Storyboard of the Demo

Goal of the Demo: To extend Darwin with Behavior and Software Connectors. The system to be modeled is an Adaptive Cruise Control (ACC) system.

Why?

An example shows ACC modeled in Darwin, highlighting "missing aspects". See Screenshots 1, 2, and 3.

Main ingredients:

See Screenshot 4

- Darwin Metamodel
- Behavior Metamodel
- Connectors Metamodel

These metamodels are stored in a tagged metamodels Repository

Metamodels composition (DarwinBehavior):

- Definition of the weaving model linking the Darwin and Behavior metamodel. See Screenshot 5
- Generation of the composed metamodel. See Screenshot 5
- Editors generation. See Screenshot 1
  - Textual editor
  - Tree-based editor
  - Graphical editor (thanks to Eugenia annotations)

Once the new ADL has been built the ACC system can be modeled:

- Creating a new model from scratch. See Screenshot 6
- How to reuse a system already modeled in Darwin?
  - By using the (automatically generated) single-to-composed migrator See Screenshot 7
- Now it is possible to use the new ADL to make changes on the model See Screenshot 8
- The composed-to-single migrator allows the architect to come back to Darwin

A similar composition is performed for adding connectors to the newly built ADL.

In the following we show the screenshots representing the activities listed above, together with an explaination on how to reproduce the demo

First of all, this demo is composed of three main parts:

1. **Installation of the byADL tool for Windows and the files related to the Demo.** An Eclipse distribution containing all the needed plugins can be downloaded by following this link:

   http://byadl.di.univaq.it/downloads.html

   Source files, and the a platform independent plugin for Eclipse are also available. To install byADL, just download the byadl0.0.2WindowsBundle.zip archive and extract its contents. To run byADL, just double-click on byADL.exe.
2. **Generation of a new (extended and customized) ADL.** The files related to this part of the Demo are contained in the demoECSA2010.zip archive. Import them as already-existing-projects into byADL (they are standard Eclipse projects) in order to use them in byADL.
3. **Utilization of the new ADL.** The files related to this part of the Demo are contained in the demoECSA2010_models.zip archive. They will be used only once the languages editors are generated, and will be imported to a new instance of Eclipse after our new language has been generated.

It is preferable that the user is already familiar with model driven engineering concepts and tools (for example EMF, GMF/GMP, ATL, and Eugenia).

For further details on these technologies, please refer to

http://www.eclipse.org/modeling/emf/

http://www.eclipse.org/modeling/gmp/

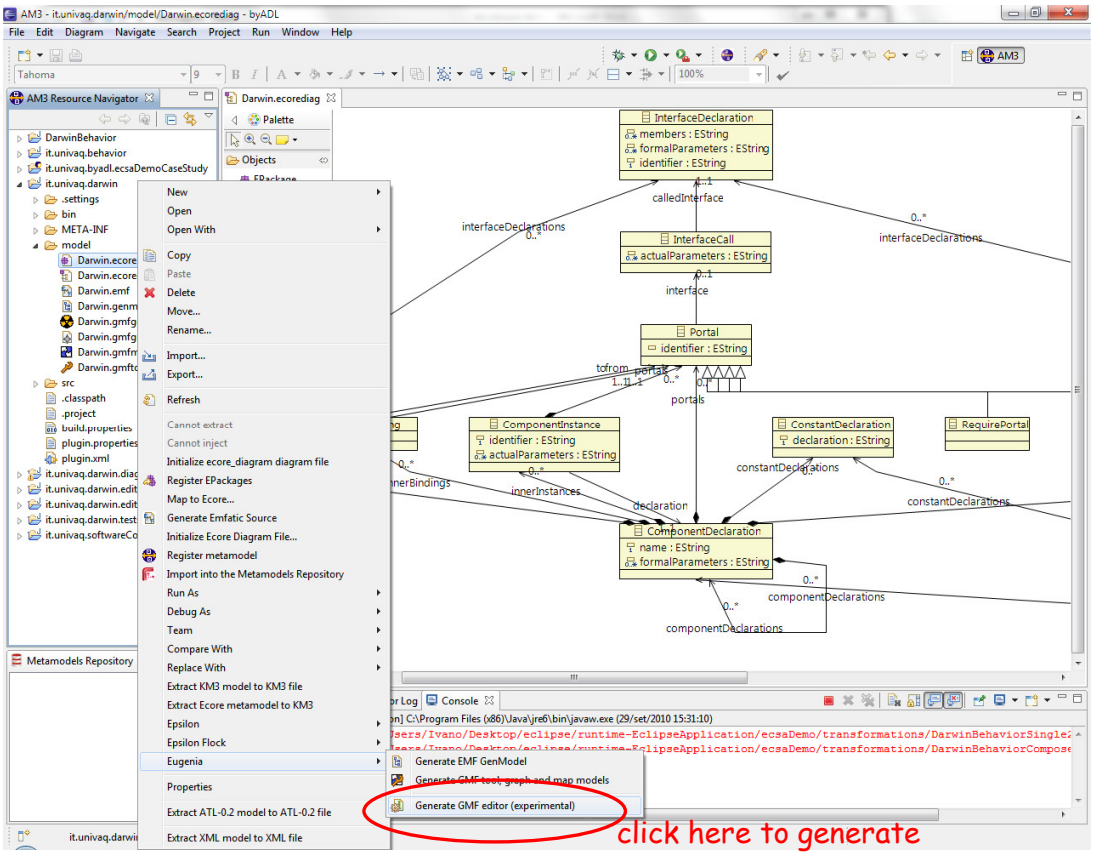
http://www.eclipse.org/atl/

http://www.eclipse.org/gmt/epsilon/doc/articles/eugenia-gmf-tutorial/

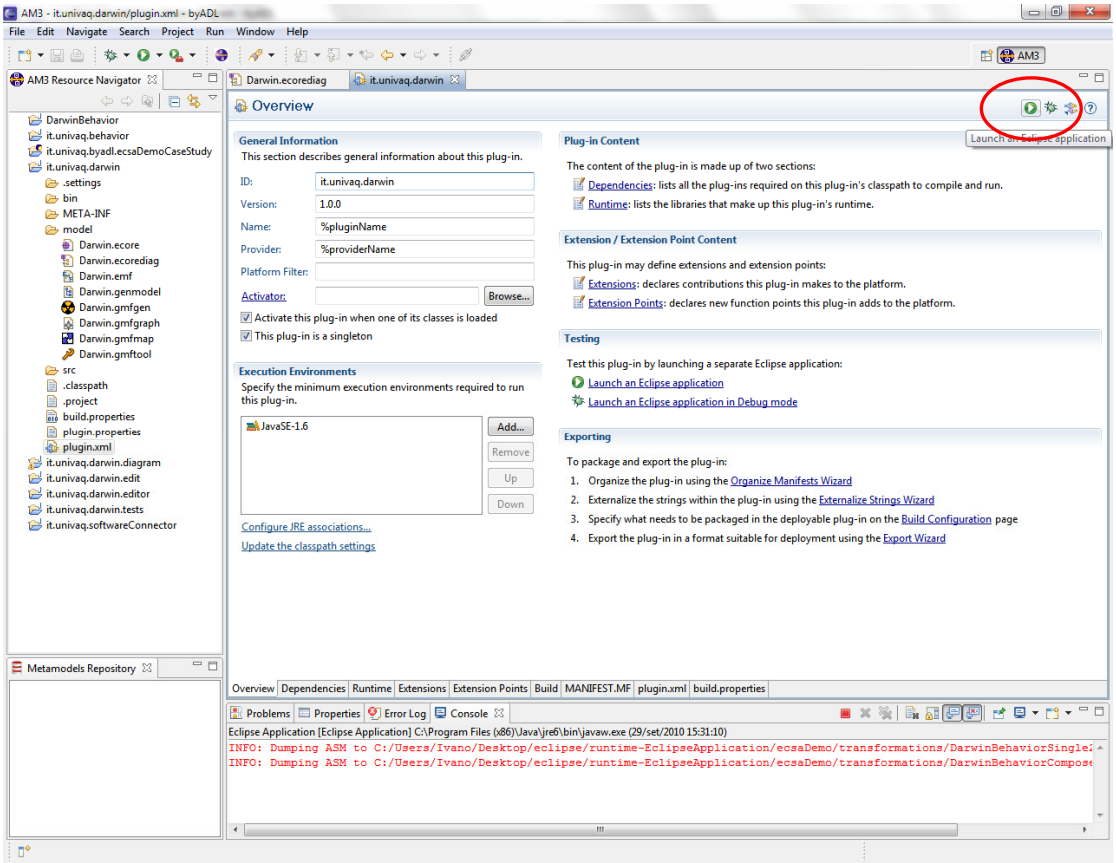
The aim of this document is to guide a Demo presented at the European Conference on Software Architecture (ECSA) 2010. This document is not a full tutorial of byADL.

Reference contacts: Ivano Malavolta, ivano.malavolta@univaq.it

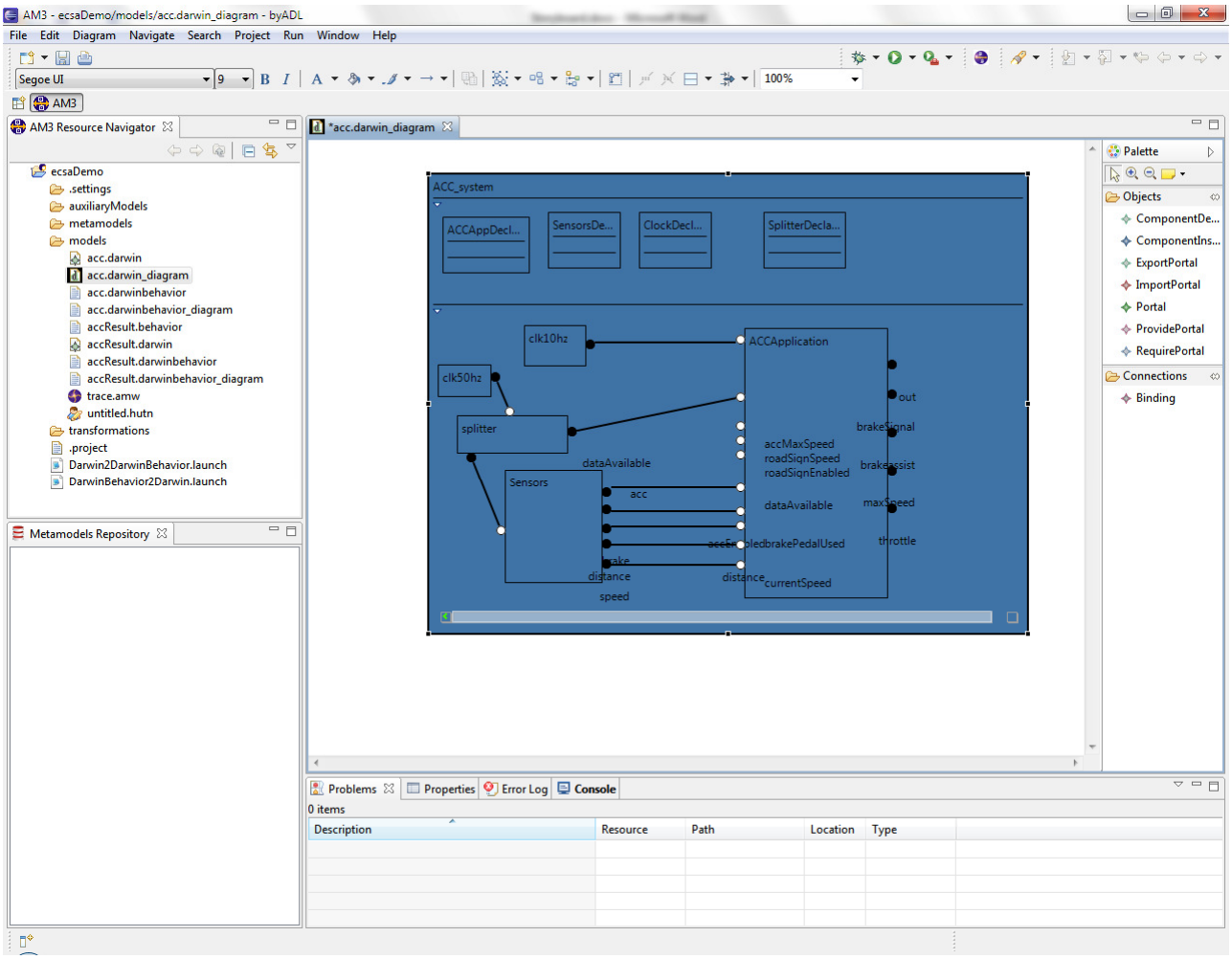
Patrizio Pelliccione, patrizio.pelliccione@univaq.it



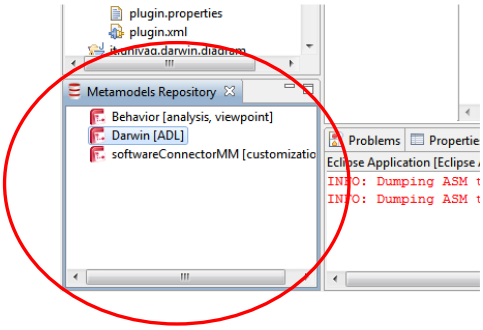
Screenshot 1. Automatic generation of the Darwin graphical editor



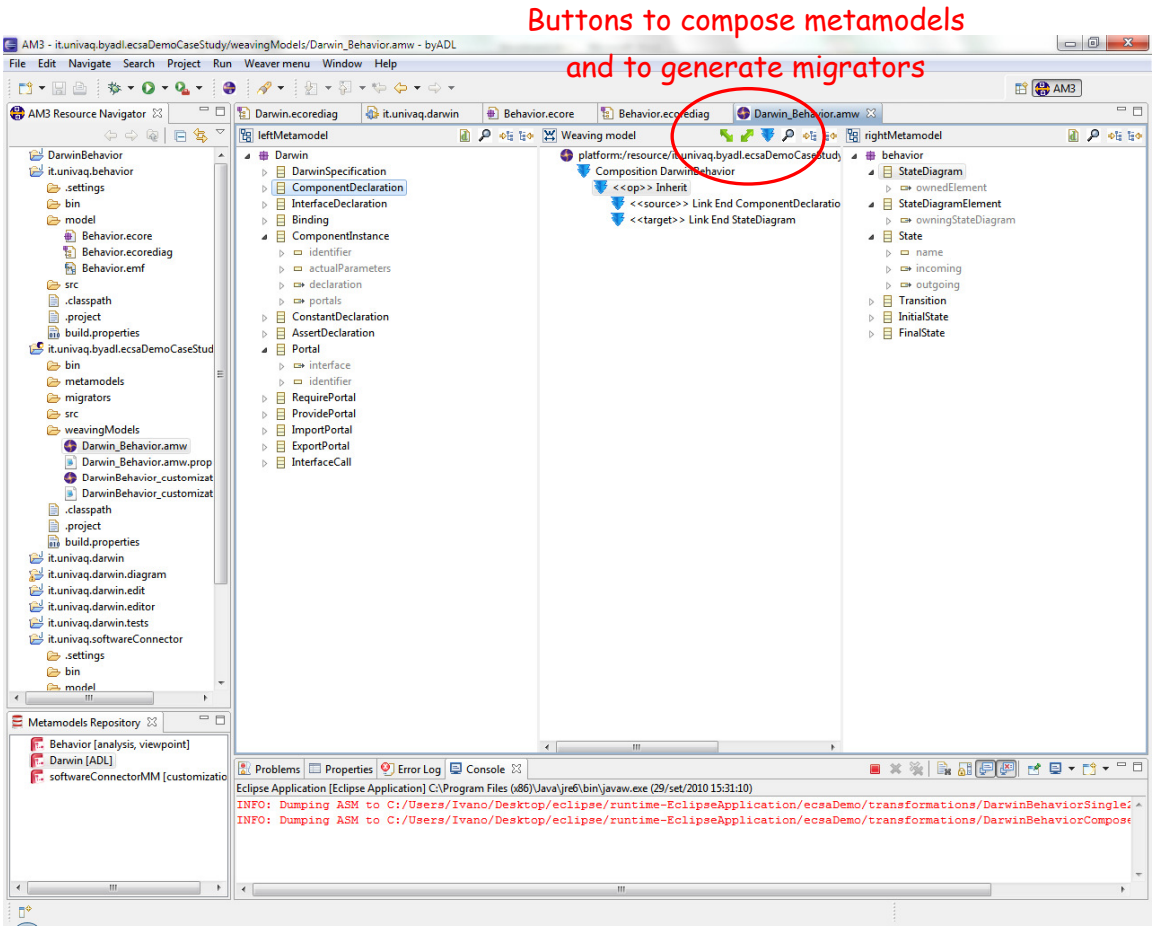
Screenshot 2. Launching a new Eclipse instance supporting the newly generated Darwin graphical editor



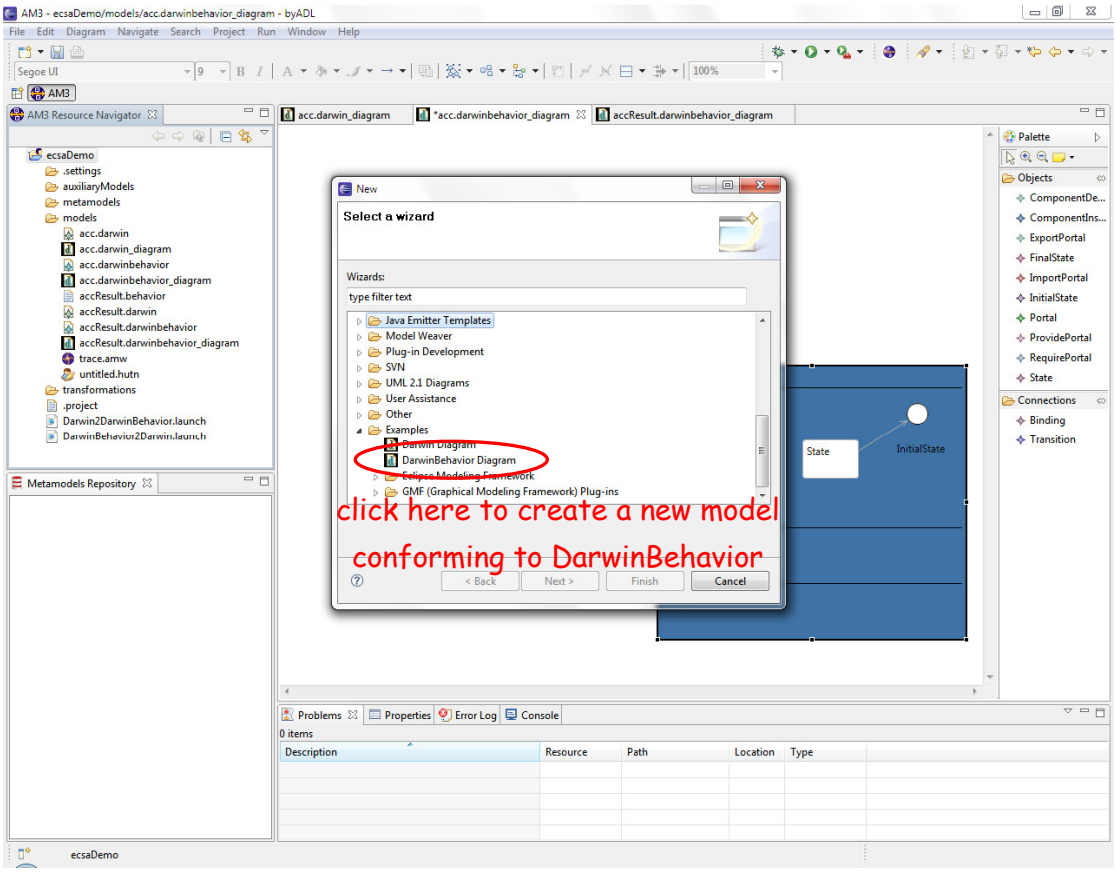
Screenshot 3. Adaptive Cruise Control modelled through the generated Darwin graphical editor (the elements listed on the left have been imported from demoECSA2010_models.zip)



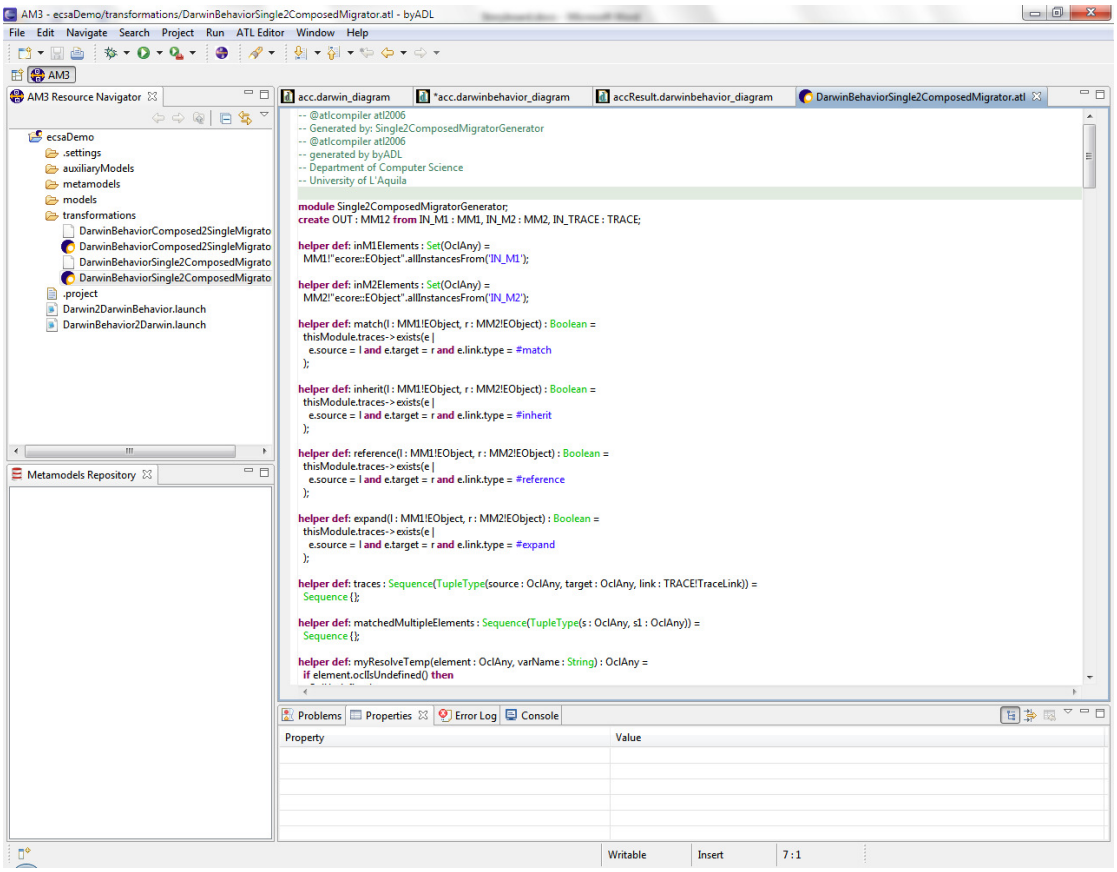
Screenshot 4. Part of the byADL tool showing the Metamodels Repository (in order to see this specific repository you have to copy the byadl.db file from demoECSA2010.zip to your Eclipse workspace folder)



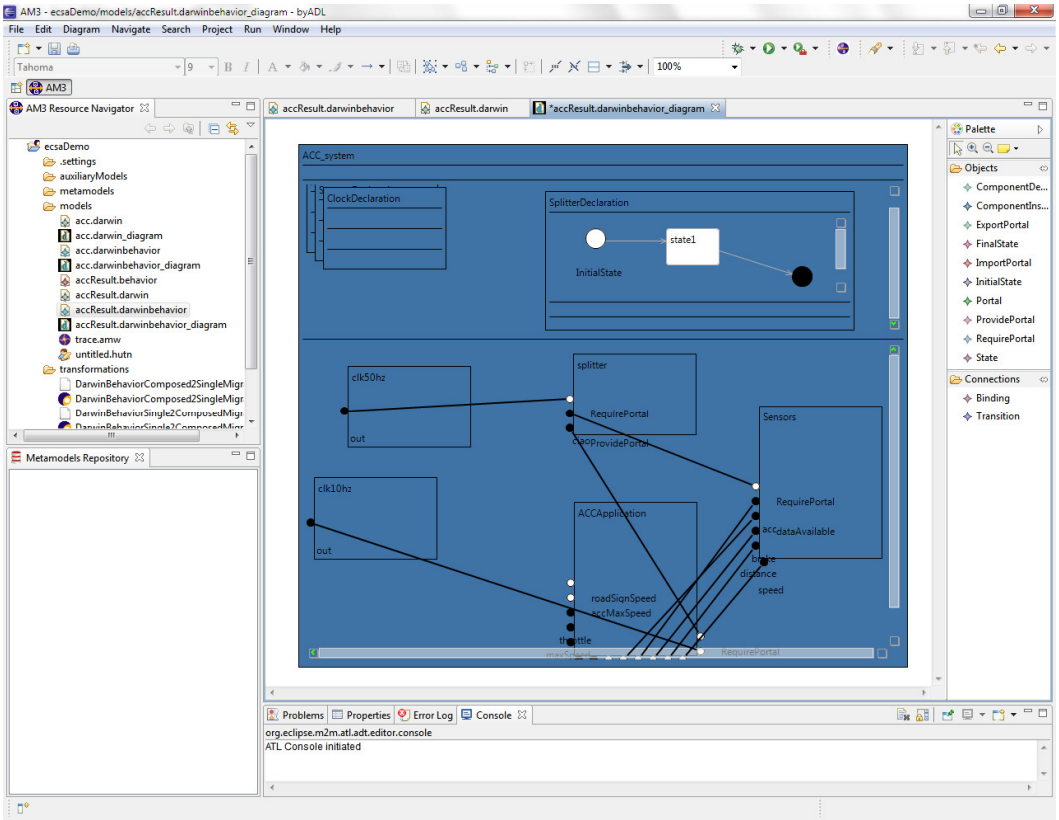
Screenshot 5. Weaving model for composing metamodels



Screenshot 6. Creation of a new model conforming to the newly created DarwinBehavior language



Screenshot 7. The automatically generated single-to-composed migrator (in ATL)



Screenshot 8. The DarwinBehavior model obtained from the execution of the migrator, and modified to add a state machine