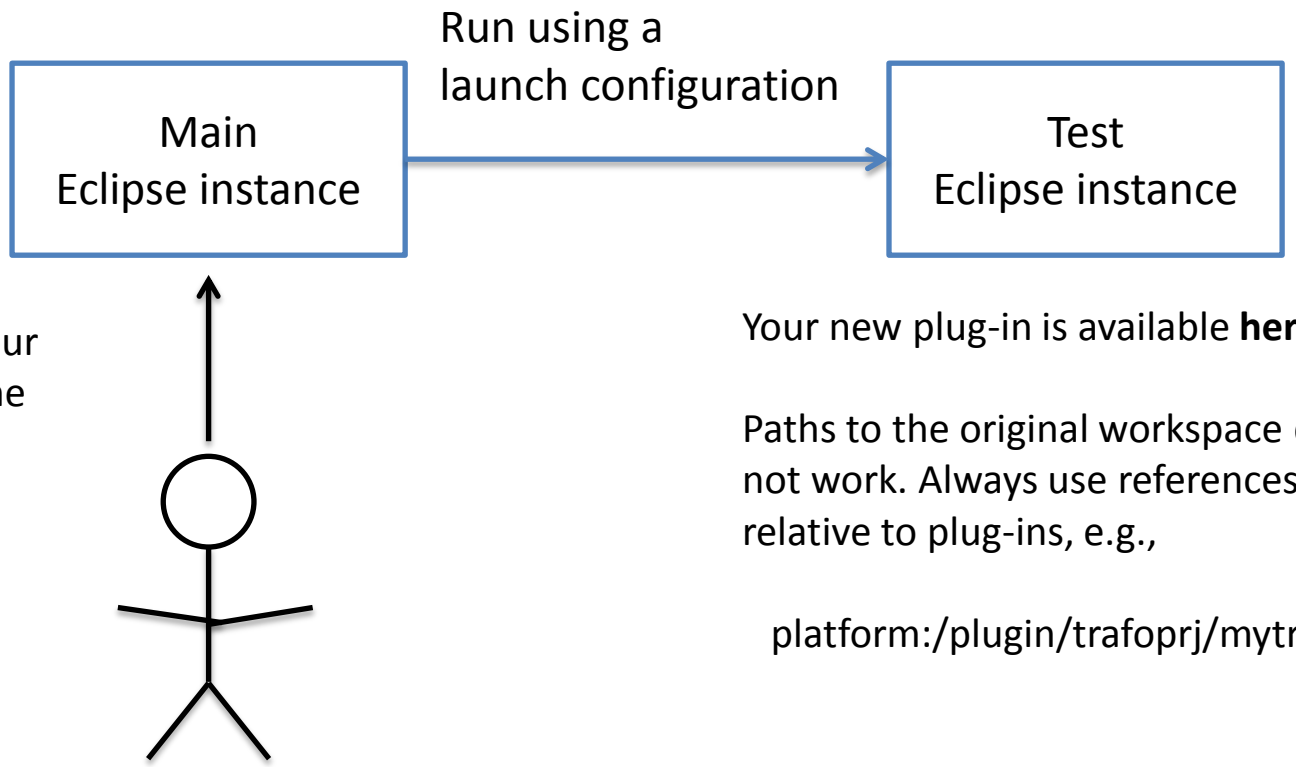
Launching architecture



In practice, one typically knows which subclass of IModel to use

* From: https://wiki.eclipse.org/ATL/Developer_Guide

Eclipse



Your are developing your transformations and the integration code **here**

Your new plug-in is available **here**.

Paths to the original workspace does not work. Always use references relative to plug-ins, e.g.,

platform:/plugin/trafoprj/mytrafo.atl

Eclipse

- Fighting Eclipse
  - SWT Spy
  - Alt+Shift+F1
    - Show information about the selected UI component
  - Alt+Shift+F2
    - Press this shortcut, and then select a menu to obtain information

* http://www.vogella.com/tutorials/EclipseCodeAccess/article.html#swt-spy

Eclipse

- Typically required plug-ins
  - org.eclipse.ui,
  - org.eclipse.core.runtime,
  - org.eclipse.core.resources
  - org.eclipse.emf
  - org.eclipse.emf.ecore.xmi
  - org.eclipse.uml2
  - org.eclipse.uml2.uml
  - org.eclipse.m2m.atl.core,
  - org.eclipse.m2m.atl.core.emf
  - org.eclipse.m2m.atl.engine.emfvm.launch
  - org.eclipse.m2m.atl.emftvm

Demo

- Create a chain of transformations with Java
- Integration in Eclipse
  - Pop-up menu over UML models
  - Eclipse Job to transform in background
- Two different ATL VMs working together

- Mini-exercise
  - Implement cancellation for the Eclipse job