Carleton University
Department of Systems and Computer Engineering

SYSC 5708 (ELG 6178) Model-Driven Development of Real-Time and Distributed Software

Fall 2011

Course Outline

Professor

Dorina C. Petriu, E-mail: petriu@sce.carleton.ca
Room 4434 Mackenzie Building, Tel. 520-5652
(Please note that my office will move soon to the new Engineering Building. I'll post the new contact info on the course web site when it happens).

Course Objectives

A paradigm shift is taking place in the field of software development moving the focus and development effort from code to models. The Model-Driven Development (MDD) paradigm is strongly endorsed by the Object Management Group (OMG) through its Model-Driven Architecture (MDA) initiative. MDA promotes the vision that software development should be based on models throughout the entire system lifecycle: from business modeling to system design, component construction, assembly, integration, code generation, deployment, management, and evolution.

Models provide abstractions of a physical system that allow engineers to reason about that system by ignoring extraneous details while focusing on relevant ones. All forms of engineering rely on models to understand complex, real-world systems. Models are used in many ways: to predict system qualities, reason about specific properties when aspects of the system are changed, etc.

The building of software systems can be organized around a set of models by imposing a series of transformations between models. In a MDD approach, the code that represents the final software product is generated by a series of model transformations implemented by tools. Models also facilitate the analysis of non-functional properties (NFPs), such as performance, scalability, reliability, security, safety, etc. To evaluate a software model for NFPs, analysis models are ideally generated automatically from the software models used for development by model transformations and become part of the model suite maintained with the product.

UML and other OMG standards provide the foundation for MDA. The instructor of the course has been active in OMG working groups as a contributor to two OMG standards, the "UML Profile for Schedulability Performance and Time" (SPT) and "UML Profile for Modeling and Analysis of Real-Time Embedded systems"(MARTE) whose main goal is extend UML with concepts necessary for the modeling and analysis of real-time and distributed systems.

The goal of the course is to teach students concepts related to MDD, such as software modeling languages (e.g., UML), metamodels, extending UML with standard mechanisms, model transformations principles and model transformation languages.

Outline of Lectures

- Introduction to Model-Driven Development (MDD)
- Review of UML 2.X
- Using UML for the development of real-time systems – the COMET methodology: requirement analysis, OO analysis and design.
- Introduction to the UML 2 metamodel. Extending UML through profiling
- Two UML profiles for real-time systems: SPT and MARTE
- Model Transformation Languages
- Principles of model to code transformation
- Performance analysis from UML extended with SPT or MARTE (with focus on model transformation).

Reading

The following titles cover issues of interest covered in this course:

- H. Gomaa, "Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures" Cambridge University Press, ISBN 9780521764148 – covers the COMET methodology. An older book by the same author can be used instead:

  H. Gomaa, "Designing Concurrent, Distributed, and Real-Time Applications with UML" Addison-Wesley, ISBN 0-201-65793-7, 2000.

- B.P. Douglass, "Real Time UML Third edition - Advances in the UML for Real-Time Systems", Addison Wesley 2004, ISBN 0-321-16076-2 – covers the UML 2.X graphical notation
- Kevin Roebuck, “Model-Driven Architecture (MDA): High-Impact Strategies - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors”, Tebbo 2011, ISBN 978-1743044735 – covers MDA concepts
- M. Fowlers, "UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third Edition, Addison-Wesley, 2000, ISBN 978-0321193681 – a brief introduction to the UML 2.X graphical notation
- Selected papers from literature
- Standard documents for UML and MARTE

Activities and Evaluation

- **Assignments (20%)**(up to three)**:** UML 2 and MARTE-based analysis and design of real-time systems.
- **Term Project (30%):** each student will take a deeper investigation of a specific topic related to model-driven development. Examples of project topics: software development

with a UML 2 tool, defining and using UML profiles, model-to model transformations (with languages such as ATL or QVT), model to text transformation, etc.

- **Class Presentation (15%):** each student will make a class presentation of the project. Each presenter will prepare slides that will be posted on the web site; both the oral presentation and the slides will be evaluated by the instructor and the peers based on a common evaluation form.
- **Final Exam (35%)**.

To pass the course, a student must pass the final examination, hand in most of the assignments and the project, and do the presentation.

Note for Students with a Disability

Students with disabilities requiring academic accommodations in this course must register with the Paul Menton Centre for Students with Disabilities for a formal evaluation of disability-related needs. Registered PMC students are required to contact the Centre, 613-520-6608, every term to ensure that I receive your Letter of Accommodation, no later than two weeks before the first assignment is due or the first in-class test/midterm requiring accommodations. If you require accommodation for your formally scheduled exam(s) in this course, please submit your request for accommodation to PMC by November 11th 2011 for Fall term (December exams).

Plagiarism

Plagiarism (copying and handing in for credit someone else's work) is a serious instructional offense that will not be tolerated. Please refer to the chapter on Academic Integrity in the Graduate Calendar for additional information.