

# CS 493/693/793: Special Topics in Model Engineering and Model Transformation

Spring 2007

Time: 4:00pm – 5:15pm Monday and Wednesday  
Classroom: CH 145

Instructor: Dr. Frédéric Jouault  
Office: CH 122  
Phone: (205)-934-2213  
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/spring2007/>  
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

This course will present the key concepts and principles of Model Engineering and how it can be used to develop software. The general topics of the course will be:

- Basic principles of Model Engineering
- Comparison of Model Engineering to other technologies (e.g., XML, Grammarware)
- Tool support for Model Engineering
- Metamodeling (with MOF, Ecore, MetaGME, KM3, DSL Tools, etc.)
- Model transformation (with QVT, ATL, GReAT, graph-based approaches, etc.)
- Bridging Model Engineering with other technologies (including source code generation)
- Model Engineering and Domain-Specific Languages

## 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

Your final grade for the CS 493 course will be determined by the following items:

Quizzes:	20%	Homework:	30%
Midterm Exam:	20%	Final Exam:	30% (Cumulative)

Your final grade for the CS 693 and CS 793 courses will be determined by the following items:

Quizzes:	15%	Homework:	15%
Midterm Exam:	20%	Final Exam:	30% (Cumulative)
Project:	20%		

There will be a total of 10 quizzes. You will be able to drop the lowest score (that is, only the top 9 scores will count toward your overall quiz grade). There will be NO makeup quizzes, *unless you give me over a weeks advanced notice* with a good reason (e.g., out of town for a conference). Requests for make-up quizzes, after the fact, will be denied and you will receive a zero for missing a quiz without a valid excuse.

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 finals 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).

Homework is due at the beginning of class. You will receive 10 points deduction for each day that your homework is late.

## 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

<b>Date</b>	<b>Topic</b>	<b>Readings (TBA)</b>	<b>Remarks (TBA)</b>
January 8	Course introduction		
January 10	Basic principles of Model Engineering		
January 17	Comparison of Model Engineering to other technologies (XML, Grammarware, Object, DSLs, ontology, etc.)		
<b><u>Standards and Tools for Model Engineering Section</u></b>			
January 22	MDA™ (MOF, XMI, QVT, UML, etc.)		
January 24			
January 29	Eclipse/Modeling (EMF, GMF, AMMA, oAW, etc.)		
January 31			
February 5	MIC and GME		

February 7	DSL Tools		
February 12	MetaEdit+		
<b><u>Metamodeling Section</u></b>			
February 14	Introduction to metamodeling		
February 19	MOF, Ecore, and KM3		
February 21	MetaGME		
February 26	With MetaEdit+ and DSL Tools		
February 28	Metamodel design		
<b><u>Model Transformation Section</u></b>			
March 5	Introduction to model transformation		
March 7	QVT		
March 19	ATL		
March 21	GReAT		
March 26	Other approaches		
March 28			
<b><u>Bridging with other technologies Section</u></b>			
April 2	Source code generation (e.g. template-based approaches)		
April 4			
April 9	Technical spaces and projection		
<b><u>Model Engineering in practice Section</u></b>			
April 11	Application of Model Engineering to various application domains (including projects presentation)		
April 16			
April 18			
April 23			
April 25	Last day of class: wrap-up		
April 30	Final exam		4:15pm – 6:45pm

## **Suggested Course Readings**

TBA

## **Other Books/Papers of Interest**

TBA

## Other 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](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/Ecasis (<http://www.eclipse.org/ecasis/>) course, **Model Driven Development**, available at:  
<http://dev.eclipse.org/viewcvs/indextech.cgi/%7Echeckout%7E/ecasis-home/downloads/MDD-v1.0.0.zip>
- **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) project proposal, available at:  
<http://www.eclipse.org/proposals/m2m/>
  - **Model-to-text** (M2T) project proposal, available at:  
<http://www.eclipse.org/proposals/m2t/>
  - Technology and Research with **GMT** (<http://www.eclipse.org/gmt/>):
    - **ATL** project page (<http://eclipse.org/gmt/atl/>)
    - **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](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](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](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](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/>