SYSC 5708 (ELG 6178) Model-Driven Development of Real-Time and Distributed Software
Fall 2017

Course Outline

Professor
Dr. Dorina C. Petriu, E-mail: petriu@sce.carleton.ca
Room 5205 Canal Building, Tel. 520-2600 x5652

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 promotes the vision that software development should be based on models throughout the entire system lifecycle: from business modeling to requirement analysis, 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 the 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 Object Management Group standards provide the foundation for OMG's approach to MDD, called Model-Driven Architecture (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 to 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, introduction to model transformations principles and model transformation languages.
Outline of Lectures

- Introduction to Model-Driven Development
- Review of UML 2.X modeling language
- 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. Standard UML profiles for real-time systems: SPT and MARTE
- Introduction to Model Transformation Languages
- 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:

- Selected papers from literature.
- Standard documents for UML and MARTE (links provided on the course website).

Activities and Evaluation

- Assignments (20%) (up to three): UML 2 and analysis and design of real-time systems and UML profiles.
- Term Project (35%): each student will take a deeper investigation of a specific topic related to model-driven development. Examples of project topics: a) model-driven design of a problem of your choice with a UML 2 tool, b) defining and using UML profiles, c) model-to model transformations (with languages such as Epsilon ETL, ATL or QVT), model to text transformation, etc.
- Class Presentation (10%): each student will make a class presentation of their project. 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.

**Academic Accommodations**

You may need special arrangements to meet your academic obligations during the term. For an accommodation request the processes are as follows:

**Pregnancy accommodation:** write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website: [http://www2.carleton.ca/equity/](http://www2.carleton.ca/equity/)

**Religious accommodation:** write to me with any requests for academic accommodation during the first two weeks of class, or as soon as possible after the need for accommodation is known to exist. For more details visit the Equity Services website: [http://www2.carleton.ca/equity/](http://www2.carleton.ca/equity/)

**Academic Accommodations for Students with Disabilities:** The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your Letter of Accommodation at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website for the deadline to request accommodations for the formally-scheduled exam at [http://www2.carleton.ca/pmc/new-and-current-students/dates-and-deadlines/](http://www2.carleton.ca/pmc/new-and-current-students/dates-and-deadlines/)

You can visit the Equity Services website to view the policies and to obtain more detailed information on academic accommodation at [http://www2.carleton.ca/equity/](http://www2.carleton.ca/equity/)

**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.

**Course material copyright**

Student or professor materials created for this course (including presentations and posted notes, labs, case studies, assignments and exams) remain the intellectual property of the author(s). They are intended for personal use by students and may not be reproduced or redistributed without prior written consent of the author(s).