

CS 493/693/793: Special Topics in Eclipse Tools for Model Engineering

Fall 2007

Time: 3:30pm – 4:45pm Tuesday and Thursday
Classroom: CH 145 (Tuesday) & CH 132 (Thursday)

Instructor: Dr. Frédéric Jouault
Office: CH 122
Phone: (205)-934-8640
Email: jouault (at) cis.uab.edu
Web: <http://www.sciences.univ-nantes.fr/lina/atl/contrib/jouault>

TA: TBA

Course Materials: <http://www.cis.uab.edu/cs693/fall2007/>
Lectures and other materials archived at Y:\cs693

Office Hours: By appointment

Course Focus

Classical approaches to software development often make use of abstract representations (called models) of the system to build. These models are typically expressed either textually (e.g., pseudo-code) or visually (e.g., class diagrams). Software developers are responsible for translating them into actual source code. Because they are only intended to be interpreted by human beings these models are called *contemplative models*.

Model Engineering is a general term used to refer to a set of new approaches considering models (and no longer source code) as the main artifacts of the software development process. The models are precisely represented so as to be automatically processed, via model transformation, to generate other artifacts such as other models or source code. Among these approaches, we can cite:

- MDA™ (Model Driven Architecture) is supported by the OMG (Object Management Group) and relies on a set of OMG recommendations (MOF, UML, OCL, QVT, XMI, etc.)
- MIC (Model Integrated Computing) has been developed at ISIS, Vanderbilt University
- DSL Tools are developed by Microsoft to build Domain-Specific Languages (DSLs) and are based on Model Engineering principles

The first part of this course will consist of the presentation of the key concepts and principles of Model Engineering, as well as of Eclipse tools implementing them. Then, in the second part of this course, you will have to work on a Model Engineering project using the tools presented in the first part of the course.

Examples of Model Engineering projects are available on the websites: *ATL use cases*, and *ATL transformations*. The addresses of these websites are given in the **Related Resources** section at the end of this syllabus.

Prerequisites

You should be comfortable writing code in at least one object-oriented language (e.g., Java, C#, or C++).

Textbook

There is no formal textbook for this course. The course web site will contain a list of required readings, as well as other related papers/books.

Grade Determination

If you have not attended CS 493/693/793 in Spring 2007, your final grade, will be determined by the following items:

Project:	70%	Midterm Exam:	30%
----------	-----	---------------	-----

If you have attended CS 493/693/793 in Spring 2007, your final grade will be determined by the following items:

Project:	100%
----------	------

There will be NO makeup exams. You will receive a zero for missing any exam without a valid excuse. Please note: **This especially includes situations where a student wants to leave for home before exams and project presentations are over – this is not acceptable!** Please do not make travel arrangements that will force you to miss an exam (this seems to happen every semester – this is an advanced warning – you will get a 0 on your final if that happens).

Honor Code and General Etiquette

Each student is to do his or her own work. This means that you are not to seek out the help of other students (or give help, if asked) in order to solve specific problems of your homework assignments. It also means that you should not sign up for mailing lists and ask for detailed help from others on the net. Of course, you may discuss generalities about an assignment with your fellow students. If you are unsure of what is permitted, in terms of discussing an assignment problem, please ask me for clarification. If you are caught in violation of the honor code, you will receive a zero for the assignment with the potential for a note being added to your academic file. If you violate the honor code

twice, you will receive a zero for the course and be reported to the Graduate Faculty director.

Please turn off all cell phones and pagers while you are in lecture. Do not surf the web or write email during the lecture.

Please arrive on time so that you do not cause a disruption in the middle of class.

Disabilities

If you have any disability that would put you at a disadvantage in performing an assignment, or in taking an exam, please meet with me privately to discuss ways in which I can assist you as you perform the required work in this course.

Tentative Schedule

Date	Topic	Readings (TBA)	Remarks (TBA)
<u>First part</u>			
August 23	Course introduction		
August 28	Basic principles of Model Engineering		
August 30	Overview of Eclipse Tools for Model Engineering		
September 4	Metamodeling with KM3		
September 6			
September 11	Textual syntax specification with TCS		
September 13			
September 18	Transforming models with ATL		
September 20			
September 27	Midterm exam		
<u>Second part</u>			
TBA	Project progress reports		

Suggested Course Readings

TBA

Other Books/Papers of Interest

TBA

Related Resources

The following is just an initial sample of related resources. Please check the course web site frequently for updates to this list.

Model Engineering Approaches

- **OMG** (<http://www.omg.org/>), **Catalog of OMG Modeling and Metadata Specifications**, available at:
http://www.omg.org/technology/documents/modeling_spec_catalog.htm
- **Eclipse modeling resources** (see dedicated section below)
- **Microsoft, Domain-Specific Language Tools** website, available at:
<http://msdn.microsoft.com/vstudio/DSLTools/>
- **ISIS, Vanderbilt University, The Generic Modeling Environment** website, available at: <http://www.isis.vanderbilt.edu/projects/gme/>
- **MetaCase, Domain-Specific Modeling with MetaEdit+** website, available at:
<http://www.metacase.com/>

Eclipse (<http://www.eclipse.org/>) modeling resources

- **Eclipse/Modeling** (<http://www.eclipse.org/modeling/>) website:
 - Abstract Syntax Development with **EMF** (<http://www.eclipse.org/emf/>)
 - Concrete Syntax Development with **GMF** (<http://www.eclipse.org/gmf/>)
 - **Model-to-model transformation (M2M)**
(<http://www.eclipse.org/proposals/m2m/>):
 - **ATL project page** (<http://www.eclipse.org/m2m/atl/>)
 - **ATL use cases** (<http://www.eclipse.org/m2m/atl/usecases/>)
 - **ATL transformations**
(<http://www.eclipse.org/m2m/atl/atlTransformations/>)
 - **Model-to-text (M2T)** project proposal, available at:
<http://www.eclipse.org/proposals/m2t/>
 - Technology and Research with **GMT** (<http://www.eclipse.org/gmt/>):
 - **openArchitectureWare** (oAW) project page
(<http://www.eclipse.org/gmt/oaw>)

ModelWare (<http://www.modelware-ist.org/>) videos

- Jean Bézivin, **Introduction to Model Engineering**, available at:
http://www.modelware-ist.org/index.php?option=com_wrapper&Itemid=132
- Ivan Kurtev, **Presentation of QVT**, available at:
http://www.modelware-ist.org/index.php?option=com_wrapper&Itemid=164
- Frédéric Jouault, **Presentation of ATL**, available at:
http://www.modelware-ist.org/index.php?option=com_wrapper&Itemid=163
- Ivan Kurtev, **Alignment of ATL and QVT**, available at:
http://www.modelware-ist.org/index.php?option=com_wrapper&Itemid=165

Miscellaneous

- **Sun, Java Metadata Interface (JMI), JSR40**, available at:
<http://java.sun.com/products/jmi/>
- **Engineering of Grammarware**, available at:
<http://www.cs.vu.nl/grammarware/>